DC plugin SDK (C++)

Generated by Doxygen 1.8.4

Fri Jul 26 2013 15:24:46

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 dcapi Namespace Reference

Classes

- class Config
- class Connections
- class Core
- class Hooks
- class Hubs
- class Logger
- class Queue
- class Tagger
- class UI
- class Util

Chapter 5

Class Documentation

5.1 dcapi::Hooks::Chat Struct Reference

```
#include <Hooks.hpp>
```

Static Public Member Functions

- static void onIncomingChat (function< bool(HubDataPtr, char ∗, bool &)> f)
- static void onIncomingPM (function< bool(UserDataPtr, char ∗, bool &)> f)
- static void onOutgoingChat (function< bool(HubDataPtr, char ∗, bool &)> f)
- static void onOutgoingPM (function< bool(UserDataPtr, char ∗, bool &)> f)

5.1.1 Detailed Description

Definition at line 54 of file Hooks.hpp.

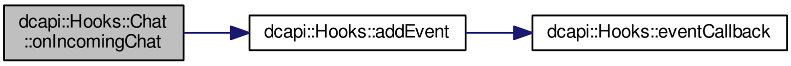
5.1.2 Member Function Documentation

5.1.2.1 void dcapi::Hooks::Chat::onIncomingChat (function< bool(HubDataPtr, char ∗, bool &)> *f*) [static]

Definition at line 38 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_CHAT_IN.

Here is the call graph for this function:

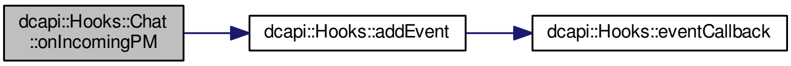

dcapi::Hooks::Chat ::onIncomingChat
dcapi::Hooks::addEvent
dcapi::Hooks::eventCallback

5.1.2.2 void dcapi::Hooks::Chat::onIncomingPM (function< bool(UserDataPtr, char ∗, bool &)> *f*) [static]

Definition at line 46 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_CHAT_PM_IN.

Here is the call graph for this function:

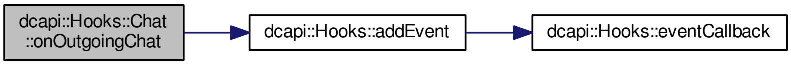

dcapi::Hooks::Chat ::onIncomingPM
dcapi::Hooks::addEvent
dcapi::Hooks::eventCallback

5.1.2.3 void dcapi::Hooks::Chat::onOutgoingChat (function< bool(HubDataPtr, char *, bool &)> *f*) [static]

Definition at line 42 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_CHAT_OUT.

Here is the call graph for this function:

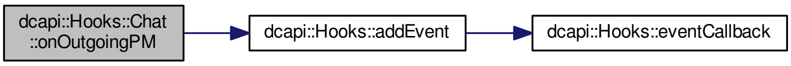

dcapi::Hooks::Chat ::onOutgoingChat
dcapi::Hooks::addEvent
dcapi::Hooks::eventCallback

5.1.2.4 void dcapi::Hooks::Chat::onOutgoingPM (function< bool(UserDataPtr, char *, bool &)> *f*) [static]

Definition at line 50 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_CHAT_PM_OUT.

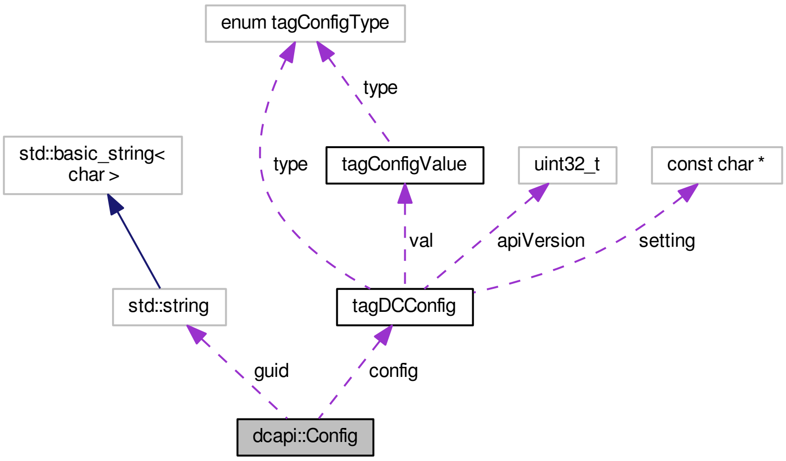
Here is the call graph for this function:


dcapi::Hooks::Chat ::onOutgoingPM
dcapi::Hooks::addEvent
dcapi::Hooks::eventCallback

5.2 dcapi::Config Class Reference

```
#include <Config.hpp>
```

Collaboration diagram for dcapi::Config:



Public Member Functions

- template<typename ConfigT , typename RetT >
  RetT getConfig (const char ∗name, ConfigType type)
- template<typename ConfigT , typename ValueT >
  void setConfig (const char ∗name, ConfigType type, ValueT value)

Static Public Member Functions

- static void freeCoreConfig (ConfigValuePtr value)
- static bool getBoolConfig (const char ∗name)
- static string getConfig (const char ∗name)
- static ConfigValuePtr getCoreConfig (const char ∗name)
- static string getInstallPath ()
- static int64_t getInt64Config (const char ∗name)
- static int32_t getIntConfig (const char ∗name)
- static string getPath (PathType type)
- static DCConfigPtr handle ()
- static bool init (string pluginGuid)
- static void init (DCConfigPtr coreConfig, string pluginGuid)
- static void setConfig (const char ∗name, const char ∗value)
- static void setConfig (const char ∗name, const string &value)
- static void setConfig (const char ∗name, bool value)
- static void setConfig (const char ∗name, int32_t value)
- static void setConfig (const char ∗name, int64_t value)

Static Private Member Functions

- template<typename ConfigT , typename RetT >
  static RetT getConfig (const char ∗name, ConfigType type)
- template<typename ConfigT , typename ValueT >
  static void setConfig (const char ∗name, ConfigType type, ValueT value)

Static Private Attributes

- static DCConfigPtr config
- static string guid

5.2.1 Detailed Description

Definition at line 32 of file Config.hpp.

5.2.2 Member Function Documentation

5.2.2.1 void dcapi::Config::freeCoreConfig (ConfigValuePtr *value*) `[static]`

Definition at line 50 of file Config.cpp.

References config.

5.2.2.2 bool dcapi::Config::getBoolConfig (const char ∗ *name*) `[static]`

Definition at line 45 of file Config.cpp.

References CFG_TYPE_BOOL.

5.2.2.3 string dcapi::Config::getConfig (const char ∗ *name*) `[static]`

Definition at line 44 of file Config.cpp.

References CFG_TYPE_STRING.

5.2.2.4 template<typename ConfigT , typename RetT > static RetT dcapi::Config::getConfig (const char ∗ *name,* ConfigType *type*) `[static]`, `[private]`

5.2.2.5 template<typename ConfigT , typename RetT > RetT dcapi::Config::getConfig (const char ∗ *name,* ConfigType *type*)

Definition at line 70 of file Config.hpp.

References config, and guid.

5.2.2.6 ConfigValuePtr dcapi::Config::getCoreConfig (const char ∗ *name*) `[static]`

Definition at line 49 of file Config.cpp.

References CFG_TYPE_UNKNOWN, and config.

5.2.2.7 string dcapi::Config::getInstallPath ( ) `[static]`

Definition at line 59 of file Config.cpp.

References config, and guid.

5.2.2.8 int64_t dcapi::Config::getInt64Config (const char ∗ *name*) `[static]`

Definition at line 47 of file Config.cpp.

References CFG_TYPE_INT64.

5.2.2.9 int32_t dcapi::Config::getIntConfig (const char ∗ *name*) [static]

Definition at line 46 of file Config.cpp.

References CFG_TYPE_INT.

5.2.2.10 string dcapi::Config::getPath (PathType *type*) [static]

Definition at line 52 of file Config.cpp.

References config.

5.2.2.11 DCConfigPtr dcapi::Config::handle ( ) [static]

Definition at line 36 of file Config.cpp.

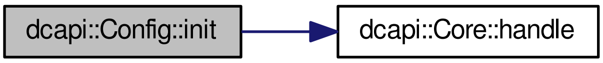
References config.

5.2.2.12 bool dcapi::Config::init (string *pluginGuid*) [static]

Definition at line 30 of file Config.cpp.

References config, DCINTF_CONFIG, DCINTF_CONFIG_VER, and dcapi::Core::handle().

Here is the call graph for this function:



5.2.2.13 void dcapi::Config::init (DCConfigPtr *coreConfig,* string *pluginGuid*) [static]

Definition at line 35 of file Config.cpp.

References config, and guid.

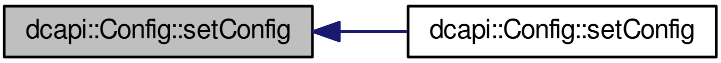
5.2.2.14 void dcapi::Config::setConfig (const char ∗ *name,* const char ∗ *value*) [static]

Definition at line 38 of file Config.cpp.

References CFG_TYPE_STRING.

Referenced by setConfig().

Here is the caller graph for this function:

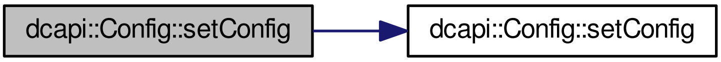


5.2.2.15 void dcapi::Config::setConfig (const char * *name*, const string & *value*) [static]

Definition at line 39 of file Config.cpp.

References setConfig().

Here is the call graph for this function:



5.2.2.16 void dcapi::Config::setConfig (const char * *name*, bool *value*) [static]

Definition at line 40 of file Config.cpp.

References CFG_TYPE_BOOL, False, and True.

5.2.2.17 void dcapi::Config::setConfig (const char * *name*, int32_t *value*) [static]

Definition at line 41 of file Config.cpp.

References CFG_TYPE_INT.

5.2.2.18 void dcapi::Config::setConfig (const char * *name*, int64_t *value*) [static]

Definition at line 42 of file Config.cpp.

References CFG_TYPE_INT64.

5.2.2.19 template<typename ConfigT , typename ValueT > static void dcapi::Config::setConfig (const char * *name*, ConfigType *type*, ValueT *value*) [static], [private]

5.2.2.20 template<typename ConfigT , typename ValueT > void dcapi::Config::setConfig (const char * *name*, ConfigType *type*, ValueT *value*)

Definition at line 65 of file Config.hpp.

References config, and guid.

5.2.3 Member Data Documentation

5.2.3.1 DCConfigPtr dcapi::Config::config [static], [private]

Definition at line 60 of file Config.hpp.

Referenced by freeCoreConfig(), getConfig(), getCoreConfig(), getInstallPath(), getPath(), handle(), init(), and setConfig().

5.2.3.2 string dcapi::Config::guid [static], [private]

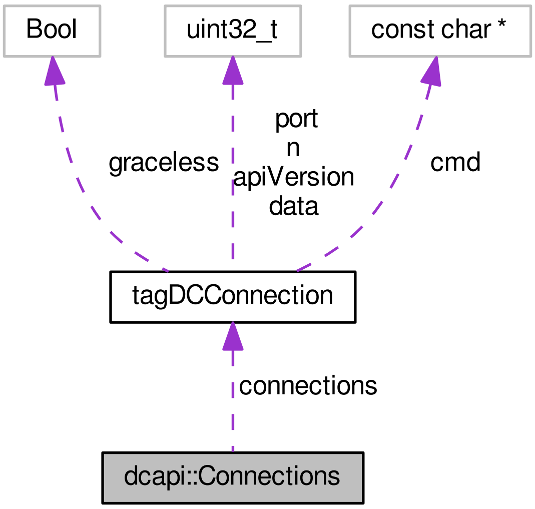
Definition at line 62 of file Config.hpp.

Referenced by getConfig(), getInstallPath(), init(), and setConfig().

5.3 dcapi::Connections Class Reference

```
#include <Connections.hpp>
```

Collaboration diagram for dcapi::Connections:



Static Public Member Functions

- static DCConnectionPtr handle ()
- static bool init ()
- static void init (DCConnectionPtr coreConnections)

Static Private Attributes

- static DCConnectionPtr connections

5.3.1 Detailed Description

Definition at line 28 of file Connections.hpp.

5.3.2 Member Function Documentation

5.3.2.1 DCConnectionPtr dcapi::Connections::handle ( ) [static]

Definition at line 35 of file Connections.cpp.

References connections.

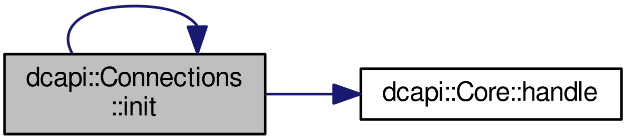
5.3.2.2 bool dcapi::Connections::init ( ) [static]

Definition at line 29 of file Connections.cpp.

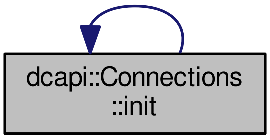
References connections, DCINTF_DCPP_CONNECTIONS, DCINTF_DCPP_CONNECTIONS_VER, dcapi::Core-::handle(), and init().

Referenced by init().

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.2.3 void dcapi::Connections::init (DCConnectionPtr *coreConnections*) [static]

Definition at line 34 of file Connections.cpp.

References connections.

5.3.3 Member Data Documentation

5.3.3.1 DCConnectionPtr dcapi::Connections::connections [static], [private]

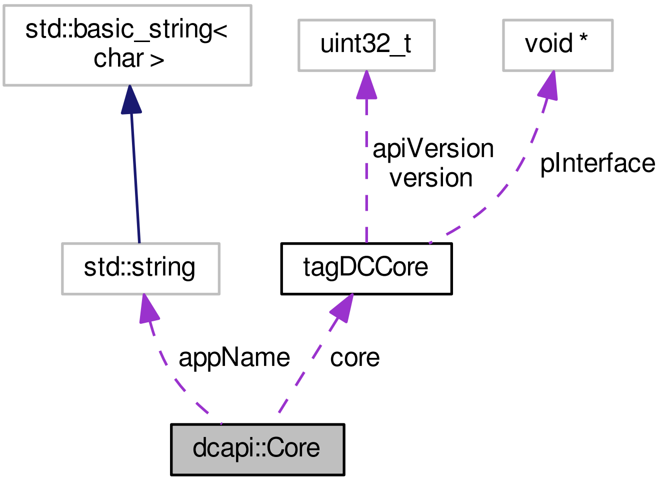
Definition at line 36 of file Connections.hpp.

Referenced by handle(), and init().

5.4 dcapi::Core Class Reference

```
#include <Core.hpp>
```

Collaboration diagram for dcapi::Core:



Static Public Member Functions

- static DCCorePtr handle ()
- static void init (DCCorePtr corePtr)

Static Public Attributes

- static string appName

Static Private Attributes

- static DCCorePtr core

5.4.1 Detailed Description

Definition at line 32 of file Core.hpp.

5.4.2 Member Function Documentation

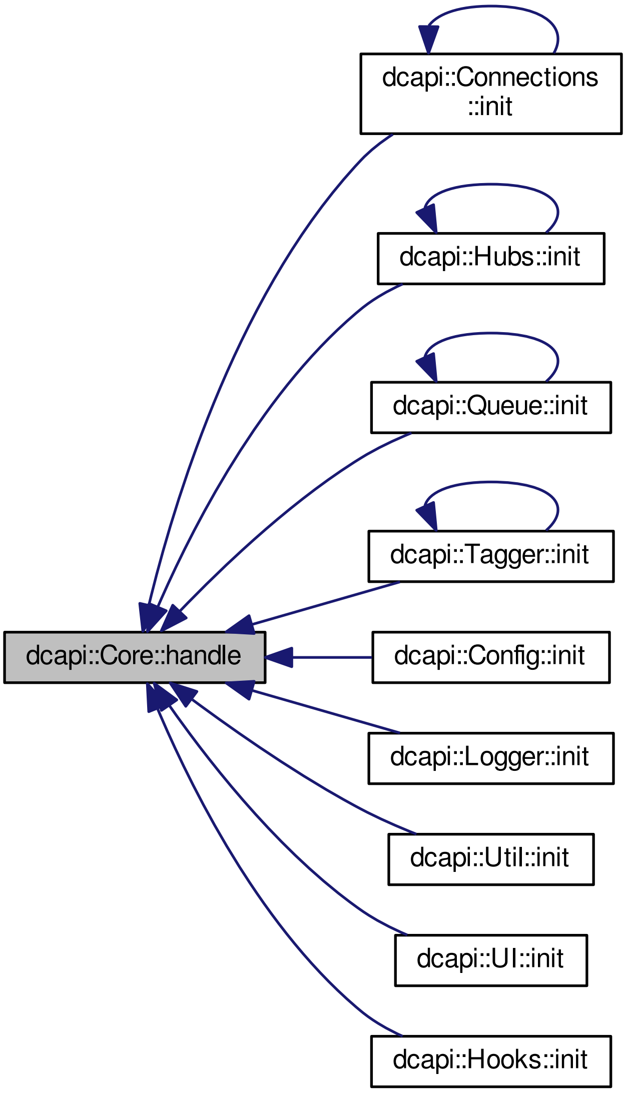
5.4.2.1 DCCorePtr dcapi::Core::handle ( ) [static]

Definition at line 33 of file Core.cpp.

References core.

Referenced by dcapi::Connections::init(), dcapi::Hubs::init(), dcapi::Queue::init(), dcapi::Tagger::init(), dcapi::Config::init(), dcapi::Logger::init(), dcapi::Util::init(), dcapi::UI::init(), and dcapi::Hooks::init().

Here is the caller graph for this function:

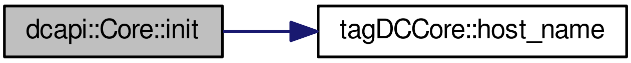


5.4.2.2 void dcapi::Core::init (**DCCorePtr** *corePtr*) [static]

Definition at line 28 of file Core.cpp.

References appName, and tagDCCore::host_name().

Here is the call graph for this function:



5.4.3 Member Data Documentation

5.4.3.1 string dcapi::Core::appName [static]

Definition at line 38 of file Core.hpp.

Referenced by init().

5.4.3.2 DCCorePtr dcapi::Core::core [static], [private]

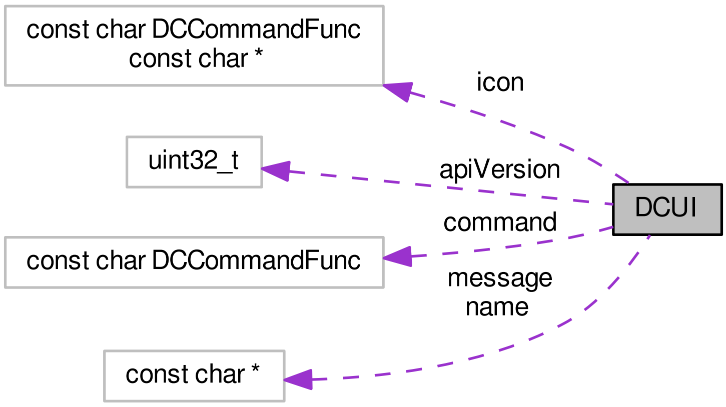
Definition at line 41 of file Core.hpp.

Referenced by handle().

5.5 DCUI Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for DCUI:



Public Member Functions

- void (DCAPI ∗add_command)(const char ∗guid
- void (DCAPI ∗remove_command)(const char ∗guid
- void (DCAPI ∗play_sound)(const char ∗path)
- void (DCAPI ∗notify)(const char ∗title

Public Attributes

- uint32_t apiVersion
- const char DCCommandFunc command
- const char DCCommandFunc const char ∗ icon
- const char ∗ message
- const char ∗ name

5.5.1 Detailed Description

Definition at line 448 of file PluginDefs.h.

5.5.2 Member Function Documentation

5.5.2.1 DCUI::void (**DCAPI** ∗ *add_command*) const

5.5.2.2 DCUI::void (**DCAPI** ∗ *remove_command*) const

5.5.2.3 DCUI::void (**DCAPI** ∗ *play_sound*) const

5.5.2.4 DCUI::void (**DCAPI** ∗ *notify*) const

5.5.3 Member Data Documentation

5.5.3.1 uint32_t DCUI::apiVersion

Definition at line 450 of file PluginDefs.h.

5.5.3.2 const char DCCommandFunc DCUI::command

Definition at line 454 of file PluginDefs.h.

5.5.3.3 const char DCCommandFunc const char∗ DCUI::icon

Definition at line 454 of file PluginDefs.h.

5.5.3.4 const char∗ DCUI::message

Definition at line 459 of file PluginDefs.h.

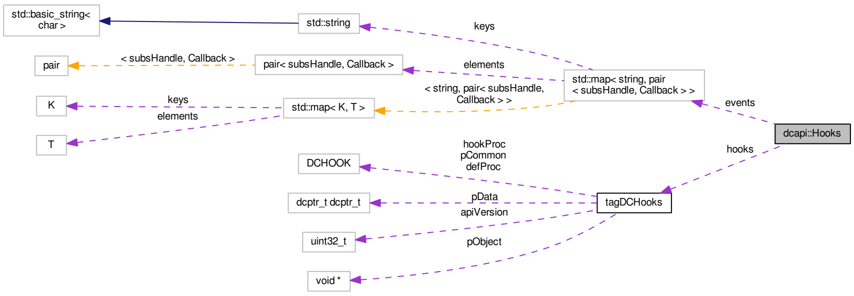
5.5.3.5 const char ∗ DCUI::name

Definition at line 454 of file PluginDefs.h.

5.6 dcapi::Hooks Class Reference

`#include <Hooks.hpp>`

Collaboration diagram for dcapi::Hooks:



Classes

- struct Chat
- struct Hubs
- struct Network
- struct Queue
- struct Timer
- struct UI
- struct Users

Static Public Member Functions

- static void clear ()
- static bool empty ()
- static DCHooksPtr handle ()
- static bool init ()
- static void init (DCHooksPtr coreHooks)
- static void remove (const char ∗id)

Private Types

- typedef function< bool(dcptr_t,
  dcptr_t, bool &)> Callback

Static Private Member Functions

- static void addEvent (const char ∗id, Callback f)
- static Bool DCAPI eventCallback (dcptr_t pObject, dcptr_t pData, dcptr_t pCommon, Bool ∗bBreak)

Static Private Attributes

- static map< string, pair
  < subsHandle, Callback > > events
- static DCHooksPtr hooks

5.6.1 Detailed Description

Definition at line 38 of file Hooks.hpp.

5.6.2 Member Typedef Documentation

5.6.2.1 typedef function<bool (**dcptr_t, dcptr_t,** bool&)> **dcapi::Hooks::Callback** `[private]`

Definition at line 105 of file Hooks.hpp.

5.6.3 Member Function Documentation

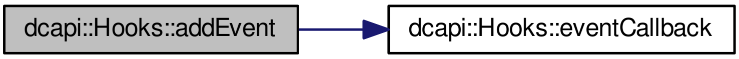
5.6.3.1 void dcapi::Hooks::addEvent (const char ∗ *id*, **Callback** *f*) `[static]`, `[private]`

Definition at line 163 of file Hooks.cpp.

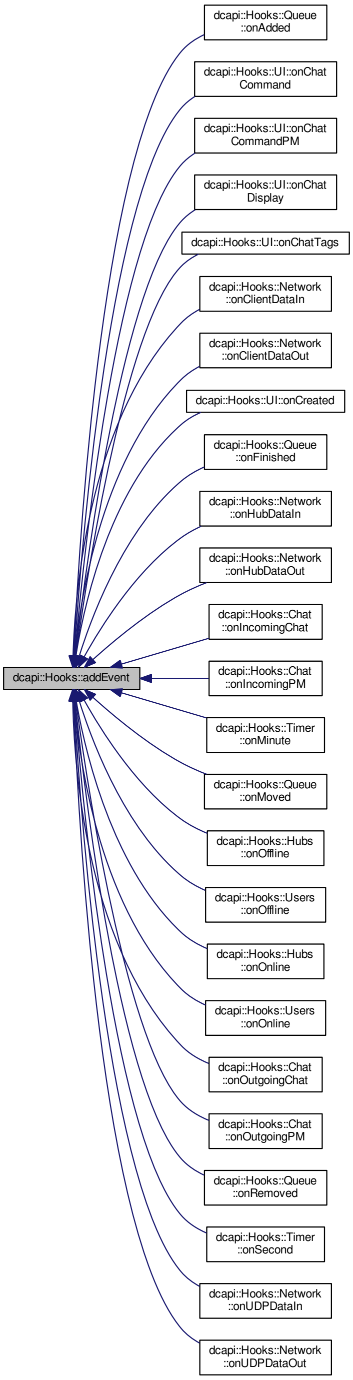
References eventCallback(), events, and hooks.

Referenced by dcapi::Hooks::Queue::onAdded(), dcapi::Hooks::UI::onChatCommand(), dcapi::Hooks::UI::onChatCommandPM(), dcapi::Hooks::UI::onChatDisplay(), dcapi::Hooks::UI::onChatTags(), dcapi::Hooks::Network::onClientDataIn(), dcapi::Hooks::Network::onClientDataOut(), dcapi::Hooks::UI::onCreated(), dcapi::Hooks::Queue::onFinished(), dcapi::Hooks::Network::onHubDataIn(), dcapi::Hooks::Network::onHubDataOut(), dcapi::Hooks::Chat::onIncomingChat(), dcapi::Hooks::Chat::onIncomingPM(), dcapi::Hooks::Timer::onMinute(), dcapi::Hooks::Queue::onMoved(), dcapi::Hooks::Hubs::onOffline(), dcapi::Hooks::Users::onOffline(), dcapi::Hooks::Hubs::onOnline(), dcapi::Hooks::Users::onOnline(), dcapi::Hooks::Chat::onOutgoingChat(), dcapi::Hooks::Chat::onOutgoingPM(), dcapi::Hooks::Queue::onRemoved(), dcapi::Hooks::Timer::onSecond(), dcapi::Hooks::Network::onUDPDataIn(), and dcapi::Hooks::Network::onUDPDataOut().

Here is the call graph for this function:



Here is the caller graph for this function:


dcapi::Hooks::addEvent
dcapi::Hooks::Queue ::onAdded
dcapi::Hooks::UI::onChat Command
dcapi::Hooks::UI::onChat CommandPM
dcapi::Hooks::UI::onChat Display
dcapi::Hooks::UI::onChatTags
dcapi::Hooks::Network ::onClientDataIn
dcapi::Hooks::Network ::onClientDataOut
dcapi::Hooks::UI::onCreated
dcapi::Hooks::Queue ::onFinished
dcapi::Hooks::Network ::onHubDataIn
dcapi::Hooks::Network ::onHubDataOut
dcapi::Hooks::Chat ::onIncomingChat
dcapi::Hooks::Chat ::onIncomingPM
dcapi::Hooks::Timer ::onMinute
dcapi::Hooks::Queue ::onMoved
dcapi::Hooks::Hubs ::onOffline
dcapi::Hooks::Users ::onOffline
dcapi::Hooks::Hubs ::onOnline
dcapi::Hooks::Users ::onOnline
dcapi::Hooks::Chat ::onOutgoingChat
dcapi::Hooks::Chat ::onOutgoingPM
dcapi::Hooks::Queue ::onRemoved
dcapi::Hooks::Timer ::onSecond
dcapi::Hooks::Network ::onUDPDataIn
dcapi::Hooks::Network ::onUDPDataOut

5.6.3.2 void dcapi::Hooks::clear ( ) [static]

Definition at line 149 of file Hooks.cpp.

References events, and hooks.

5.6.3.3 bool dcapi::Hooks::empty ( ) [static]

Definition at line 145 of file Hooks.cpp.

References events.

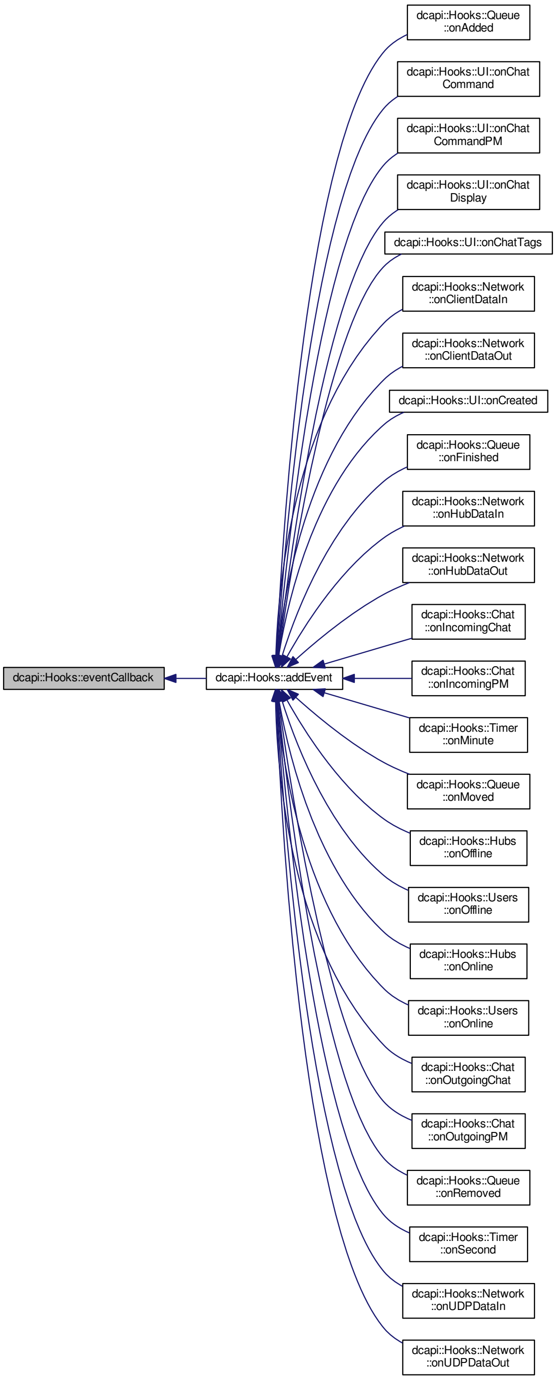
5.6.3.4 Bool DCAPI dcapi::Hooks::eventCallback (dcptr_t *pObject*, dcptr_t *pData*, dcptr_t *pCommon*, Bool ∗ *bBreak*) [static], [private]

Definition at line 169 of file Hooks.cpp.

References dcTrue, events, False, and True.

Referenced by addEvent().

Here is the caller graph for this function:


dcapi::Hooks::eventCallback
dcapi::Hooks::addEvent
dcapi::Hooks::Queue ::onAdded
dcapi::Hooks::UI::onChat Command
dcapi::Hooks::UI::onChat CommandPM
dcapi::Hooks::UI::onChat Display
dcapi::Hooks::UI::onChatTags
dcapi::Hooks::Network ::onClientDataIn
dcapi::Hooks::Network ::onClientDataOut
dcapi::Hooks::UI::onCreated
dcapi::Hooks::Queue ::onFinished
dcapi::Hooks::Network ::onHubDataIn
dcapi::Hooks::Network ::onHubDataOut
dcapi::Hooks::Chat ::onIncomingChat
dcapi::Hooks::Chat ::onIncomingPM
dcapi::Hooks::Timer ::onMinute
dcapi::Hooks::Queue ::onMoved
dcapi::Hooks::Hubs ::onOffline
dcapi::Hooks::Users ::onOffline
dcapi::Hooks::Hubs ::onOnline
dcapi::Hooks::Users ::onOnline
dcapi::Hooks::Chat ::onOutgoingChat
dcapi::Hooks::Chat ::onOutgoingPM
dcapi::Hooks::Queue ::onRemoved
dcapi::Hooks::Timer ::onSecond
dcapi::Hooks::Network ::onUDPDataIn
dcapi::Hooks::Network ::onUDPDataOut

5.6.3.5 DCHooksPtr dcapi::Hooks::handle ( ) [static]

Definition at line 36 of file Hooks.cpp.

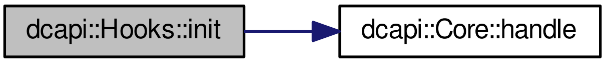
References hooks.

5.6.3.6 bool dcapi::Hooks::init ( ) [static]

Definition at line 30 of file Hooks.cpp.

References DCINTF_HOOKS, DCINTF_HOOKS_VER, dcapi::Core::handle(), and hooks.

Here is the call graph for this function:



5.6.3.7 void dcapi::Hooks::init (DCHooksPtr *coreHooks*) [static]

Definition at line 35 of file Hooks.cpp.

References hooks.

5.6.3.8 void dcapi::Hooks::remove (const char ∗ *id*) [static]

Definition at line 155 of file Hooks.cpp.

References events, and hooks.

5.6.4 Member Data Documentation

5.6.4.1 map< string, pair< subsHandle, Hooks::Callback > > dcapi::Hooks::events [static], [private]

Definition at line 111 of file Hooks.hpp.

Referenced by addEvent(), clear(), empty(), eventCallback(), and remove().

5.6.4.2 DCHooksPtr dcapi::Hooks::hooks [static], [private]

Definition at line 109 of file Hooks.hpp.

Referenced by addEvent(), clear(), handle(), init(), and remove().

5.7 dcapi::Hooks::Hubs Struct Reference

```
#include <Hooks.hpp>
```

Static Public Member Functions

- static void onOffline (function< bool(HubDataPtr, bool &)> f)
- static void onOnline (function< bool(HubDataPtr, bool &)> f)

5.7.1 Detailed Description

Definition at line 66 of file Hooks.hpp.

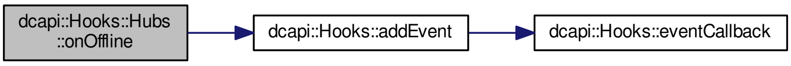
5.7.2 Member Function Documentation

5.7.2.1 void dcapi::Hooks::Hubs::onOffline (function< bool(HubDataPtr, bool &)> *f*) [static]

Definition at line 68 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_HUB_OFFLINE.

Here is the call graph for this function:

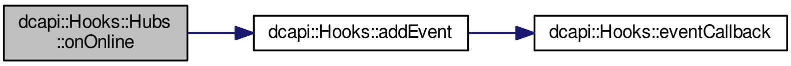


5.7.2.2 void dcapi::Hooks::Hubs::onOnline (function< bool(HubDataPtr, bool &)> *f*) [static]

Definition at line 64 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_HUB_ONLINE.

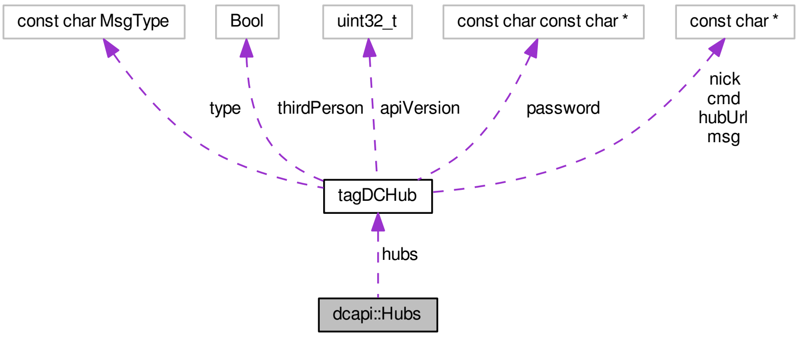
Here is the call graph for this function:



5.8 dcapi::Hubs Class Reference

```
#include <Hubs.hpp>
```

Collaboration diagram for dcapi::Hubs:



Static Public Member Functions

- static DCHubPtr handle ()
- static bool init ()
- static void init (DCHubPtr coreHubs)

Static Private Attributes

- static DCHubPtr hubs

5.8.1 Detailed Description

Definition at line 28 of file Hubs.hpp.

5.8.2 Member Function Documentation

5.8.2.1 DCHubPtr dcapi::Hubs::handle ( ) [static]

Definition at line 35 of file Hubs.cpp.

References hubs.

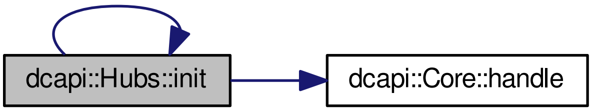
5.8.2.2 bool dcapi::Hubs::init ( ) [static]

Definition at line 29 of file Hubs.cpp.

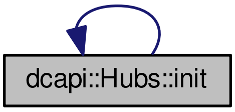
References DCINTF_DCPP_HUBS, DCINTF_DCPP_HUBS_VER, dcapi::Core::handle(), hubs, and init().

Referenced by init().

Here is the call graph for this function:



Here is the caller graph for this function:



5.8.2.3 void dcapi::Hubs::init (DCHubPtr *coreHubs*) [static]

Definition at line 34 of file Hubs.cpp.

References hubs.

5.8.3 Member Data Documentation

5.8.3.1 DCHubPtr dcapi::Hubs::hubs [static], [private]

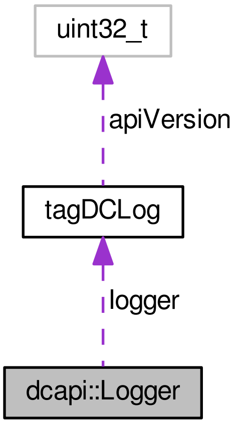
Definition at line 36 of file Hubs.hpp.

Referenced by handle(), and init().

5.9 dcapi::Logger Class Reference

```
#include <Logger.hpp>
```

Collaboration diagram for dcapi::Logger:



Static Public Member Functions

- static DCLogPtr handle ()
- static bool init ()
- static void init (DCLogPtr coreLogger)
- static void log (const string &message)

Static Private Attributes

- static DCLogPtr logger

5.9.1 Detailed Description

Definition at line 32 of file Logger.hpp.

5.9.2 Member Function Documentation

5.9.2.1 DCLogPtr dcapi::Logger::handle ( ) [static]

Definition at line 35 of file Logger.cpp.

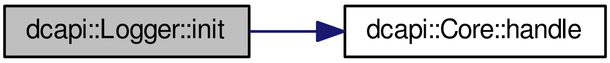
References logger.

5.9.2.2 bool dcapi::Logger::init ( ) [static]

Definition at line 29 of file Logger.cpp.

References DCINTF_LOGGING, DCINTF_LOGGING_VER, dcapi::Core::handle(), and logger.

Here is the call graph for this function:



5.9.2.3 void dcapi::Logger::init (DCLogPtr *coreLogger*) [static]

Definition at line 34 of file Logger.cpp.

References logger.

5.9.2.4 void dcapi::Logger::log (const string & *message*) [static]

Definition at line 37 of file Logger.cpp.

References logger.

5.9.3 Member Data Documentation

5.9.3.1 DCLogPtr dcapi::Logger::logger [static], [private]

Definition at line 42 of file Logger.hpp.

Referenced by handle(), init(), and log().

5.10 dcapi::Hooks::Network Struct Reference

```
#include <Hooks.hpp>
```

Static Public Member Functions

- static void onClientDataIn (function< bool(ConnectionDataPtr, char ∗, bool &)> f)
- static void onClientDataOut (function< bool(ConnectionDataPtr, char ∗, bool &)> f)
- static void onHubDataIn (function< bool(HubDataPtr, char ∗, bool &)> f)
- static void onHubDataOut (function< bool(HubDataPtr, char ∗, bool &)> f)
- static void onUDPDataIn (function< bool(UDPDataPtr, char ∗, bool &)> f)
- static void onUDPDataOut (function< bool(UDPDataPtr, char ∗, bool &)> f)

5.10.1 Detailed Description

Definition at line 76 of file Hooks.hpp.

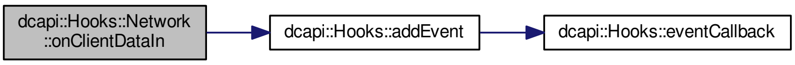
5.10.2 Member Function Documentation

5.10.2.1 void dcapi::Hooks::Network::onClientDataIn (function< bool(ConnectionDataPtr, char *, bool &)> *f*) [static]

Definition at line 90 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_NETWORK_CONN_IN.

Here is the call graph for this function:

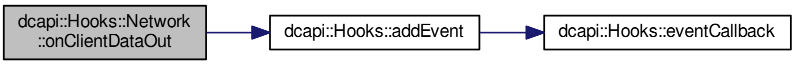

dcapi::Hooks::Network ::onClientDataIn
dcapi::Hooks::addEvent
dcapi::Hooks::eventCallback

5.10.2.2 void dcapi::Hooks::Network::onClientDataOut (function< bool(ConnectionDataPtr, char *, bool &)> *f*) [static]

Definition at line 94 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_NETWORK_CONN_OUT.

Here is the call graph for this function:

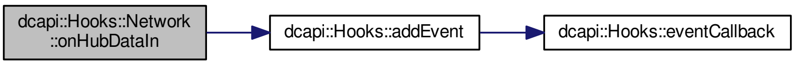

dcapi::Hooks::Network ::onClientDataOut
dcapi::Hooks::addEvent
dcapi::Hooks::eventCallback

5.10.2.3 void dcapi::Hooks::Network::onHubDataIn (function< bool(HubDataPtr, char *, bool &)> *f*) [static]

Definition at line 82 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_NETWORK_HUB_IN.

Here is the call graph for this function:

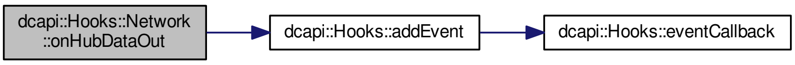

dcapi::Hooks::Network ::onHubDataIn
dcapi::Hooks::addEvent
dcapi::Hooks::eventCallback

5.10.2.4 void dcapi::Hooks::Network::onHubDataOut (function< bool(HubDataPtr, char *, bool &)> *f*) [static]

Definition at line 86 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_NETWORK_HUB_OUT.

Here is the call graph for this function:

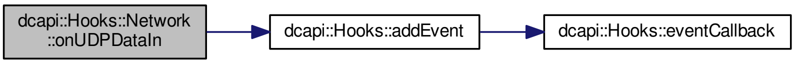

dcapi::Hooks::Network ::onHubDataOut
dcapi::Hooks::addEvent
dcapi::Hooks::eventCallback

5.10.2.5 void dcapi::Hooks::Network::onUDPDataIn (function< bool(UDPDataPtr, char *, bool &)> *f*) [static]

Definition at line 98 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_NETWORK_UDP_IN.

Here is the call graph for this function:

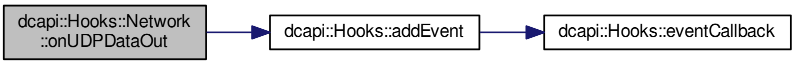

dcapi::Hooks::Network ::onUDPDataIn
dcapi::Hooks::addEvent
dcapi::Hooks::eventCallback

5.10.2.6 void dcapi::Hooks::Network::onUDPDataOut (function< bool(UDPDataPtr, char *, bool &)> *f*) [static]

Definition at line 102 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_NETWORK_UDP_OUT.

Here is the call graph for this function:


dcapi::Hooks::Network ::onUDPDataOut
dcapi::Hooks::addEvent
dcapi::Hooks::eventCallback

5.11 Plugin Class Reference

`#include <Plugin.h>`

Static Public Member Functions

- static Bool DCAPI main (PluginState state, DCCorePtr core, dcptr_t)

Private Member Functions

- bool onLoad (DCCorePtr core, bool install, bool runtime)
- Plugin ()
- ∼Plugin ()

5.11.1 Detailed Description

Definition at line 24 of file Plugin.h.

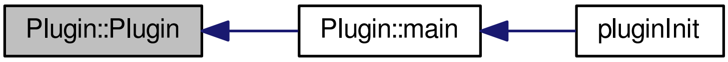
5.11.2 Constructor & Destructor Documentation

5.11.2.1 Plugin::Plugin ( ) [private]

Definition at line 37 of file Plugin.cpp.

Referenced by main().

Here is the caller graph for this function:



5.11.2.2 Plugin::∼Plugin ( ) [private]

Definition at line 40 of file Plugin.cpp.

5.11.3 Member Function Documentation

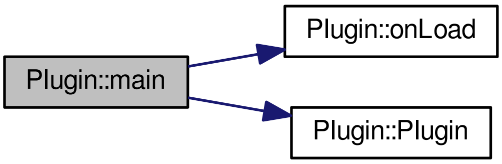
5.11.3.1 Bool DCAPI Plugin::main (PluginState *state*, DCCorePtr *core*, dcptr_t) [static]

Definition at line 44 of file Plugin.cpp.

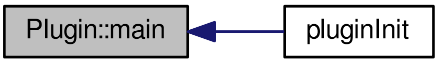
References False, ON_INSTALL, ON_LOAD, ON_LOAD_RUNTIME, ON_UNINSTALL, ON_UNLOAD, onLoad(), Plugin(), and True.

Referenced by pluginInit().

Here is the call graph for this function:



Here is the caller graph for this function:



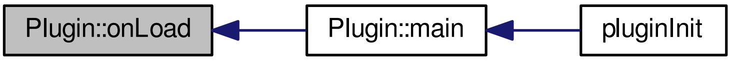
5.11.3.2 bool Plugin::onLoad (DCCorePtr *core*, bool *install*, bool *runtime*) [private]

Definition at line 71 of file Plugin.cpp.

References PLUGIN_GUID.

Referenced by main().

Here is the caller graph for this function:



5.12 dcapi::Hooks::Queue Struct Reference

```
#include <Hooks.hpp>
```

Static Public Member Functions

- static void onAdded (function< bool(QueueDataPtr, bool &)> f)

- static void onFinished (function< bool(QueueDataPtr, bool &)> f)
- static void onMoved (function< bool(QueueDataPtr, bool &)> f)
- static void onRemoved (function< bool(QueueDataPtr, bool &)> f)

5.12.1 Detailed Description

Definition at line 85 of file Hooks.hpp.

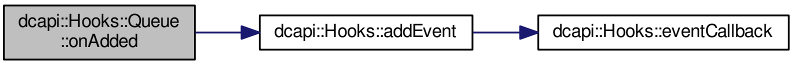
5.12.2 Member Function Documentation

5.12.2.1 void dcapi::Hooks::Queue::onAdded (function< bool(QueueDataPtr, bool &)> *f*) [static]

Definition at line 107 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_QUEUE_ADDED.

Here is the call graph for this function:

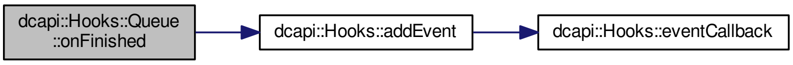


5.12.2.2 void dcapi::Hooks::Queue::onFinished (function< bool(QueueDataPtr, bool &)> *f*) [static]

Definition at line 119 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_QUEUE_FINISHED.

Here is the call graph for this function:

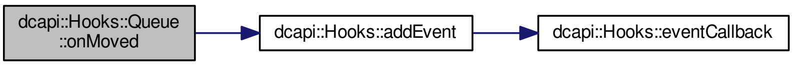


5.12.2.3 void dcapi::Hooks::Queue::onMoved (function< bool(QueueDataPtr, bool &)> *f*) [static]

Definition at line 111 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_QUEUE_MOVED.

Here is the call graph for this function:

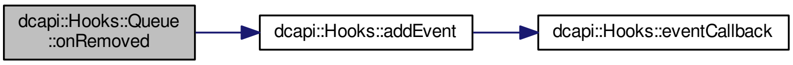


5.12.2.4 void dcapi::Hooks::Queue::onRemoved (function< bool(QueueDataPtr, bool &)> *f*) [static]

Definition at line 115 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_QUEUE_REMOVED.

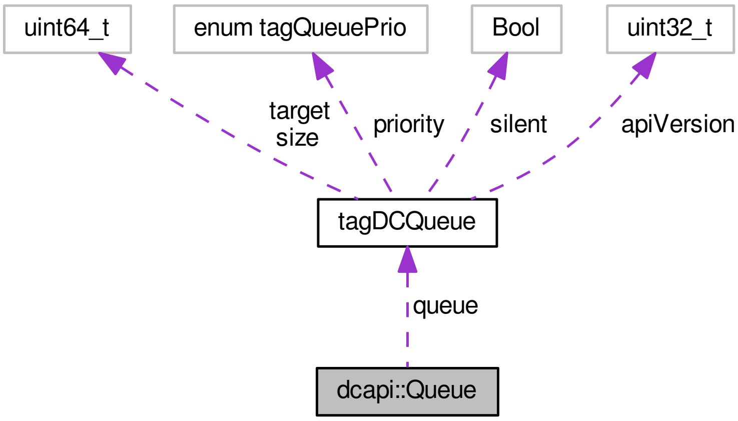
Here is the call graph for this function:



5.13 dcapi::Queue Class Reference

`#include <Queue.hpp>`

Collaboration diagram for dcapi::Queue:



Static Public Member Functions

- static DCQueuePtr handle ()

- static bool init ()
- static void init (DCQueuePtr coreQueue)

Static Private Attributes

- static DCQueuePtr queue

5.13.1 Detailed Description

Definition at line 28 of file Queue.hpp.

5.13.2 Member Function Documentation

5.13.2.1 DCQueuePtr dcapi::Queue::handle ( ) [static]

Definition at line 35 of file Queue.cpp.

References queue.

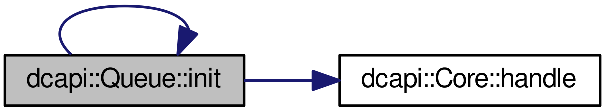
5.13.2.2 bool dcapi::Queue::init ( ) [static]

Definition at line 29 of file Queue.cpp.

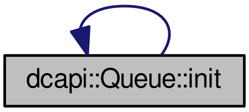
References DCINTF_DCPP_QUEUE, DCINTF_DCPP_QUEUE_VER, dcapi::Core::handle(), init(), and queue.

Referenced by init().

Here is the call graph for this function:



Here is the caller graph for this function:



5.13.2.3 void dcapi::Queue::init (DCQueuePtr *coreQueue*) [static]

Definition at line 34 of file Queue.cpp.

References queue.

5.13.3 Member Data Documentation

5.13.3.1 DCQueuePtr dcapi::Queue::queue [static], [private]

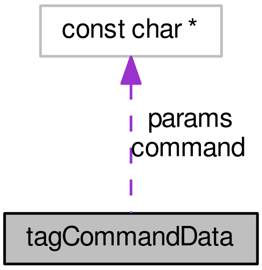
Definition at line 36 of file Queue.hpp.

Referenced by handle(), and init().

5.14 tagCommandData Struct Reference

`#include <PluginDefs.h>`

Collaboration diagram for tagCommandData:



Public Attributes

- const char $*$ command
- const char $*$ params

5.14.1 Detailed Description

Definition at line 212 of file PluginDefs.h.

5.14.2 Member Data Documentation

5.14.2.1 const char$*$ tagCommandData::command

Definition at line 213 of file PluginDefs.h.

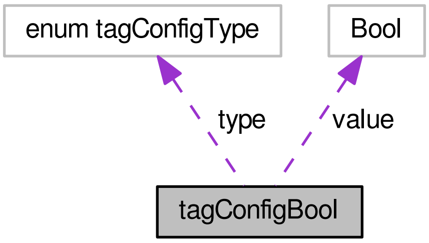
5.14.2.2 const char∗ tagCommandData::params

Definition at line 214 of file PluginDefs.h.

5.15 tagConfigBool Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagConfigBool:



Public Attributes

- ConfigType type
- Bool value

5.15.1 Detailed Description

Definition at line 194 of file PluginDefs.h.

5.15.2 Member Data Documentation

5.15.2.1 ConfigType tagConfigBool::type

Definition at line 195 of file PluginDefs.h.

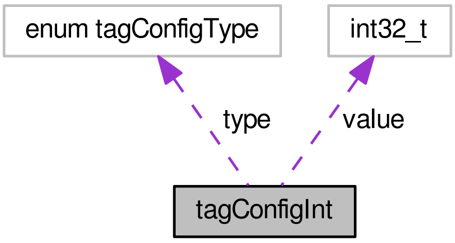
5.15.2.2 Bool tagConfigBool::value

Definition at line 196 of file PluginDefs.h.

5.16 tagConfigInt Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagConfigInt:



Public Attributes

- ConfigType type
- int32_t value

5.16.1 Detailed Description

Definition at line 188 of file PluginDefs.h.

5.16.2 Member Data Documentation

5.16.2.1 ConfigType tagConfigInt::type

Definition at line 189 of file PluginDefs.h.

Referenced by set_cfg_int().

5.16.2.2 int32_t tagConfigInt::value

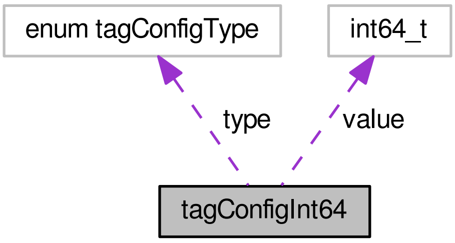
Definition at line 190 of file PluginDefs.h.

Referenced by get_cfg_int(), and set_cfg_int().

5.17 tagConfigInt64 Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagConfigInt64:



Public Attributes

- ConfigType type
- int64_t value

5.17.1 Detailed Description

Definition at line 200 of file PluginDefs.h.

5.17.2 Member Data Documentation

5.17.2.1 ConfigType tagConfigInt64::type

Definition at line 201 of file PluginDefs.h.

Referenced by set_cfg_int64().

5.17.2.2 int64_t tagConfigInt64::value

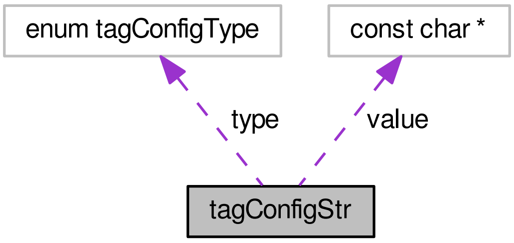
Definition at line 202 of file PluginDefs.h.

Referenced by get_cfg_int64(), and set_cfg_int64().

5.18 tagConfigStr Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagConfigStr:



Public Attributes

- ConfigType type
- const char ∗ value

5.18.1 Detailed Description

Definition at line 182 of file PluginDefs.h.

5.18.2 Member Data Documentation

5.18.2.1 ConfigType tagConfigStr::type

Definition at line 183 of file PluginDefs.h.

Referenced by set_cfg().

5.18.2.2 const char∗ tagConfigStr::value

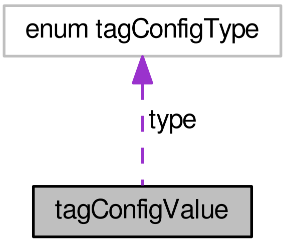
Definition at line 184 of file PluginDefs.h.

Referenced by get_cfg(), get_cfg_path(), get_install_path(), and set_cfg().

5.19 tagConfigValue Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagConfigValue:



Public Attributes

- ConfigType type

5.19.1 Detailed Description

Definition at line 177 of file PluginDefs.h.

5.19.2 Member Data Documentation

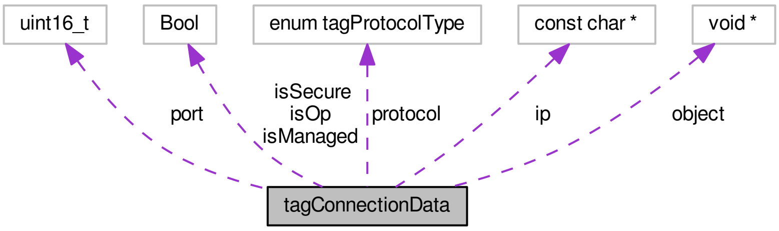
5.19.2.1 ConfigType tagConfigValue::type

Definition at line 178 of file PluginDefs.h.

5.20 tagConnectionData Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagConnectionData:



Public Attributes

- const char ∗ ip
- Bool isManaged
- Bool isOp
- Bool isSecure
- dcptr_t object
- uint16_t port
- ProtocolType protocol

5.20.1 Detailed Description

Definition at line 242 of file PluginDefs.h.

5.20.2 Member Data Documentation

5.20.2.1 const char∗ tagConnectionData::ip

Definition at line 243 of file PluginDefs.h.

5.20.2.2 Bool tagConnectionData::isManaged

Definition at line 249 of file PluginDefs.h.

5.20.2.3 Bool tagConnectionData::isOp

Definition at line 247 of file PluginDefs.h.

5.20.2.4 Bool tagConnectionData::isSecure

Definition at line 248 of file PluginDefs.h.

5.20.2.5 dcptr_t tagConnectionData::object

Definition at line 244 of file PluginDefs.h.

5.20.2.6 uint16_t tagConnectionData::port

Definition at line 245 of file PluginDefs.h.

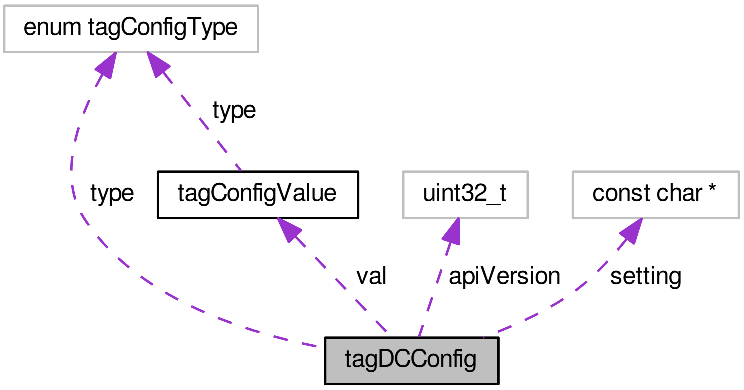
5.20.2.7 ProtocolType tagConnectionData::protocol

Definition at line 246 of file PluginDefs.h.

5.21 tagDCConfig Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagDCConfig:



Public Member Functions

- ConfigStrPtr (DCAPI ∗get_path)(PathType type)
- ConfigStrPtr (DCAPI ∗get_install_path)(const char ∗guid)
- ConfigStrPtr (DCAPI ∗get_language)()
- ConfigValuePtr (DCAPI ∗get_cfg)(const char ∗guid
- ConfigValuePtr (DCAPI ∗copy)(const ConfigValuePtr val)
- void (DCAPI ∗set_cfg)(const char ∗guid
- void (DCAPI ∗release)(ConfigValuePtr val)

Public Attributes

- uint32_t apiVersion
- const char ∗ setting
- const char ConfigType type
- const char ConfigValuePtr val

5.21.1 Detailed Description

Definition at line 337 of file PluginDefs.h.

5.21.2 Member Function Documentation

5.21.2.1 tagDCConfig::ConfigStrPtr (**DCAPI** ∗ *get_path*)

5.21.2.2 tagDCConfig::ConfigStrPtr (**DCAPI** ∗ *get_install_path*) const

5.21.2.3 tagDCConfig::ConfigStrPtr (**DCAPI** ∗ *get_language*)

5.21.2.4 tagDCConfig::ConfigValuePtr (**DCAPI** ∗ *get_cfg*) const

5.21.2.5 tagDCConfig::ConfigValuePtr (**DCAPI** ∗ *copy*) const

5.21.2.6 tagDCConfig::void (**DCAPI** ∗ *set_cfg*) const

5.21.2.7 tagDCConfig::void (**DCAPI** ∗ *release*)

5.21.3 Member Data Documentation

5.21.3.1 uint32_t tagDCConfig::apiVersion

Definition at line 339 of file PluginDefs.h.

5.21.3.2 const char ∗ tagDCConfig::setting

Definition at line 344 of file PluginDefs.h.

5.21.3.3 const char **ConfigType** tagDCConfig::type

Definition at line 345 of file PluginDefs.h.

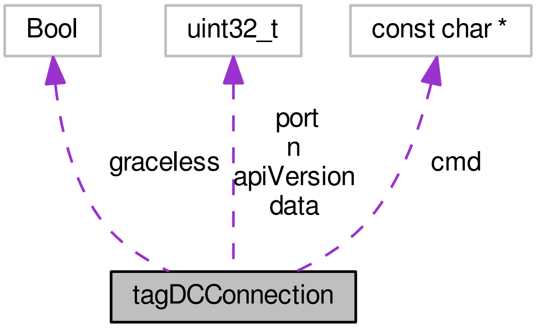
5.21.3.4 const char **ConfigValuePtr** tagDCConfig::val

Definition at line 344 of file PluginDefs.h.

5.22 tagDCConnection Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagDCConnection:



Public Member Functions

- UserDataPtr (DCAPI ∗get_user)(ConnectionDataPtr hConn)
- void (DCAPI ∗send_udp_data)(const char ∗ip
- void (DCAPI ∗send_protocol_cmd)(ConnectionDataPtr hConn

- void (DCAPI ∗terminate_conn)(ConnectionDataPtr hConn

Public Attributes

- uint32_t apiVersion
- const char ∗ cmd
- uint32_t dcptr_t data
- Bool graceless
- uint32_t dcptr_t size_t n
- uint32_t port

5.22.1 Detailed Description

Definition at line 365 of file PluginDefs.h.

5.22.2 Member Function Documentation

5.22.2.1 tagDCConnection::UserDataPtr (DCAPI ∗ *get_user*)

5.22.2.2 tagDCConnection::void (DCAPI ∗ *send_udp_data*) const

5.22.2.3 tagDCConnection::void (DCAPI ∗ *send_protocol_cmd*)

5.22.2.4 tagDCConnection::void (DCAPI ∗ *terminate_conn*)

5.22.3 Member Data Documentation

5.22.3.1 uint32_t tagDCConnection::apiVersion

Definition at line 367 of file PluginDefs.h.

5.22.3.2 const char∗ tagDCConnection::cmd

Definition at line 370 of file PluginDefs.h.

5.22.3.3 uint32_t dcptr_t tagDCConnection::data

Definition at line 369 of file PluginDefs.h.

5.22.3.4 Bool tagDCConnection::graceless

Definition at line 371 of file PluginDefs.h.

5.22.3.5 uint32_t dcptr_t size_t tagDCConnection::n

Definition at line 369 of file PluginDefs.h.

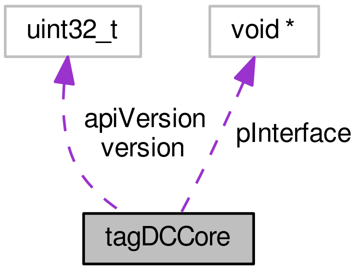
5.22.3.6 uint32_t tagDCConnection::port

Definition at line 369 of file PluginDefs.h.

5.23 tagDCCore Struct Reference

`#include <PluginDefs.h>`

Collaboration diagram for tagDCCore:



Public Member Functions

- Bool (DCAPI ∗release_interface)(intfHandle hInterface)
- Bool (DCAPI ∗has_plugin)(const char ∗guid)
- DCInterfacePtr (DCAPI ∗query_interface)(const char ∗guid
- const char ∗DCAPI ∗ host_name ()
- intfHandle (DCAPI ∗register_interface)(const char ∗guid

Public Attributes

- uint32_t apiVersion
- dcptr_t pInterface
- uint32_t version

5.23.1 Detailed Description

Definition at line 295 of file PluginDefs.h.

5.23.2 Member Function Documentation

5.23.2.1 tagDCCore::Bool (DCAPI ∗ *release_interface*)

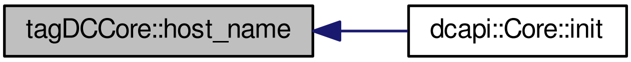
5.23.2.2 tagDCCore::Bool (DCAPI ∗ *has_plugin*) const

5.23.2.3 tagDCCore::DCInterfacePtr (DCAPI ∗ *query_interface*) const

5.23.2.4 const char∗ DCAPI∗ tagDCCore::host_name ( )

Referenced by dcapi::Core::init().

Here is the caller graph for this function:



5.23.2.5 tagDCCore::intfHandle (DCAPI ∗ *register_interface*) const

5.23.3 Member Data Documentation

5.23.3.1 uint32_t tagDCCore::apiVersion

Definition at line 297 of file PluginDefs.h.

5.23.3.2 dcptr_t tagDCCore::pInterface

Definition at line 300 of file PluginDefs.h.

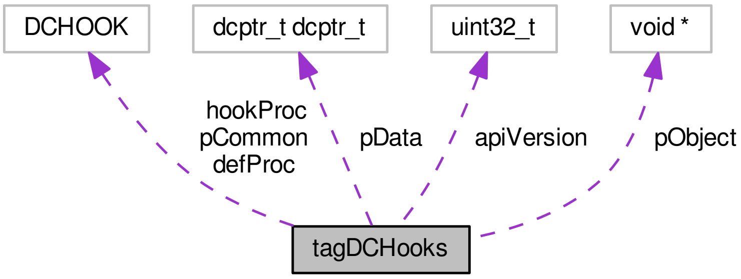
5.23.3.3 uint32_t tagDCCore::version

Definition at line 301 of file PluginDefs.h.

5.24 tagDCHooks Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagDCHooks:



Public Member Functions

- Bool (DCAPI ∗destroy_hook)(hookHandle hHook)
- Bool (DCAPI ∗run_hook)(hookHandle hHook
- hookHandle (DCAPI ∗create_hook)(const char ∗guid
- size_t (DCAPI ∗release_hook)(subsHandle hHook)
- subsHandle (DCAPI ∗bind_hook)(const char ∗guid

Public Attributes

- uint32_t apiVersion
- DCHOOK defProc
- DCHOOK hookProc
- DCHOOK void ∗ pCommon
- dcptr_t dcptr_t pData
- dcptr_t pObject

5.24.1 Detailed Description

Definition at line 320 of file PluginDefs.h.

5.24.2 Member Function Documentation

5.24.2.1 tagDCHooks::Bool (DCAPI ∗ *destroy_hook*)

5.24.2.2 tagDCHooks::Bool (DCAPI ∗ *run_hook*)

5.24.2.3 tagDCHooks::hookHandle (DCAPI ∗ *create_hook*) const

5.24.2.4 tagDCHooks::size_t (DCAPI ∗ *release_hook*)

5.24.2.5 tagDCHooks::subsHandle (DCAPI ∗ *bind_hook*) const

5.24.3 Member Data Documentation

5.24.3.1 uint32_t tagDCHooks::apiVersion

Definition at line 322 of file PluginDefs.h.

5.24.3.2 DCHOOK tagDCHooks::defProc

Definition at line 325 of file PluginDefs.h.

5.24.3.3 DCHOOK tagDCHooks::hookProc

Definition at line 329 of file PluginDefs.h.

5.24.3.4 DCHOOK void∗ tagDCHooks::pCommon

Definition at line 329 of file PluginDefs.h.

5.24.3.5 dcptr_t dcptr_t tagDCHooks::pData

Definition at line 330 of file PluginDefs.h.

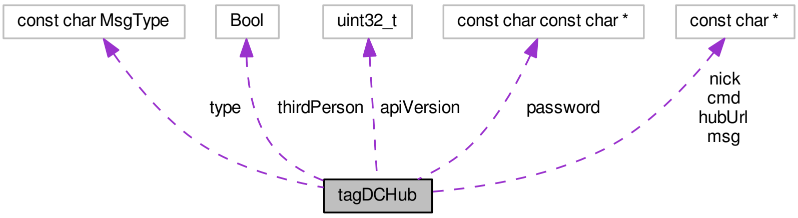
5.24.3.6 dcptr_t tagDCHooks::pObject

Definition at line 330 of file PluginDefs.h.

5.25 tagDCHub Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagDCHub:



Public Member Functions

- Bool (DCAPI ∗send_private_message)(UserDataPtr hUser
- HubDataPtr (DCAPI ∗add_hub)(const char ∗url
- HubDataPtr (DCAPI ∗find_hub)(const char ∗url)
- HubDataPtr (DCAPI ∗copy)(const HubDataPtr hHub)
- UserDataPtr (DCAPI ∗find_user)(const char ∗cid
- UserDataPtr (DCAPI ∗copy_user)(const UserDataPtr hUser)
- void (DCAPI ∗remove_hub)(HubDataPtr hHub)
- void (DCAPI ∗emulate_protocol_cmd)(HubDataPtr hHub
- void (DCAPI ∗send_protocol_cmd)(HubDataPtr hHub
- void (DCAPI ∗send_message)(HubDataPtr hHub
- void (DCAPI ∗local_message)(HubDataPtr hHub
- void (DCAPI ∗release_user)(UserDataPtr hCopy)
- void (DCAPI ∗release)(HubDataPtr hCopy)

Public Attributes

- uint32_t apiVersion
- const char ∗ cmd
- const char ∗ hubUrl
- const char ∗ msg
- const char ∗ nick
- const char const char ∗ password
- const char Bool thirdPerson
- const char MsgType type

5.25.1 Detailed Description

Definition at line 377 of file PluginDefs.h.

5.25.2 Member Function Documentation

5.25.2.1 tagDCHub::Bool (DCAPI ∗ *send_private_message*)

5.25.2.2 tagDCHub::HubDataPtr (DCAPI ∗ *add_hub*) const

5.25.2.3 tagDCHub::HubDataPtr (DCAPI ∗ *find_hub*) const

5.25.2.4 tagDCHub::HubDataPtr (DCAPI ∗ *copy*) const

5.25.2.5 tagDCHub::UserDataPtr (DCAPI ∗ *find_user*) const

5.25.2.6 tagDCHub::UserDataPtr (DCAPI ∗ *copy_user*) const

5.25.2.7 tagDCHub::void (DCAPI ∗ *remove_hub*)

5.25.2.8 tagDCHub::void (DCAPI ∗ *emulate_protocol_cmd*)

5.25.2.9 tagDCHub::void (DCAPI ∗ *send_protocol_cmd*)

5.25.2.10 tagDCHub::void (DCAPI ∗ *send_message*)

5.25.2.11 tagDCHub::void (DCAPI ∗ *local_message*)

5.25.2.12 tagDCHub::void (DCAPI ∗ *release_user*)

5.25.2.13 tagDCHub::void (DCAPI ∗ *release*)

5.25.3 Member Data Documentation

5.25.3.1 uint32_t tagDCHub::apiVersion

Definition at line 379 of file PluginDefs.h.

5.25.3.2 const char ∗ tagDCHub::cmd

Definition at line 385 of file PluginDefs.h.

5.25.3.3 const char∗ tagDCHub::hubUrl

Definition at line 392 of file PluginDefs.h.

5.25.3.4 const char ∗ tagDCHub::msg

Definition at line 388 of file PluginDefs.h.

5.25.3.5 const char∗ tagDCHub::nick

Definition at line 381 of file PluginDefs.h.

5.25.3.6 const char const char∗ tagDCHub::password

Definition at line 381 of file PluginDefs.h.

5.25.3.7 const char **Bool** tagDCHub::thirdPerson

Definition at line 388 of file PluginDefs.h.

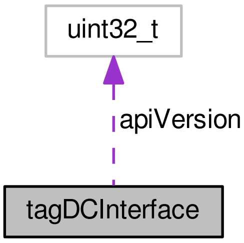
5.25.3.8 const char **MsgType** tagDCHub::type

Definition at line 389 of file PluginDefs.h.

5.26 tagDCInterface Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagDCInterface:



Public Attributes

- uint32_t apiVersion

5.26.1 Detailed Description

Definition at line 289 of file PluginDefs.h.

5.26.2 Member Data Documentation

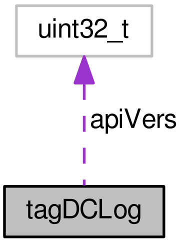
5.26.2.1 uint32_t tagDCInterface::apiVersion

Definition at line 291 of file PluginDefs.h.

5.27 tagDCLog Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagDCLog:



Public Member Functions

- void (DCAPI ∗log)(const char ∗msg)

Public Attributes

- uint32_t apiVersion

5.27.1 Detailed Description

Definition at line 355 of file PluginDefs.h.

5.27.2 Member Function Documentation

5.27.2.1 tagDCLog::void (DCAPI ∗ *log*) const

5.27.3 Member Data Documentation

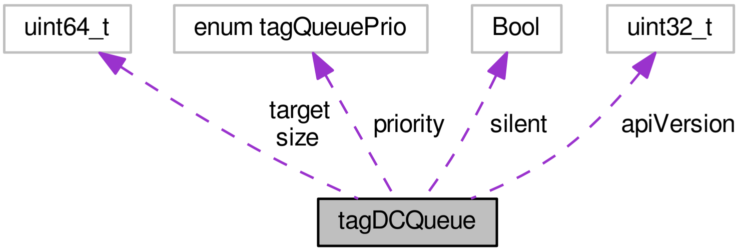
5.27.3.1 uint32_t tagDCLog::apiVersion

Definition at line 357 of file PluginDefs.h.

5.28 tagDCQueue Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagDCQueue:



Public Member Functions

- Bool (DCAPI ∗pause)(QueueDataPtr hItem)
- QueueDataPtr (DCAPI ∗add_list)(UserDataPtr hUser
- QueueDataPtr (DCAPI ∗add_download)(const char ∗hash
- QueueDataPtr (DCAPI ∗find_download)(const char ∗target)
- QueueDataPtr (DCAPI ∗copy)(const QueueDataPtr hItem)
- void (DCAPI ∗remove_download)(QueueDataPtr hItem)
- void (DCAPI ∗set_priority)(QueueDataPtr hItem
- void (DCAPI ∗release)(QueueDataPtr hCopy)

Public Attributes

- uint32_t apiVersion
- QueuePrio priority
- Bool silent
- uint64_t size
- uint64_t const char ∗ target

5.28.1 Detailed Description

Definition at line 401 of file PluginDefs.h.

5.28.2 Member Function Documentation

5.28.2.1 tagDCQueue::Bool (**DCAPI** ∗ *pause*)

5.28.2.2 tagDCQueue::QueueDataPtr (**DCAPI** ∗ *add_list*)

5.28.2.3 tagDCQueue::QueueDataPtr (**DCAPI** ∗ *add_download*) const

5.28.2.4 tagDCQueue::QueueDataPtr (**DCAPI** ∗ *find_download*) const

5.28.2.5 tagDCQueue::QueueDataPtr (**DCAPI** ∗ *copy*) const

5.28.2.6 tagDCQueue::void (DCAPI ∗ *remove_download*)

5.28.2.7 tagDCQueue::void (DCAPI ∗ *set_priority*)

5.28.2.8 tagDCQueue::void (DCAPI ∗ *release*)

5.28.3 Member Data Documentation

5.28.3.1 uint32_t tagDCQueue::apiVersion

Definition at line 403 of file PluginDefs.h.

5.28.3.2 QueuePrio tagDCQueue::priority

Definition at line 410 of file PluginDefs.h.

5.28.3.3 Bool tagDCQueue::silent

Definition at line 405 of file PluginDefs.h.

5.28.3.4 uint64_t tagDCQueue::size

Definition at line 406 of file PluginDefs.h.

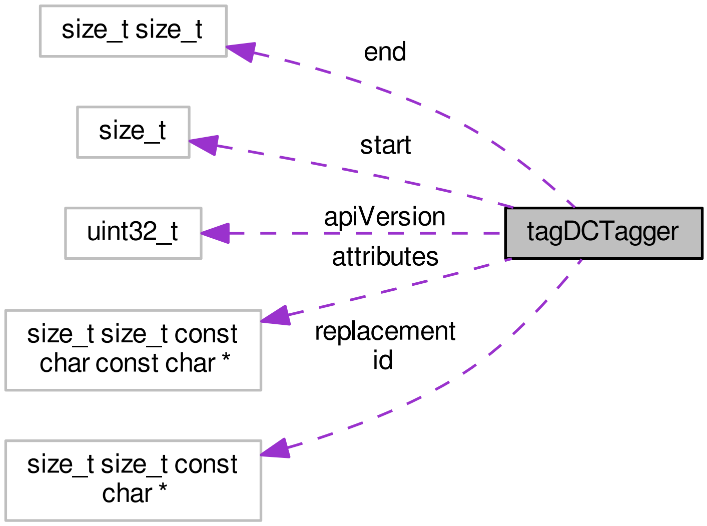
5.28.3.5 uint64_t const char∗ tagDCQueue::target

Definition at line 406 of file PluginDefs.h.

5.29 tagDCTagger Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagDCTagger:



Public Member Functions

- const char ∗DCAPI ∗ get_text (TagDataPtr hTags)
- void (DCAPI ∗add_tag)(TagDataPtr hTags
- void (DCAPI ∗replace_text)(TagDataPtr hTags

Public Attributes

- uint32_t apiVersion
- size_t size_t const char const char ∗ attributes
- size_t size_t end
- size_t size_t const char ∗ id
- size_t size_t const char ∗ replacement
- size_t start

5.29.1 Detailed Description

Definition at line 435 of file PluginDefs.h.

5.29.2 Member Function Documentation

5.29.2.1 const char∗ DCAPI∗ tagDCTagger::get_text (TagDataPtr *hTags*)

5.29.2.2 tagDCTagger::void (DCAPI ∗ *add_tag*)

5.29.2.3 tagDCTagger::void (DCAPI ∗ *replace_text*)

5.29.3 Member Data Documentation

5.29.3.1 uint32_t tagDCTagger::apiVersion

Definition at line 437 of file PluginDefs.h.

5.29.3.2 size_t size_t const char const char∗ tagDCTagger::attributes

Definition at line 441 of file PluginDefs.h.

5.29.3.3 size_t size_t tagDCTagger::end

Definition at line 441 of file PluginDefs.h.

5.29.3.4 size_t size_t const char∗ tagDCTagger::id

Definition at line 441 of file PluginDefs.h.

5.29.3.5 size_t size_t const char∗ tagDCTagger::replacement

Definition at line 442 of file PluginDefs.h.

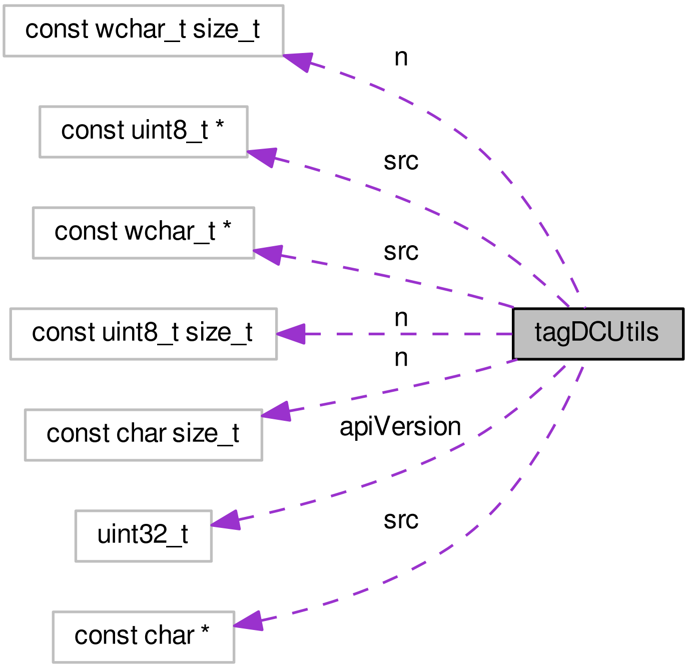
5.29.3.6 size_t tagDCTagger::start

Definition at line 441 of file PluginDefs.h.

5.30 tagDCUtils Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagDCUtils:



Public Member Functions

- size_t (DCAPI ∗to_utf8)(char ∗dst
- size_t (DCAPI ∗from_utf8)(char ∗dst
- size_t (DCAPI ∗utf8_to_wcs)(wchar_t ∗dst
- size_t (DCAPI ∗wcs_to_utf8)(char ∗dst
- size_t (DCAPI ∗to_base32)(char ∗dst
- size_t (DCAPI ∗from_base32)(uint8_t ∗dst

Public Attributes

- uint32_t apiVersion
- const char size_t n
- const wchar_t size_t n
- const uint8_t size_t n
- const char ∗ src
- const wchar_t ∗ src
- const uint8_t ∗ src

5.30.1 Detailed Description

Definition at line 418 of file PluginDefs.h.

5.30.2 Member Function Documentation

5.30.2.1 tagDCUtils::size_t (DCAPI ∗ *to_utf8*)

5.30.2.2 tagDCUtils::size_t (DCAPI ∗ *from_utf8*)

5.30.2.3 tagDCUtils::size_t (DCAPI ∗ *utf8_to_wcs*)

5.30.2.4 tagDCUtils::size_t (DCAPI ∗ *wcs_to_utf8*)

5.30.2.5 tagDCUtils::size_t (DCAPI ∗ *to_base32*)

5.30.2.6 tagDCUtils::size_t (DCAPI ∗ *from_base32*)

5.30.3 Member Data Documentation

5.30.3.1 uint32_t tagDCUtils::apiVersion

Definition at line 420 of file PluginDefs.h.

5.30.3.2 const char size_t tagDCUtils::n

Definition at line 424 of file PluginDefs.h.

5.30.3.3 const wchar_t size_t tagDCUtils::n

Definition at line 428 of file PluginDefs.h.

5.30.3.4 const uint8_t size_t tagDCUtils::n

Definition at line 430 of file PluginDefs.h.

5.30.3.5 const char ∗ tagDCUtils::src

Definition at line 424 of file PluginDefs.h.

5.30.3.6 const wchar_t∗ tagDCUtils::src

Definition at line 428 of file PluginDefs.h.

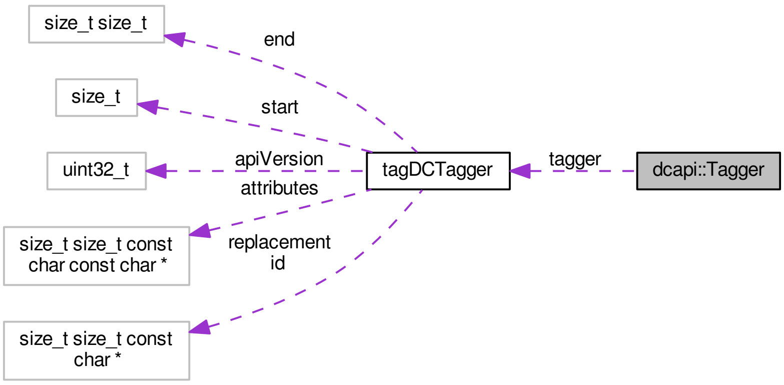
5.30.3.7 const uint8_t∗ tagDCUtils::src

Definition at line 430 of file PluginDefs.h.

5.31 dcapi::Tagger Class Reference

```
#include <Tagger.hpp>
```

Collaboration diagram for dcapi::Tagger:



Static Public Member Functions

- static DCTaggerPtr handle ()
- static bool init ()
- static void init (DCTaggerPtr coreTagger)

Static Private Attributes

- static DCTaggerPtr tagger

5.31.1 Detailed Description

Definition at line 28 of file Tagger.hpp.

5.31.2 Member Function Documentation

5.31.2.1 DCTaggerPtr dcapi::Tagger::handle ( ) [static]

Definition at line 35 of file Tagger.cpp.

References tagger.

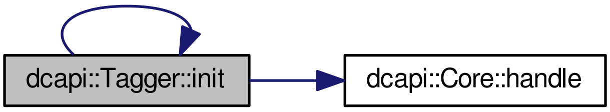
5.31.2.2 bool dcapi::Tagger::init ( ) [static]

Definition at line 29 of file Tagger.cpp.

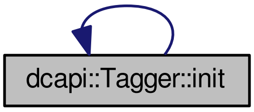
References DCINTF_DCPP_TAGGER, DCINTF_DCPP_TAGGER_VER, dcapi::Core::handle(), init(), and tagger.

Referenced by init().

Here is the call graph for this function:



Here is the caller graph for this function:



5.31.2.3 void dcapi::Tagger::init (DCTaggerPtr *coreTagger*) `[static]`

Definition at line 34 of file Tagger.cpp.

References tagger.

5.31.3 Member Data Documentation

5.31.3.1 DCTaggerPtr dcapi::Tagger::tagger `[static]`, `[private]`

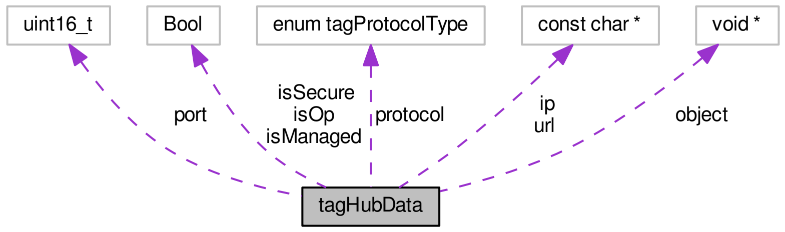
Definition at line 36 of file Tagger.hpp.

Referenced by handle(), and init().

5.32 tagHubData Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagHubData:



Public Attributes

- const char ∗ ip
- Bool isManaged
- Bool isOp
- Bool isSecure
- dcptr_t object
- uint16_t port
- ProtocolType protocol
- const char ∗ url

5.32.1 Detailed Description

Definition at line 230 of file PluginDefs.h.

5.32.2 Member Data Documentation

5.32.2.1 const char∗ tagHubData::ip

Definition at line 232 of file PluginDefs.h.

5.32.2.2 Bool tagHubData::isManaged

Definition at line 238 of file PluginDefs.h.

5.32.2.3 Bool tagHubData::isOp

Definition at line 236 of file PluginDefs.h.

5.32.2.4 Bool tagHubData::isSecure

Definition at line 237 of file PluginDefs.h.

5.32.2.5 dcptr_t tagHubData::object

Definition at line 233 of file PluginDefs.h.

5.32.2.6 uint16_t tagHubData::port

Definition at line 234 of file PluginDefs.h.

5.32.2.7 ProtocolType tagHubData::protocol

Definition at line 235 of file PluginDefs.h.

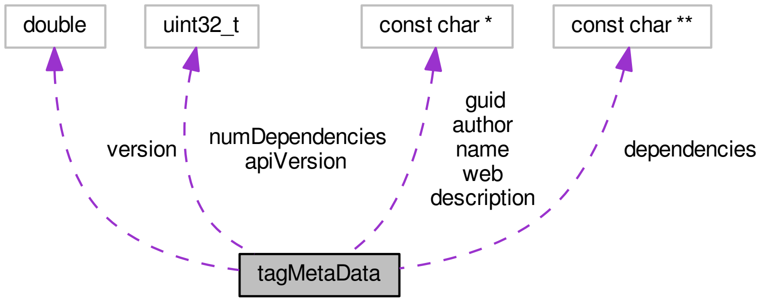
5.32.2.8 const char∗ tagHubData::url

Definition at line 231 of file PluginDefs.h.

5.33 tagMetaData Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagMetaData:



Public Attributes

- uint32_t apiVersion
- const char ∗ author
- const char ∗∗ dependencies
- const char ∗ description
- const char ∗ guid
- const char ∗ name
- uint32_t numDependencies
- double version
- const char ∗ web

5.33.1 Detailed Description

Definition at line 276 of file PluginDefs.h.

5.33.2 Member Data Documentation

5.33.2.1 uint32_t tagMetaData::apiVersion

Definition at line 284 of file PluginDefs.h.

Referenced by pluginInit().

5.33.2.2 const char∗ tagMetaData::author

Definition at line 278 of file PluginDefs.h.

Referenced by pluginInit().

5.33.2.3 const char∗∗ tagMetaData::dependencies

Definition at line 282 of file PluginDefs.h.

5.33.2.4 const char∗ tagMetaData::description

Definition at line 279 of file PluginDefs.h.

Referenced by pluginInit().

5.33.2.5 const char∗ tagMetaData::guid

Definition at line 281 of file PluginDefs.h.

Referenced by pluginInit().

5.33.2.6 const char∗ tagMetaData::name

Definition at line 277 of file PluginDefs.h.

Referenced by pluginInit().

5.33.2.7 uint32_t tagMetaData::numDependencies

Definition at line 283 of file PluginDefs.h.

5.33.2.8 double tagMetaData::version

Definition at line 285 of file PluginDefs.h.

Referenced by pluginInit().

5.33.2.9 const char∗ tagMetaData::web

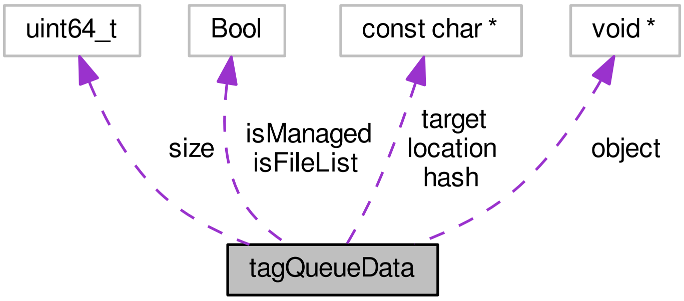
Definition at line 280 of file PluginDefs.h.

Referenced by pluginInit().

5.34 tagQueueData Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagQueueData:



Public Attributes

- const char ∗ hash
- Bool isFileList
- Bool isManaged
- const char ∗ location
- dcptr_t object
- uint64_t size
- const char ∗ target

5.34.1 Detailed Description

Definition at line 259 of file PluginDefs.h.

5.34.2 Member Data Documentation

5.34.2.1 const char∗ tagQueueData::hash

Definition at line 262 of file PluginDefs.h.

5.34.2.2 Bool tagQueueData::isFileList

Definition at line 265 of file PluginDefs.h.

5.34.2.3 Bool tagQueueData::isManaged

Definition at line 266 of file PluginDefs.h.

5.34.2.4 const char∗ tagQueueData::location

Definition at line 261 of file PluginDefs.h.

5.34.2.5 dcptr_t tagQueueData::object

Definition at line 263 of file PluginDefs.h.

5.34.2.6 uint64_t tagQueueData::size

Definition at line 264 of file PluginDefs.h.

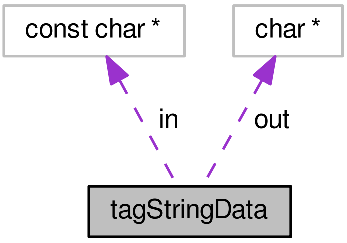
5.34.2.7 const char∗ tagQueueData::target

Definition at line 260 of file PluginDefs.h.

5.35 tagStringData Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagStringData:



Public Attributes

- const char ∗ in
- char ∗ out

5.35.1 Detailed Description

Definition at line 206 of file PluginDefs.h.

5.35.2 Member Data Documentation

5.35.2.1 const char∗ tagStringData::in

Definition at line 207 of file PluginDefs.h.

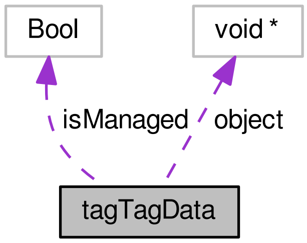
5.35.2.2 char∗ tagStringData::out

Definition at line 208 of file PluginDefs.h.

5.36 tagTagData Struct Reference

`#include <PluginDefs.h>`

Collaboration diagram for tagTagData:



Public Attributes

- Bool isManaged
- dcptr_t object

5.36.1 Detailed Description

Definition at line 270 of file PluginDefs.h.

5.36.2 Member Data Documentation

5.36.2.1 Bool tagTagData::isManaged

Definition at line 272 of file PluginDefs.h.

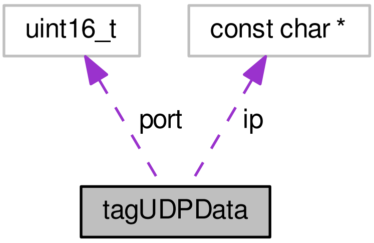
5.36.2.2 dcptr_t tagTagData::object

Definition at line 271 of file PluginDefs.h.

5.37 tagUDPData Struct Reference

`#include <PluginDefs.h>`

Collaboration diagram for tagUDPData:



Public Attributes

- const char ∗ ip
- uint16_t port

5.37.1 Detailed Description

Definition at line 253 of file PluginDefs.h.

5.37.2 Member Data Documentation

5.37.2.1 const char∗ tagUDPData::ip

Definition at line 254 of file PluginDefs.h.

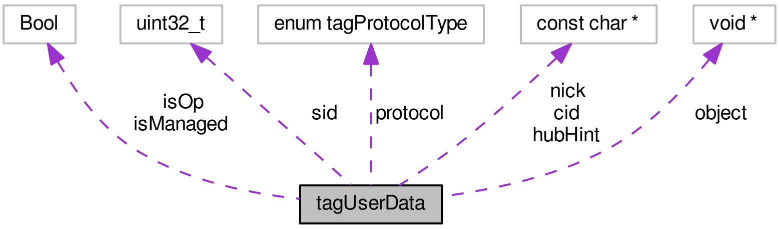
5.37.2.2 uint16_t tagUDPData::port

Definition at line 255 of file PluginDefs.h.

5.38 tagUserData Struct Reference

```
#include <PluginDefs.h>
```

Collaboration diagram for tagUserData:



Public Attributes

- const char ∗ cid
- const char ∗ hubHint
- Bool isManaged
- Bool isOp
- const char ∗ nick
- dcptr_t object
- ProtocolType protocol
- uint32_t sid

5.38.1 Detailed Description

Definition at line 218 of file PluginDefs.h.

5.38.2 Member Data Documentation

5.38.2.1 const char∗ tagUserData::cid

Definition at line 221 of file PluginDefs.h.

5.38.2.2 const char∗ tagUserData::hubHint

Definition at line 220 of file PluginDefs.h.

5.38.2.3 Bool tagUserData::isManaged

Definition at line 226 of file PluginDefs.h.

5.38.2.4 Bool tagUserData::isOp

Definition at line 225 of file PluginDefs.h.

5.38.2.5 const char∗ tagUserData::nick

Definition at line 219 of file PluginDefs.h.

5.38.2.6 dcptr_t tagUserData::object

Definition at line 222 of file PluginDefs.h.

5.38.2.7 ProtocolType tagUserData::protocol

Definition at line 224 of file PluginDefs.h.

5.38.2.8 uint32_t tagUserData::sid

Definition at line 223 of file PluginDefs.h.

5.39 dcapi::Hooks::Timer Struct Reference

```
#include <Hooks.hpp>
```

Static Public Member Functions

- static void onMinute (function< bool(uint64_t, bool &)> f)
- static void onSecond (function< bool(uint64_t, bool &)> f)

5.39.1 Detailed Description

Definition at line 61 of file Hooks.hpp.

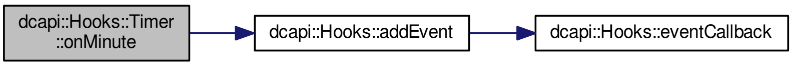
5.39.2 Member Function Documentation

5.39.2.1 void dcapi::Hooks::Timer::onMinute (function< bool(uint64_t, bool &)> *f*) [static]

Definition at line 59 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_TIMER_MINUTE.

Here is the call graph for this function:

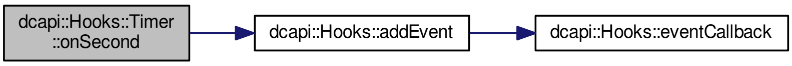


5.39.2.2 void dcapi::Hooks::Timer::onSecond (function< bool(uint64_t, bool &)> *f*) [static]

Definition at line 55 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_TIMER_SECOND.

Here is the call graph for this function:



5.40 dcapi::Hooks::UI Struct Reference

```
#include <Hooks.hpp>
```

Static Public Member Functions

- static void onChatCommand (function< bool(HubDataPtr, CommandDataPtr, bool &)> f)
- static void onChatCommandPM (function< bool(UserDataPtr, CommandDataPtr, bool &)> f)
- static void onChatDisplay (function< bool(UserDataPtr, StringDataPtr, bool &)> f)
- static void onChatTags (function< bool(UserDataPtr, TagDataPtr, bool &)> f)
- static void onCreated (function< bool(dcptr_t, bool &)> f)

5.40.1 Detailed Description

Definition at line 92 of file Hooks.hpp.

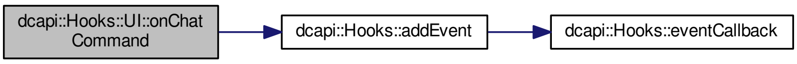
5.40.2 Member Function Documentation

5.40.2.1 void dcapi::Hooks::UI::onChatCommand (function< bool(HubDataPtr, CommandDataPtr, bool &)> *f*) [static]

Definition at line 136 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_UI_CHAT_COMMAND.

Here is the call graph for this function:

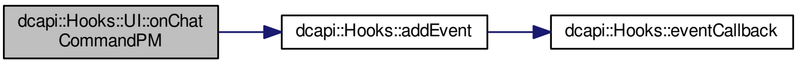


5.40.2.2 void dcapi::Hooks::UI::onChatCommandPM (function< bool(UserDataPtr, CommandDataPtr, bool &)> *f*) [static]

Definition at line 140 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_UI_CHAT_COMMAND_PM.

Here is the call graph for this function:

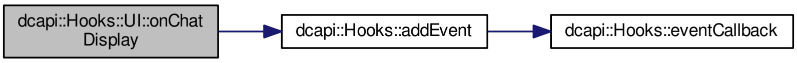

dcapi::Hooks::UI::onChat CommandPM
dcapi::Hooks::addEvent
dcapi::Hooks::eventCallback

5.40.2.3 void dcapi::Hooks::UI::onChatDisplay (function< bool(UserDataPtr, StringDataPtr, bool &)> *f*) [static]

Definition at line 132 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_UI_CHAT_DISPLAY.

Here is the call graph for this function:

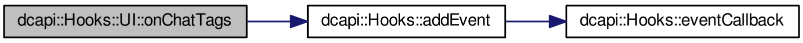


5.40.2.4 void dcapi::Hooks::UI::onChatTags (function< bool(UserDataPtr, TagDataPtr, bool &)> *f*) [static]

Definition at line 128 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_UI_CHAT_TAGS.

Here is the call graph for this function:

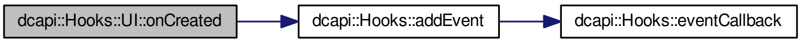


5.40.2.5 void dcapi::Hooks::UI::onCreated (function< bool(dcptr_t, bool &)> *f*) [static]

Definition at line 124 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_UI_CREATED.

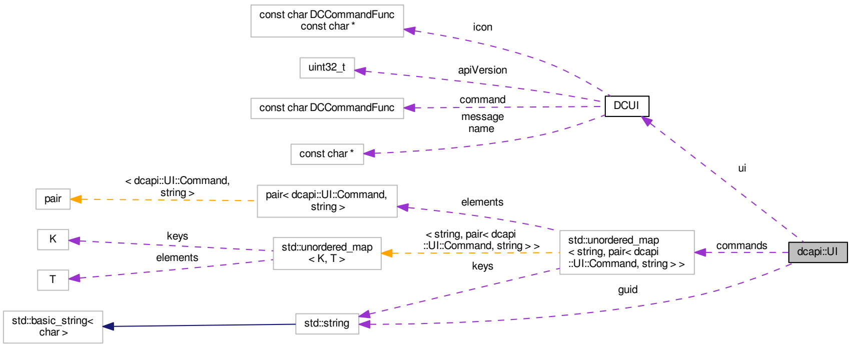
Here is the call graph for this function:



5.41 dcapi::UI Class Reference

```
#include <UI.hpp>
```

Collaboration diagram for dcapi::UI:



Public Types

- typedef function< void()> Command

Static Public Member Functions

- static void addCommand (string name, Command command, string icon)
- static void clearCommands ()
- static DCUIPtr handle ()
- static bool init (string pluginGuid)
- static void init (DCUIPtr coreUI, string pluginGuid)
- static void removeCommand (const string &name)

Static Private Member Functions

- static void DCAPI commandCallback (const char ∗name)

Static Private Attributes

- static unordered_map< string,
  pair< UI::Command, string > > commands
- static string guid
- static DCUIPtr ui

5.41.1 Detailed Description

Definition at line 37 of file UI.hpp.

5.41.2 Member Typedef Documentation

5.41.2.1 typedef function<void ()> dcapi::UI::Command

Definition at line 44 of file UI.hpp.

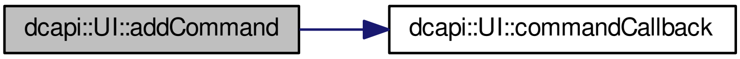
5.41.3 Member Function Documentation

5.41.3.1 void dcapi::UI::addCommand (string *name,* Command *command,* string *icon*) [static]

Definition at line 39 of file UI.cpp.

References commandCallback(), commands, guid, and ui.

Here is the call graph for this function:



5.41.3.2 void dcapi::UI::clearCommands ( ) [static]

Definition at line 49 of file UI.cpp.

References commands, guid, and ui.

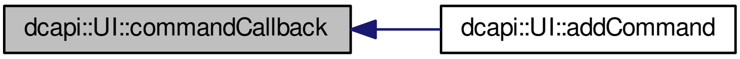
5.41.3.3 void DCAPI dcapi::UI::commandCallback (const char ∗ *name*) [static], [private]

Definition at line 56 of file UI.cpp.

References commands.

Referenced by addCommand().

Here is the caller graph for this function:



5.41.3.4 DCUIPtr dcapi::UI::handle ( ) [static]

Definition at line 37 of file UI.cpp.

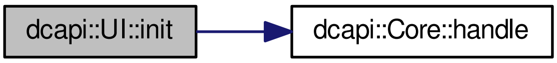
References ui.

5.41.3.5 bool dcapi::UI::init (string *pluginGuid*) [static]

Definition at line 31 of file UI.cpp.

References DCINTF_DCPP_UI, DCINTF_DCPP_UI_VER, dcapi::Core::handle(), and ui.

Here is the call graph for this function:



5.41.3.6 void dcapi::UI::init (DCUIPtr *coreUI,* string *pluginGuid*) [static]

Definition at line 36 of file UI.cpp.

References guid, and ui.

5.41.3.7 void dcapi::UI::removeCommand (const string & *name*) [static]

Definition at line 44 of file UI.cpp.

References commands, guid, and ui.

5.41.4 Member Data Documentation

5.41.4.1 unordered_map< string, pair< UI::Command, string > > dcapi::UI::commands [static], [private]

Definition at line 57 of file UI.hpp.

Referenced by addCommand(), clearCommands(), commandCallback(), and removeCommand().

5.41.4.2 string dcapi::UI::guid [static], [private]

Definition at line 54 of file UI.hpp.

Referenced by addCommand(), clearCommands(), init(), and removeCommand().

5.41.4.3 DCUIPtr dcapi::UI::ui [static], [private]

Definition at line 52 of file UI.hpp.

Referenced by addCommand(), clearCommands(), handle(), init(), and removeCommand().

5.42 dcapi::Hooks::Users Struct Reference

```
#include <Hooks.hpp>
```

Static Public Member Functions

- static void onOffline (function< bool(UserDataPtr, bool &)> f)
- static void onOnline (function< bool(UserDataPtr, bool &)> f)

5.42.1 Detailed Description

Definition at line 71 of file Hooks.hpp.

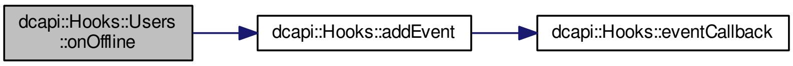
5.42.2 Member Function Documentation

5.42.2.1 void dcapi::Hooks::Users::onOffline (function< bool(UserDataPtr, bool &)> *f*) [static]

Definition at line 77 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_USER_OFFLINE.

Here is the call graph for this function:

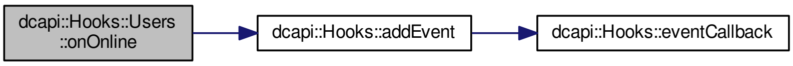


5.42.2.2 void dcapi::Hooks::Users::onOnline (function< bool(UserDataPtr, bool &)> *f*) [static]

Definition at line 73 of file Hooks.cpp.

References dcapi::Hooks::addEvent(), and HOOK_USER_ONLINE.

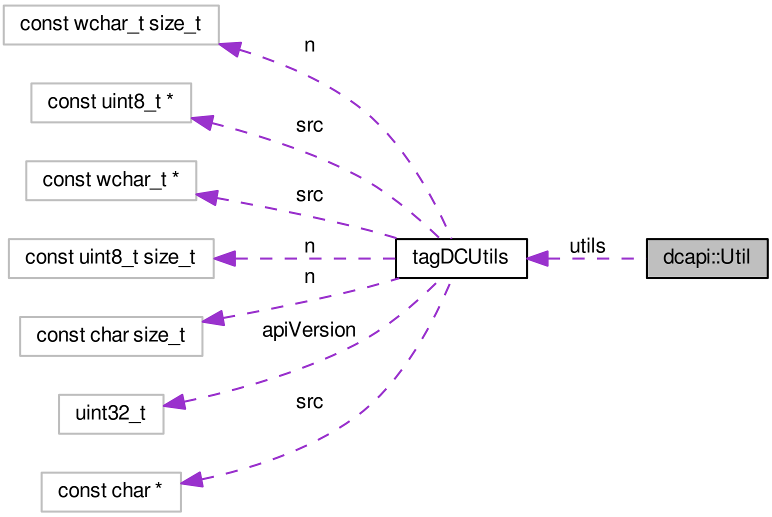
Here is the call graph for this function:



5.43 dcapi::Util Class Reference

```
#include <Util.hpp>
```

Collaboration diagram for dcapi::Util:



Static Public Member Functions

- static string fromUtf8 (const string &str)
- static DCUtilsPtr handle ()
- static bool init ()
- static void init (DCUtilsPtr coreUtils)
- static string toUtf8 (const string &str)

Static Private Attributes

- static DCUtilsPtr utils

5.43.1 Detailed Description

Definition at line 35 of file Util.hpp.

5.43.2 Member Function Documentation

5.43.2.1 string dcapi::Util::fromUtf8 (const string & *str*) [static]

Definition at line 68 of file Util.cpp.

References utils.

5.43.2.2 DCUtilsPtr dcapi::Util::handle ( ) [static]

Definition at line 35 of file Util.cpp.

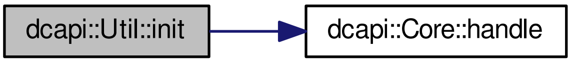
References utils.

5.43.2.3 bool dcapi::Util::init ( ) [static]

Definition at line 29 of file Util.cpp.

References DCINTF_DCPP_UTILS, DCINTF_DCPP_UTILS_VER, dcapi::Core::handle(), and utils.

Here is the call graph for this function:



5.43.2.4 void dcapi::Util::init (DCUtilsPtr *coreUtils*) [static]

Definition at line 34 of file Util.cpp.

References utils.

5.43.2.5 string dcapi::Util::toUtf8 (const string & *str*) [static]

Definition at line 81 of file Util.cpp.

References utils.

5.43.3 Member Data Documentation

5.43.3.1 DCUtilsPtr dcapi::Util::utils [static], [private]

Definition at line 55 of file Util.hpp.

Referenced by fromUtf8(), handle(), init(), and toUtf8().

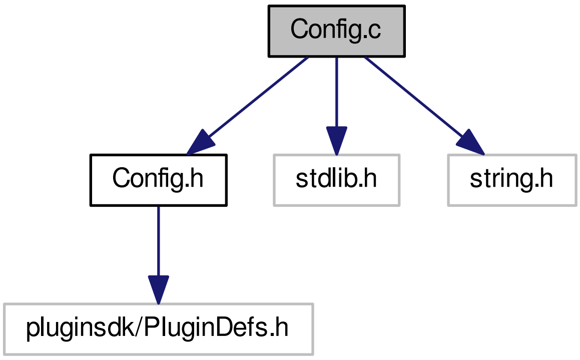
Chapter 6

File Documentation

6.1 Config.c File Reference

```
#include "Config.h"
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for Config.c:



Functions

- void DCAPI free_core_cfg (ConfigValuePtr val)
- char ∗DCAPI get_cfg (const char ∗name)
- int32_t DCAPI get_cfg_int (const char ∗name)
- int64_t DCAPI get_cfg_int64 (const char ∗name)
- char ∗DCAPI get_cfg_path (PathType path)
- ConfigValuePtr DCAPI get_core_cfg (const char ∗name)
- char ∗DCAPI get_install_path ()
- Bool DCAPI init_cfg (DCCorePtr core, const char ∗pluginGuid)
- void DCAPI set_cfg (const char ∗name, const char ∗value)
- void DCAPI set_cfg_int (const char ∗name, int32_t value)
- void DCAPI set_cfg_int64 (const char ∗name, int64_t value)
- void DCAPI uninit_cfg ()

Variables

- DCConfigPtr config
- char ∗ guid

6.1.1 Function Documentation

6.1.1.1 void DCAPI free_core_cfg (ConfigValuePtr *val*)

Definition at line 100 of file Config.c.

6.1.1.2 char∗ DCAPI get_cfg (const char ∗ *name*)

Definition at line 43 of file Config.c.

References CFG_TYPE_STRING, guid, and tagConfigStr::value.

6.1.1.3 int32_t DCAPI get_cfg_int (const char ∗ *name*)

Definition at line 54 of file Config.c.

References CFG_TYPE_INT, guid, and tagConfigInt::value.

6.1.1.4 int64_t DCAPI get_cfg_int64 (const char ∗ *name*)

Definition at line 63 of file Config.c.

References CFG_TYPE_INT64, guid, and tagConfigInt64::value.

6.1.1.5 char∗ DCAPI get_cfg_path (PathType *path*)

Definition at line 102 of file Config.c.

References tagConfigStr::value.

6.1.1.6 ConfigValuePtr DCAPI get_core_cfg (const char ∗ *name*)

Definition at line 99 of file Config.c.

References CFG_TYPE_UNKNOWN.

6.1.1.7 char∗ DCAPI get_install_path ( )

Definition at line 113 of file Config.c.

References guid, and tagConfigStr::value.

6.1.1.8 Bool DCAPI init_cfg (DCCorePtr *core,* const char ∗ *pluginGuid*)

Definition at line 29 of file Config.c.

References DCINTF_CONFIG, DCINTF_CONFIG_VER, False, guid, and True.

6.1.1.9 void DCAPI set_cfg (const char * *name,* const char * *value*)

Definition at line 72 of file Config.c.

References CFG_TYPE_STRING, guid, tagConfigStr::type, and tagConfigStr::value.

6.1.1.10 void DCAPI set_cfg_int (const char * *name,* int32_t *value*)

Definition at line 81 of file Config.c.

References CFG_TYPE_INT, guid, tagConfigInt::type, and tagConfigInt::value.

6.1.1.11 void DCAPI set_cfg_int64 (const char * *name,* int64_t *value*)

Definition at line 90 of file Config.c.

References CFG_TYPE_INT64, guid, tagConfigInt64::type, and tagConfigInt64::value.

6.1.1.12 void DCAPI uninit_cfg ( )

Definition at line 39 of file Config.c.

References guid.

6.1.2 Variable Documentation

6.1.2.1 DCConfigPtr config

Definition at line 26 of file Config.c.

6.1.2.2 char* guid

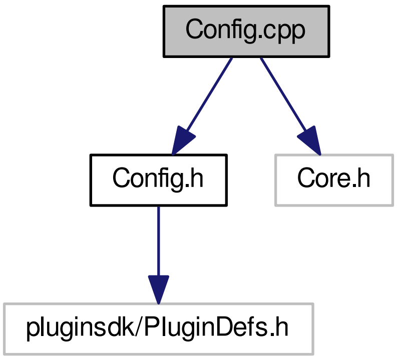
Definition at line 27 of file Config.c.

Referenced by get_cfg(), get_cfg_int(), get_cfg_int64(), get_install_path(), init_cfg(), set_cfg(), set_cfg_int(), set_cfg_int64(), and uninit_cfg().

6.2 Config.cpp File Reference

```
#include "Config.h"
#include "Core.h"
```

Include dependency graph for Config.cpp:



Namespaces

- dcapi

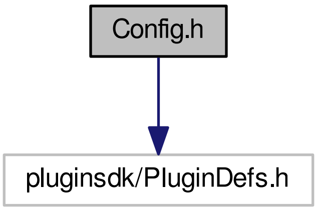
Constant Groups

- dcapi

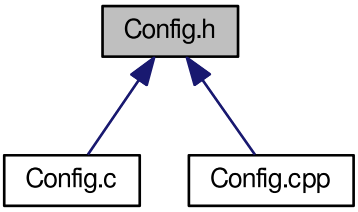
6.3 Config.h File Reference

```
#include <pluginsdk/PluginDefs.h>
```

Include dependency graph for Config.h:



This graph shows which files directly or indirectly include this file:



Functions

- void DCAPI free_core_cfg (ConfigValuePtr val)
- char ∗DCAPI get_cfg (const char ∗name)
- int32_t DCAPI get_cfg_int (const char ∗name)
- int64_t DCAPI get_cfg_int64 (const char ∗name)
- char ∗DCAPI get_cfg_path (PathType path)
- ConfigValuePtr DCAPI get_core_cfg (const char ∗name)
- char ∗DCAPI get_install_path ()
- Bool DCAPI init_cfg (DCCorePtr core, const char ∗pluginGuid)
- void DCAPI set_cfg (const char ∗name, const char ∗value)
- void DCAPI set_cfg_int (const char ∗name, int32_t value)
- void DCAPI set_cfg_int64 (const char ∗name, int64_t value)
- void DCAPI uninit_cfg ()

6.3.1 Function Documentation

6.3.1.1 void DCAPI free_core_cfg (ConfigValuePtr *val*)

Definition at line 100 of file Config.c.

6.3.1.2 char∗ DCAPI get_cfg (const char ∗ *name*)

Definition at line 43 of file Config.c.

References CFG_TYPE_STRING, guid, and tagConfigStr::value.

6.3.1.3 int32_t DCAPI get_cfg_int (const char ∗ *name*)

Definition at line 54 of file Config.c.

References CFG_TYPE_INT, guid, and tagConfigInt::value.

6.3.1.4 int64_t DCAPI get_cfg_int64 (const char ∗ *name*)

Definition at line 63 of file Config.c.

References CFG_TYPE_INT64, guid, and tagConfigInt64::value.

6.3.1.5 char∗ DCAPI get_cfg_path (PathType *path*)

Definition at line 102 of file Config.c.

References tagConfigStr::value.

6.3.1.6 ConfigValuePtr DCAPI get_core_cfg (const char ∗ *name*)

Definition at line 99 of file Config.c.

References CFG_TYPE_UNKNOWN.

6.3.1.7 char∗ DCAPI get_install_path ( )

Definition at line 113 of file Config.c.

References guid, and tagConfigStr::value.

6.3.1.8 Bool DCAPI init_cfg (DCCorePtr *core,* const char ∗ *pluginGuid*)

Definition at line 29 of file Config.c.

References DCINTF_CONFIG, DCINTF_CONFIG_VER, False, guid, and True.

6.3.1.9 void DCAPI set_cfg (const char ∗ *name,* const char ∗ *value*)

Definition at line 72 of file Config.c.

References CFG_TYPE_STRING, guid, tagConfigStr::type, and tagConfigStr::value.

6.3.1.10 void DCAPI set_cfg_int (const char ∗ *name,* int32_t *value*)

Definition at line 81 of file Config.c.

References CFG_TYPE_INT, guid, tagConfigInt::type, and tagConfigInt::value.

6.3.1.11 void DCAPI set_cfg_int64 (const char ∗ *name,* int64_t *value*)

Definition at line 90 of file Config.c.

References CFG_TYPE_INT64, guid, tagConfigInt64::type, and tagConfigInt64::value.

6.3.1.12 void DCAPI uninit_cfg ( )

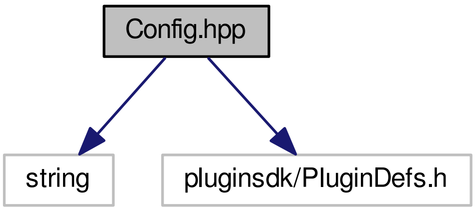
Definition at line 39 of file Config.c.

References guid.

6.4 Config.hpp File Reference

```
#include <string>
#include <pluginsdk/PluginDefs.h>
```

Include dependency graph for Config.hpp:



Classes

- class dcapi::Config

Namespaces

- dcapi

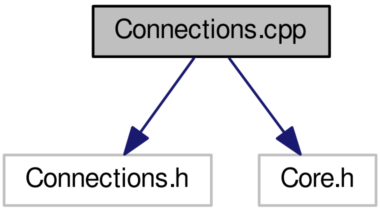
Constant Groups

- dcapi

6.5 Connections.cpp File Reference

```
#include "Connections.h"
#include "Core.h"
```

Include dependency graph for Connections.cpp:



Namespaces

- dcapi

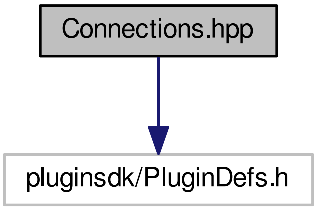
Constant Groups

- dcapi

6.6 Connections.hpp File Reference

```
#include <pluginsdk/PluginDefs.h>
```

Include dependency graph for Connections.hpp:



Classes

- class dcapi::Connections

Namespaces

- dcapi

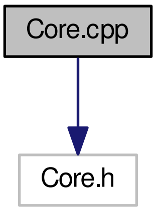
Constant Groups

- dcapi

6.7 Core.cpp File Reference

```
#include "Core.h"
```

Include dependency graph for Core.cpp:



Namespaces

- dcapi

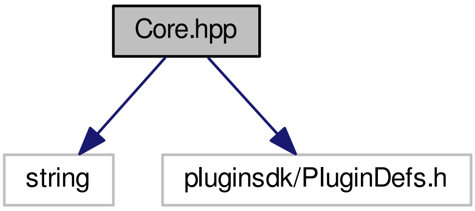
Constant Groups

- dcapi

6.8 Core.hpp File Reference

```
#include <string>
#include <pluginsdk/PluginDefs.h>
```

Include dependency graph for Core.hpp:



Classes

- class dcapi::Core

Namespaces

- dcapi

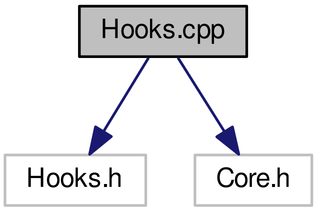
Constant Groups

- dcapi

6.9 Hooks.cpp File Reference

```
#include "Hooks.h"
#include "Core.h"
```

Include dependency graph for Hooks.cpp:



Namespaces

- dcapi

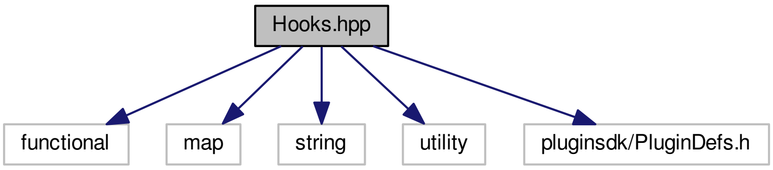
Constant Groups

- dcapi

6.10 Hooks.hpp File Reference

```
#include <functional>
#include <map>
#include <string>
#include <utility>
#include <pluginsdk/PluginDefs.h>
```

Include dependency graph for Hooks.hpp:



Classes

- struct dcapi::Hooks::Chat
- class dcapi::Hooks
- struct dcapi::Hooks::Hubs
- struct dcapi::Hooks::Network
- struct dcapi::Hooks::Queue
- struct dcapi::Hooks::Timer
- struct dcapi::Hooks::UI
- struct dcapi::Hooks::Users

Namespaces

- dcapi

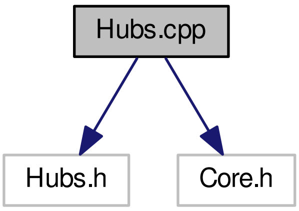
Constant Groups

- dcapi

6.11 Hubs.cpp File Reference

```
#include "Hubs.h"
#include "Core.h"
```

Include dependency graph for Hubs.cpp:



Namespaces

- dcapi

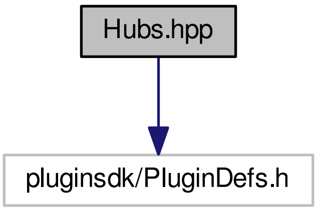
Constant Groups

- dcapi

6.12 Hubs.hpp File Reference

```
#include <pluginsdk/PluginDefs.h>
```

Include dependency graph for Hubs.hpp:



Classes

- class dcapi::Hubs

Namespaces

- dcapi

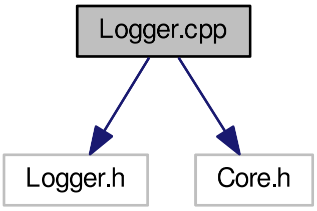
Constant Groups

- dcapi

6.13 Logger.cpp File Reference

```
#include "Logger.h"
#include "Core.h"
```

Include dependency graph for Logger.cpp:



Namespaces

- dcapi

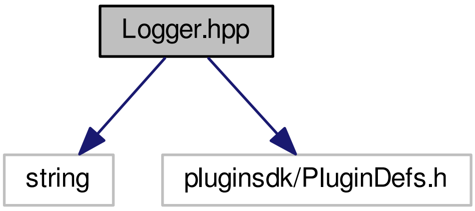
Constant Groups

- dcapi

6.14 Logger.hpp File Reference

```
#include <string>
#include <pluginsdk/PluginDefs.h>
```

Include dependency graph for Logger.hpp:



Classes

- class dcapi::Logger

Namespaces

- dcapi

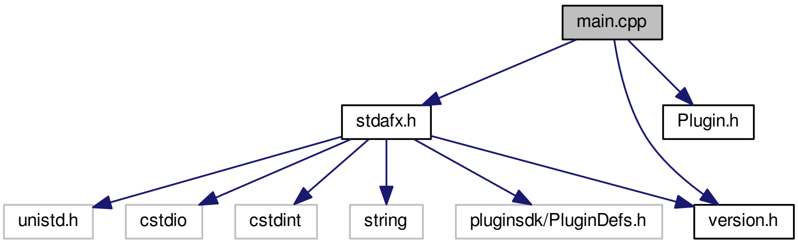
Constant Groups

- dcapi

6.15 main.cpp File Reference

```
#include "stdafx.h"
#include "Plugin.h"
#include "version.h"
```

Include dependency graph for main.cpp:



Functions

- DCEXP DCMAIN DCAPI pluginInit (MetaDataPtr info)

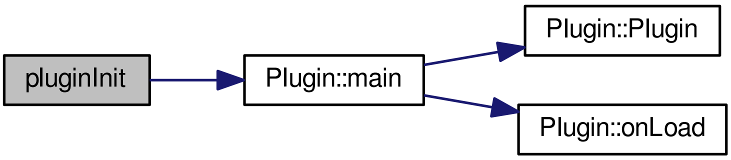
6.15.1 Function Documentation

6.15.1.1 DCEXP DCMAIN DCAPI pluginInit (MetaDataPtr *info*)

Definition at line 26 of file main.cpp.

References tagMetaData::apiVersion, tagMetaData::author, DCAPI_CORE_VER, tagMetaData::description, tagMetaData::guid, Plugin::main(), tagMetaData::name, PLUGIN_AUTHOR, PLUGIN_DESC, PLUGIN_GUID, PLUGIN_NAME, PLUGIN_VERSION, PLUGIN_WEB, tagMetaData::version, and tagMetaData::web.

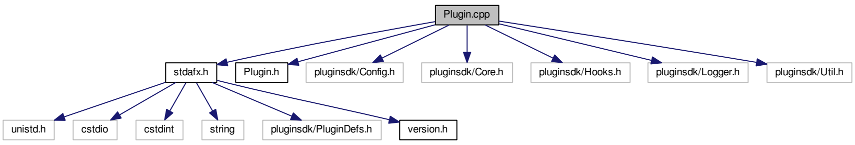
Here is the call graph for this function:



6.16 Plugin.cpp File Reference

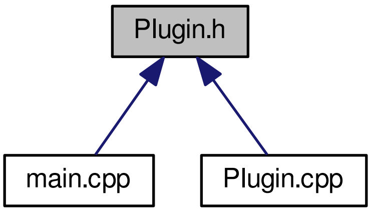
```
#include "stdafx.h"
#include "Plugin.h"
#include <pluginsdk/Config.h>
#include <pluginsdk/Core.h>
#include <pluginsdk/Hooks.h>
#include <pluginsdk/Logger.h>
#include <pluginsdk/Util.h>
```

Include dependency graph for Plugin.cpp:



6.17 Plugin.h File Reference

This graph shows which files directly or indirectly include this file:



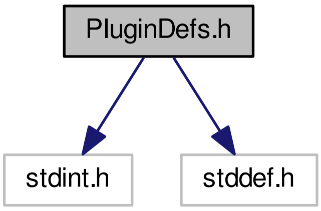
Classes

- class Plugin

6.18 PluginDefs.h File Reference

```
#include <stdint.h>
#include <stddef.h>
```

Include dependency graph for PluginDefs.h:



Classes

- struct DCUI
- struct tagCommandData
- struct tagConfigBool
- struct tagConfigInt
- struct tagConfigInt64
- struct tagConfigStr
- struct tagConfigValue
- struct tagConnectionData
- struct tagDCConfig
- struct tagDCConnection
- struct tagDCCore
- struct tagDCHooks
- struct tagDCHub
- struct tagDCInterface
- struct tagDCLog
- struct tagDCQueue
- struct tagDCTagger
- struct tagDCUtils
- struct tagHubData
- struct tagMetaData
- struct tagQueueData
- struct tagStringData
- struct tagTagData
- struct tagUDPData
- struct tagUserData

Macros

- #define Bool dcBool
- #define DCAPI
- #define DCAPI_CORE_VER 8
- #define DCEXP __attribute__ ((visibility("default")))
- #define DCIMP __attribute__ ((visibility("default")))
- #define DCINTF_CONFIG "generic.plugins.DCConfig" /∗ Config management ∗/
- #define DCINTF_CONFIG_VER 1
- #define DCINTF_DCPP_CONNECTIONS "dcpp.network.DCConnection" /∗ Peer connections ∗/
- #define DCINTF_DCPP_CONNECTIONS_VER 1
- #define DCINTF_DCPP_HUBS "dcpp.network.DCHub" /∗ Hubs ∗/
- #define DCINTF_DCPP_HUBS_VER 1
- #define DCINTF_DCPP_QUEUE "dcpp.queue.DCQueue" /∗ Download Queue ∗/
- #define DCINTF_DCPP_QUEUE_VER 1
- #define DCINTF_DCPP_TAGGER "dcpp.xml.DCTagger" /∗ Manipulation of an XML tagger ∗/
- #define DCINTF_DCPP_TAGGER_VER 1
- #define DCINTF_DCPP_UI "dcpp.ui.DCUI" /∗ User interface ∗/
- #define DCINTF_DCPP_UI_VER 1
- #define DCINTF_DCPP_UTILS "dcpp.utils.DCUtils" /∗ Utility and convenience functions ∗/
- #define DCINTF_DCPP_UTILS_VER 1
- #define DCINTF_HOOKS "generic.plugins.DCHooks"
- #define DCINTF_HOOKS_VER DCAPI_CORE_VER
- #define DCINTF_LOGGING "generic.plugins.DCLog" /∗ Logging functions ∗/
- #define DCINTF_LOGGING_VER 1
- #define False dcFalse

- #define HOOK_CHAT_IN "dcpp.chat.onIncomingChat" /∗ Incoming chat from hub (obj: HubData) ∗/
- #define HOOK_CHAT_OUT "dcpp.chat.onOutgoingChat" /∗ Outgoing chat (obj: HubData) ∗/
- #define HOOK_CHAT_PM_IN "dcpp.chat.onIncomingPM" /∗ Incoming private message (obj: UserData) ∗/
- #define HOOK_CHAT_PM_OUT "dcpp.chat.onOutgoingPM" /∗ Outgoing private message (obj: UserData) ∗/
- #define HOOK_HUB_OFFLINE "dcpp.hubs.onOffline" /∗ Hub has just gone offline (obj: HubData) ∗/
- #define HOOK_HUB_ONLINE "dcpp.hubs.onOnline" /∗ (New) hub has just gone online (obj: HubData) ∗/
- #define HOOK_NETWORK_CONN_IN "dcpp.network.onClientDataIn" /∗ Incoming client<->client protocol message (obj: ConnectionData) ∗/
- #define HOOK_NETWORK_CONN_OUT "dcpp.network.onClientDataOut" /∗ Outgoing client<->client protocol message (obj: ConnectionData) ∗/
- #define HOOK_NETWORK_HUB_IN "dcpp.network.onHubDataIn" /∗ Incoming protocol messages from hub (obj: HubData) ∗/
- #define HOOK_NETWORK_HUB_OUT "dcpp.network.onHubDataOut" /∗ Outgoing protocol message to hub (obj: HubData) ∗/
- #define HOOK_NETWORK_UDP_IN "dcpp.network.onUDPDataIn" /∗ Incoming UDP data (obj: UDPData) ∗/
- #define HOOK_NETWORK_UDP_OUT "dcpp.network.onUDPDataOut" /∗ Outgoing UDP data (obj: UDPData) ∗/
- #define HOOK_QUEUE_ADDED "dcpp.queue.onAdded" /∗ (New) item has been added to download queue (obj: QueueData) ∗/
- #define HOOK_QUEUE_FINISHED "dcpp.queue.onFinished" /∗ Item has just finished downloading (obj: QueueData) ∗/
- #define HOOK_QUEUE_MOVED "dcpp.queue.onMoved" /∗ Download queue item has been moved to new location (obj: QueueData) ∗/
- #define HOOK_QUEUE_REMOVED "dcpp.queue.onRemoved" /∗ Item has just been removed from download queue (obj: QueueData) ∗/
- #define HOOK_TIMER_MINUTE "dcpp.timer.onMinute" /∗ Timer event fired once per minute (data: tick value) ∗/
- #define HOOK_TIMER_SECOND "dcpp.timer.onSecond" /∗ Timer event fired once per second (data: tick value) ∗/
- #define HOOK_UI_CHAT_COMMAND "dcpp.ui.onChatCommand" /∗ Client side commands in hub chat (obj: HubData; data: CommandData) ∗/
- #define HOOK_UI_CHAT_COMMAND_PM "dcpp.ui.onChatCommandPM" /∗ Client side commands in private chat (obj: UserData; data: CommandData) ∗/
- #define HOOK_UI_CHAT_DISPLAY "dcpp.ui.onChatDisplay" /∗ Chat messages before they are displayed in chat (obj: UserData; data: StringData) ∗/
- #define HOOK_UI_CHAT_TAGS "dcpp.ui.onChatTags" /∗ Chat message tags before tag merging (obj: UserData; data: TagData) ∗/
- #define HOOK_UI_CREATED "dcpp.ui.onCreated" /∗ Host application UI has been created (obj: if any, impl. dependant) ∗/
- #define HOOK_USER_OFFLINE "dcpp.users.onOffline" /∗ User is offline (obj: UserData) ∗/
- #define HOOK_USER_ONLINE "dcpp.users.onOnline" /∗ User is online (obj: UserData) ∗/
- #define True dcTrue

Typedefs

- typedef dcptr_t dcptr_t Bool ∗ bBreak
- typedef struct tagCommandData CommandData
- typedef struct tagCommandData ∗ CommandDataPtr
- typedef struct tagConfigBool ConfigBool
- typedef struct tagConfigBool ∗ ConfigBoolPtr
- typedef struct tagConfigInt ConfigInt
- typedef struct tagConfigInt64 ConfigInt64
- typedef struct tagConfigInt64 ∗ ConfigInt64Ptr
- typedef struct tagConfigInt ∗ ConfigIntPtr
- typedef struct tagConfigStr ConfigStr

- typedef struct tagConfigStr ∗ ConfigStrPtr
- typedef enum tagConfigType ConfigType
- typedef struct tagConfigValue ConfigValue
- typedef struct tagConfigValue ∗ ConfigValuePtr
- typedef struct tagConnectionData ConnectionData
- typedef struct tagConnectionData ∗ ConnectionDataPtr
- typedef DCCorePtr core
- typedef enum tagDCBool dcBool
- typedef struct tagDCConfig DCConfig
- typedef struct tagDCConfig ∗ DCConfigPtr
- typedef struct tagDCConnection DCConnection
- typedef struct tagDCConnection ∗ DCConnectionPtr
- typedef struct tagDCCore DCCore
- typedef struct tagDCCore ∗ DCCorePtr
- typedef struct tagDCHooks DCHooks
- typedef struct tagDCHooks ∗ DCHooksPtr
- typedef struct tagDCHub DCHub
- typedef struct tagDCHub ∗ DCHubPtr
- typedef struct tagDCInterface DCInterface
- typedef struct tagDCInterface ∗ DCInterfacePtr
- typedef struct tagDCLog DCLog
- typedef struct tagDCLog ∗ DCLogPtr
- typedef void ∗ dcptr_t
- typedef struct tagDCQueue DCQueue
- typedef struct tagDCQueue ∗ DCQueuePtr
- typedef struct tagDCTagger DCTagger
- typedef struct tagDCTagger ∗ DCTaggerPtr
- typedef uint64_t dctime_t
- typedef struct DCUI DCUI
- typedef struct DCUI ∗ DCUIPtr
- typedef struct tagDCUtils DCUtils
- typedef struct tagDCUtils ∗ DCUtilsPtr
- typedef void ∗ hookHandle
- typedef struct tagHubData HubData
- typedef struct tagHubData ∗ HubDataPtr
- typedef void ∗ intfHandle
- typedef struct tagMetaData MetaData
- typedef struct tagMetaData ∗ MetaDataPtr
- typedef enum tagMsgType MsgType
- typedef enum tagPathType PathType
- typedef dcptr_t dcptr_t pCommon
- typedef DCCorePtr dcptr_t pData
- typedef enum tagPluginState PluginState
- typedef enum tagProtocolType ProtocolType
- typedef struct tagQueueData QueueData
- typedef struct tagQueueData ∗ QueueDataPtr
- typedef enum tagQueuePrio QueuePrio
- typedef struct tagStringData StringData
- typedef struct tagStringData ∗ StringDataPtr
- typedef void ∗ subsHandle
- typedef struct tagTagData TagData
- typedef struct tagTagData ∗ TagDataPtr
- typedef struct tagUDPData UDPData
- typedef struct tagUDPData ∗ UDPDataPtr
- typedef struct tagUserData UserData
- typedef struct tagUserData ∗ UserDataPtr

Enumerations

- enum tagConfigType { CFG_TYPE_UNKNOWN = -2, CFG_TYPE_REMOVE, CFG_TYPE_STRING, CFG_TYPE_INT, CFG_TYPE_BOOL, CFG_TYPE_INT64 }
- enum tagDCBool { dcFalse = 0, dcTrue }
- enum tagMsgType { MSG_CLIENT = 0, MSG_STATUS, MSG_SYSTEM }
- enum tagPathType { PATH_GLOBAL_CONFIG = 0, PATH_USER_CONFIG, PATH_USER_LOCAL, PATH_RESOURCES, PATH_LOCALE }
- enum tagPluginState { ON_INSTALL = 0, ON_LOAD, ON_LOAD_RUNTIME, ON_CONFIGURE, ON_UNLOAD, ON_UNINSTALL }
- enum tagProtocolType { PROTOCOL_ADC = 0, PROTOCOL_NMDC, PROTOCOL_DHT }
- enum tagQueuePrio { PRIO_DEFAULT = -1, PRIO_LOWEST = 1, PRIO_LOW, PRIO_NORMAL, PRIO_HIGH, PRIO_HIGHEST }

Functions

- typedef Bool (DCAPI ∗DCMAIN)(PluginState pluginState
- typedef void (DCAPI ∗DCCommandFunc)(const char ∗name)

6.18.1 Macro Definition Documentation

6.18.1.1 #define Bool dcBool

Definition at line 172 of file PluginDefs.h.

6.18.1.2 #define DCAPI

Definition at line 49 of file PluginDefs.h.

6.18.1.3 #define DCAPI_CORE_VER 8

Definition at line 35 of file PluginDefs.h.

Referenced by pluginInit().

6.18.1.4 #define DCEXP __attribute__ ((visibility("default")))

Definition at line 51 of file PluginDefs.h.

6.18.1.5 #define DCIMP __attribute__ ((visibility("default")))

Definition at line 52 of file PluginDefs.h.

6.18.1.6 #define DCINTF_CONFIG "generic.plugins.DCConfig" /∗ Config management ∗/

Definition at line 60 of file PluginDefs.h.

Referenced by dcapi::Config::init(), and init_cfg().

6.18.1.7 #define DCINTF_CONFIG_VER 1

Definition at line 61 of file PluginDefs.h.

Referenced by dcapi::Config::init(), and init_cfg().

6.18.1.8 #define DCINTF_DCPP_CONNECTIONS "dcpp.network.DCConnection" /∗ Peer connections ∗/

Definition at line 67 of file PluginDefs.h.

Referenced by dcapi::Connections::init().

6.18.1.9 #define DCINTF_DCPP_CONNECTIONS_VER 1

Definition at line 68 of file PluginDefs.h.

Referenced by dcapi::Connections::init().

6.18.1.10 #define DCINTF_DCPP_HUBS "dcpp.network.DCHub" /∗ Hubs ∗/

Definition at line 70 of file PluginDefs.h.

Referenced by dcapi::Hubs::init().

6.18.1.11 #define DCINTF_DCPP_HUBS_VER 1

Definition at line 71 of file PluginDefs.h.

Referenced by dcapi::Hubs::init().

6.18.1.12 #define DCINTF_DCPP_QUEUE "dcpp.queue.DCQueue" /∗ Download Queue ∗/

Definition at line 73 of file PluginDefs.h.

Referenced by dcapi::Queue::init().

6.18.1.13 #define DCINTF_DCPP_QUEUE_VER 1

Definition at line 74 of file PluginDefs.h.

Referenced by dcapi::Queue::init().

6.18.1.14 #define DCINTF_DCPP_TAGGER "dcpp.xml.DCTagger" /∗ Manipulation of an XML tagger ∗/

Definition at line 79 of file PluginDefs.h.

Referenced by dcapi::Tagger::init().

6.18.1.15 #define DCINTF_DCPP_TAGGER_VER 1

Definition at line 80 of file PluginDefs.h.

Referenced by dcapi::Tagger::init().

6.18.1.16 #define DCINTF_DCPP_UI "dcpp.ui.DCUI" /∗ User interface ∗/

Definition at line 82 of file PluginDefs.h.

Referenced by dcapi::UI::init().

6.18.1.17 #define DCINTF_DCPP_UI_VER 1

Definition at line 83 of file PluginDefs.h.

Referenced by dcapi::UI::init().

6.18.1.18 #define DCINTF_DCPP_UTILS "dcpp.utils.DCUtils" /∗ Utility and convenience functions ∗/

Definition at line 76 of file PluginDefs.h.

Referenced by dcapi::Util::init().

6.18.1.19 #define DCINTF_DCPP_UTILS_VER 1

Definition at line 77 of file PluginDefs.h.

Referenced by dcapi::Util::init().

6.18.1.20 #define DCINTF_HOOKS "generic.plugins.DCHooks"

Definition at line 56 of file PluginDefs.h.

Referenced by dcapi::Hooks::init().

6.18.1.21 #define DCINTF_HOOKS_VER DCAPI_CORE_VER

Definition at line 57 of file PluginDefs.h.

Referenced by dcapi::Hooks::init().

6.18.1.22 #define DCINTF_LOGGING "generic.plugins.DCLog" /∗ Logging functions ∗/

Definition at line 63 of file PluginDefs.h.

Referenced by dcapi::Logger::init().

6.18.1.23 #define DCINTF_LOGGING_VER 1

Definition at line 64 of file PluginDefs.h.

Referenced by dcapi::Logger::init().

6.18.1.24 #define False dcFalse

Definition at line 174 of file PluginDefs.h.

Referenced by dcapi::Hooks::eventCallback(), init_cfg(), Plugin::main(), and dcapi::Config::setConfig().

6.18.1.25 #define HOOK_CHAT_IN "dcpp.chat.onIncomingChat" /∗ Incoming chat from hub (obj: HubData) ∗/

Definition at line 86 of file PluginDefs.h.

Referenced by dcapi::Hooks::Chat::onIncomingChat().

6.18.1.26 #define HOOK_CHAT_OUT "dcpp.chat.onOutgoingChat" /∗ Outgoing chat (obj: HubData) ∗/

Definition at line 87 of file PluginDefs.h.

Referenced by dcapi::Hooks::Chat::onOutgoingChat().

6.18.1.27 #define HOOK_CHAT_PM_IN "dcpp.chat.onIncomingPM" /∗ Incoming private message (obj: UserData) ∗/

Definition at line 88 of file PluginDefs.h.

Referenced by dcapi::Hooks::Chat::onIncomingPM().

6.18.1.28 #define HOOK_CHAT_PM_OUT "dcpp.chat.onOutgoingPM" /∗ Outgoing private message (obj: UserData) ∗/

Definition at line 89 of file PluginDefs.h.

Referenced by dcapi::Hooks::Chat::onOutgoingPM().

6.18.1.29 #define HOOK_HUB_OFFLINE "dcpp.hubs.onOffline" /∗ Hub has just gone offline (obj: HubData) ∗/

Definition at line 95 of file PluginDefs.h.

Referenced by dcapi::Hooks::Hubs::onOffline().

6.18.1.30 #define HOOK_HUB_ONLINE "dcpp.hubs.onOnline" /∗ (New) hub has just gone online (obj: HubData) ∗/

Definition at line 94 of file PluginDefs.h.

Referenced by dcapi::Hooks::Hubs::onOnline().

6.18.1.31 #define HOOK_NETWORK_CONN_IN "dcpp.network.onClientDataIn" /∗ Incoming client<->client protocol message (obj: ConnectionData) ∗/

Definition at line 101 of file PluginDefs.h.

Referenced by dcapi::Hooks::Network::onClientDataIn().

6.18.1.32 #define HOOK_NETWORK_CONN_OUT "dcpp.network.onClientDataOut" /∗ Outgoing client<->client protocol message (obj: ConnectionData) ∗/

Definition at line 102 of file PluginDefs.h.

Referenced by dcapi::Hooks::Network::onClientDataOut().

6.18.1.33 #define HOOK_NETWORK_HUB_IN "dcpp.network.onHubDataIn" /∗ Incoming protocol messages from hub (obj: HubData) ∗/

Definition at line 99 of file PluginDefs.h.

Referenced by dcapi::Hooks::Network::onHubDataIn().

6.18.1.34 #define HOOK_NETWORK_HUB_OUT "dcpp.network.onHubDataOut" /∗ Outgoing protocol message to hub (obj: HubData) ∗/

Definition at line 100 of file PluginDefs.h.

Referenced by dcapi::Hooks::Network::onHubDataOut().

6.18.1.35 #define HOOK_NETWORK_UDP_IN "dcpp.network.onUDPDataIn" /∗ Incoming UDP data (obj: UDPData) ∗/

Definition at line 103 of file PluginDefs.h.

Referenced by dcapi::Hooks::Network::onUDPDataIn().

6.18.1.36 #define HOOK_NETWORK_UDP_OUT "dcpp.network.onUDPDataOut" /∗ Outgoing UDP data (obj: UDPData) ∗/

Definition at line 104 of file PluginDefs.h.

Referenced by dcapi::Hooks::Network::onUDPDataOut().

6.18.1.37 #define HOOK_QUEUE_ADDED "dcpp.queue.onAdded" /∗ (New) item has been added to download queue (obj: QueueData) ∗/

Definition at line 106 of file PluginDefs.h.

Referenced by dcapi::Hooks::Queue::onAdded().

6.18.1.38 #define HOOK_QUEUE_FINISHED "dcpp.queue.onFinished" /∗ Item has just finished downloading (obj: QueueData) ∗/

Definition at line 109 of file PluginDefs.h.

Referenced by dcapi::Hooks::Queue::onFinished().

6.18.1.39 #define HOOK_QUEUE_MOVED "dcpp.queue.onMoved" /∗ Download queue item has been moved to new location (obj: QueueData) ∗/

Definition at line 107 of file PluginDefs.h.

Referenced by dcapi::Hooks::Queue::onMoved().

6.18.1.40 #define HOOK_QUEUE_REMOVED "dcpp.queue.onRemoved" /∗ Item has just been removed from download queue (obj: QueueData) ∗/

Definition at line 108 of file PluginDefs.h.

Referenced by dcapi::Hooks::Queue::onRemoved().

6.18.1.41 #define HOOK_TIMER_MINUTE "dcpp.timer.onMinute" /∗ Timer event fired once per minute (data: tick value) ∗/

Definition at line 92 of file PluginDefs.h.

Referenced by dcapi::Hooks::Timer::onMinute().

6.18.1.42 #define HOOK_TIMER_SECOND "dcpp.timer.onSecond" /∗ Timer event fired once per second (data: tick value) ∗/

Definition at line 91 of file PluginDefs.h.

Referenced by dcapi::Hooks::Timer::onSecond().

6.18.1.43 #define HOOK_UI_CHAT_COMMAND "dcpp.ui.onChatCommand" /∗ Client side commands in hub chat (obj: HubData; data: CommandData) ∗/

Definition at line 114 of file PluginDefs.h.

Referenced by dcapi::Hooks::UI::onChatCommand().

6.18.1.44 #define HOOK_UI_CHAT_COMMAND_PM "dcpp.ui.onChatCommandPM" /∗ Client side commands in private chat (obj: UserData; data: CommandData) ∗/

Definition at line 115 of file PluginDefs.h.

Referenced by dcapi::Hooks::UI::onChatCommandPM().

6.18.1.45 #define HOOK_UI_CHAT_DISPLAY "dcpp.ui.onChatDisplay" /∗ Chat messages before they are displayed in chat (obj: UserData; data: StringData) ∗/

Definition at line 113 of file PluginDefs.h.

Referenced by dcapi::Hooks::UI::onChatDisplay().

6.18.1.46 #define HOOK_UI_CHAT_TAGS "dcpp.ui.onChatTags" /∗ Chat message tags before tag merging (obj: UserData; data: TagData) ∗/

Definition at line 112 of file PluginDefs.h.

Referenced by dcapi::Hooks::UI::onChatTags().

6.18.1.47 #define HOOK_UI_CREATED "dcpp.ui.onCreated" /∗ Host application UI has been created (obj: if any, impl. dependant) ∗/

Definition at line 111 of file PluginDefs.h.

Referenced by dcapi::Hooks::UI::onCreated().

6.18.1.48 #define HOOK_USER_OFFLINE "dcpp.users.onOffline" /∗ User is offline (obj: UserData) ∗/

Definition at line 97 of file PluginDefs.h.

Referenced by dcapi::Hooks::Users::onOffline().

6.18.1.49 #define HOOK_USER_ONLINE "dcpp.users.onOnline" /∗ User is online (obj: UserData) ∗/

Definition at line 96 of file PluginDefs.h.

Referenced by dcapi::Hooks::Users::onOnline().

6.18.1.50 #define True dcTrue

Definition at line 173 of file PluginDefs.h.

Referenced by dcapi::Hooks::eventCallback(), init_cfg(), Plugin::main(), and dcapi::Config::setConfig().

6.18.2 Typedef Documentation

6.18.2.1 typedef dcptr_t dcptr_t Bool∗ bBreak

Definition at line 317 of file PluginDefs.h.

6.18.2.2 typedef struct tagCommandData CommandData

6.18.2.3 typedef struct tagCommandData ∗ CommandDataPtr

6.18.2.4 typedef struct tagConfigBool ConfigBool

6.18.2.5 typedef struct tagConfigBool ∗ ConfigBoolPtr

6.18.2.6 typedef struct tagConfigInt ConfigInt

6.18.2.7 typedef struct tagConfigInt64 ConfigInt64

6.18.2.8 typedef struct tagConfigInt64 ∗ ConfigInt64Ptr

6.18.2.9 typedef struct tagConfigInt ∗ ConfigIntPtr

6.18.2.10 typedef struct tagConfigStr ConfigStr

6.18.2.11 typedef struct tagConfigStr ∗ ConfigStrPtr

6.18.2.12 typedef enum tagConfigType ConfigType

6.18.2.13 typedef struct tagConfigValue ConfigValue

6.18.2.14 typedef struct tagConfigValue ∗ ConfigValuePtr

6.18.2.15 typedef struct tagConnectionData ConnectionData

6.18.2.16 typedef struct tagConnectionData ∗ ConnectionDataPtr

6.18.2.17 typedef DCCorePtr core

Definition at line 312 of file PluginDefs.h.

6.18.2.18 typedef enum tagDCBool dcBool

6.18.2.19 typedef struct tagDCConfig DCConfig

6.18.2.20 typedef struct tagDCConfig ∗ DCConfigPtr

6.18.2.21 typedef struct tagDCConnection DCConnection

6.18.2.22 typedef struct tagDCConnection ∗ DCConnectionPtr

6.18.2.23 typedef struct tagDCCore DCCore

6.18.2.24 typedef struct tagDCCore ∗ DCCorePtr

6.18.2.25 typedef struct tagDCHooks DCHooks

6.18.2.26 typedef struct tagDCHooks ∗ DCHooksPtr

6.18.2.27 typedef struct tagDCHub DCHub

6.18.2.28 typedef struct tagDCHub ∗ DCHubPtr

6.18.2.29 typedef struct tagDCInterface DCInterface

6.18.2.30 typedef struct tagDCInterface ∗ DCInterfacePtr

6.18.2.31 typedef struct tagDCLog DCLog

6.18.2.32 typedef struct tagDCLog ∗ DCLogPtr

6.18.2.33 typedef void ∗ dcptr_t

Definition at line 167 of file PluginDefs.h.

6.18.2.34 typedef struct tagDCQueue DCQueue

6.18.2.35 typedef struct tagDCQueue ∗ DCQueuePtr

6.18.2.36 typedef struct tagDCTagger DCTagger

6.18.2.37 typedef struct tagDCTagger ∗ DCTaggerPtr

6.18.2.38 typedef uint64_t dctime_t

Definition at line 169 of file PluginDefs.h.

6.18.2.39 typedef struct DCUI DCUI

6.18.2.40 typedef struct DCUI ∗ DCUIPtr

6.18.2.41 typedef struct tagDCUtils DCUtils

6.18.2.42 typedef struct tagDCUtils ∗ DCUtilsPtr

6.18.2.43 typedef void∗ hookHandle

Definition at line 167 of file PluginDefs.h.

6.18.2.44 typedef struct tagHubData HubData

6.18.2.45 typedef struct tagHubData ∗ HubDataPtr

6.18.2.46 typedef void ∗ intfHandle

Definition at line 167 of file PluginDefs.h.

6.18.2.47 typedef struct tagMetaData MetaData

6.18.2.48 typedef struct tagMetaData ∗ MetaDataPtr

6.18.2.49 typedef enum tagMsgType MsgType

6.18.2.50 typedef enum tagPathType PathType

6.18.2.51 typedef dcptr_t dcptr_t pCommon

Definition at line 317 of file PluginDefs.h.

6.18.2.52 typedef dcptr_t pData

Definition at line 312 of file PluginDefs.h.

6.18.2.53 typedef enum tagPluginState PluginState

6.18.2.54 typedef enum tagProtocolType ProtocolType

6.18.2.55 typedef struct tagQueueData QueueData

6.18.2.56 typedef struct tagQueueData ∗ QueueDataPtr

6.18.2.57 typedef enum tagQueuePrio QueuePrio

6.18.2.58 typedef struct tagStringData StringData

6.18.2.59 typedef struct tagStringData ∗ StringDataPtr

6.18.2.60 typedef void ∗ subsHandle

Definition at line 167 of file PluginDefs.h.

6.18.2.61 typedef struct tagTagData TagData

6.18.2.62 typedef struct tagTagData ∗ TagDataPtr

6.18.2.63 typedef struct tagUDPData UDPData

6.18.2.64 typedef struct tagUDPData ∗ UDPDataPtr

6.18.2.65 typedef struct tagUserData UserData

6.18.2.66 typedef struct tagUserData ∗ UserDataPtr

6.18.3 Enumeration Type Documentation

6.18.3.1 enum tagConfigType

Enumerator

CFG_TYPE_UNKNOWN

CFG_TYPE_REMOVE

CFG_TYPE_STRING

CFG_TYPE_INT

CFG_TYPE_BOOL

CFG_TYPE_INT64

Definition at line 128 of file PluginDefs.h.

6.18.3.2 enum tagDCBool

Enumerator

- ***dcFalse***
- ***dcTrue***

Definition at line 168 of file PluginDefs.h.

6.18.3.3 enum tagMsgType

Enumerator

- ***MSG_CLIENT***
- ***MSG_STATUS***
- ***MSG_SYSTEM***

Definition at line 151 of file PluginDefs.h.

6.18.3.4 enum tagPathType

Enumerator

- ***PATH_GLOBAL_CONFIG***
- ***PATH_USER_CONFIG***
- ***PATH_USER_LOCAL***
- ***PATH_RESOURCES***
- ***PATH_LOCALE***

Definition at line 143 of file PluginDefs.h.

6.18.3.5 enum tagPluginState

Enumerator

- ***ON_INSTALL***
- ***ON_LOAD***
- ***ON_LOAD_RUNTIME***
- ***ON_CONFIGURE***
- ***ON_UNLOAD***
- ***ON_UNINSTALL***

Definition at line 118 of file PluginDefs.h.

6.18.3.6 enum tagProtocolType

Enumerator

- ***PROTOCOL_ADC***
- ***PROTOCOL_NMDC***
- ***PROTOCOL_DHT***

Definition at line 137 of file PluginDefs.h.

6.18.3.7 enum tagQueuePrio

Enumerator

PRIO_DEFAULT

PRIO_LOWEST

PRIO_LOW

PRIO_NORMAL

PRIO_HIGH

PRIO_HIGHEST

Definition at line 157 of file PluginDefs.h.

6.18.4 Function Documentation

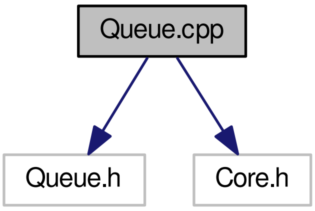
6.18.4.1 typedef Bool (DCAPI * *DCMAIN*)

6.18.4.2 typedef void (DCAPI * *DCCommandFunc*) const

6.19 Queue.cpp File Reference

```
#include "Queue.h"
#include "Core.h"
```

Include dependency graph for Queue.cpp:



Namespaces

- dcapi

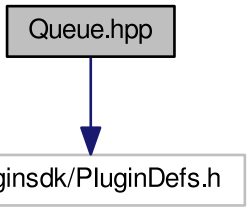
Constant Groups

- dcapi

6.20 Queue.hpp File Reference

`#include <pluginsdk/PluginDefs.h>`

Include dependency graph for Queue.hpp:



Classes

- class dcapi::Queue

Namespaces

- dcapi

Constant Groups

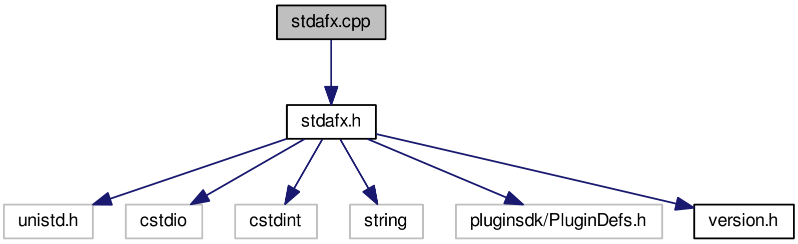
- dcapi

6.21 resource.h File Reference

6.22 stdafx.cpp File Reference

`#include "stdafx.h"`

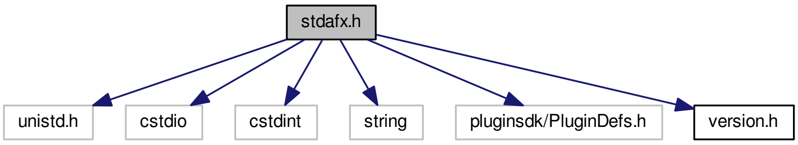
Include dependency graph for stdafx.cpp:



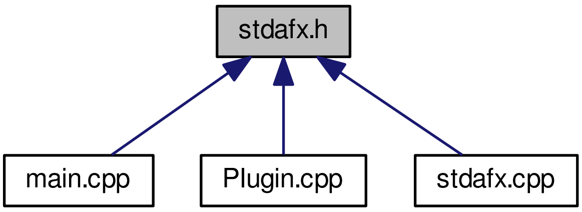
6.23 stdafx.h File Reference

```
#include <unistd.h>
#include <cstdio>
#include <cstdint>
#include <string>
#include <pluginsdk/PluginDefs.h>
#include "version.h"
```

Include dependency graph for stdafx.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define PLUGIN_VERSION

Typedefs

- typedef std::string tstring

6.23.1 Macro Definition Documentation

6.23.1.1 #define PLUGIN_VERSION

Value:

```
(PLUGIN_VERSION_MAJOR + (PLUGIN_VERSION_MINOR >= 10 ?
      PLUGIN_VERSION_MINOR >= 100 ? \
    PLUGIN_VERSION_MINOR / 1000. : PLUGIN_VERSION_MINOR / 100. :
      PLUGIN_VERSION_MINOR / 10.))
```

Definition at line 41 of file stdafx.h.

Referenced by pluginInit().

6.23.2 Typedef Documentation

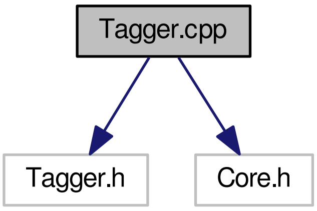
6.23.2.1 typedef std::string tstring

Definition at line 47 of file stdafx.h.

6.24 Tagger.cpp File Reference

```
#include "Tagger.h"
#include "Core.h"
```

Include dependency graph for Tagger.cpp:



Namespaces

- dcapi

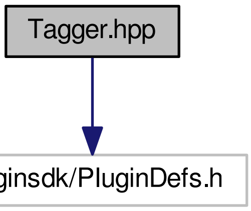
Constant Groups

- dcapi

6.25 Tagger.hpp File Reference

```
#include <pluginsdk/PluginDefs.h>
```

Include dependency graph for Tagger.hpp:



Classes

- class dcapi::Tagger

Namespaces

- dcapi

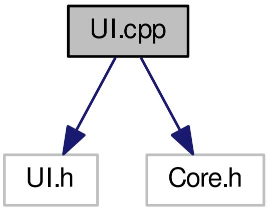
Constant Groups

- dcapi

6.26 UI.cpp File Reference

```
#include "UI.h"
#include "Core.h"
```

Include dependency graph for UI.cpp:



Namespaces

- dcapi

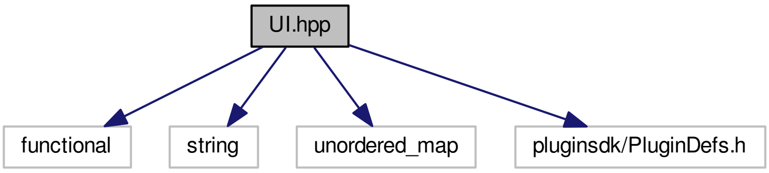
Constant Groups

- dcapi

6.27 UI.hpp File Reference

```
#include <functional>
#include <string>
#include <unordered_map>
#include <pluginsdk/PluginDefs.h>
```

Include dependency graph for UI.hpp:



Classes

- class dcapi::UI

Namespaces

- dcapi

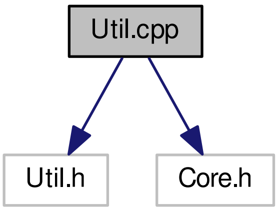
Constant Groups

- dcapi

6.28 Util.cpp File Reference

```
#include "Util.h"
#include "Core.h"
```

Include dependency graph for Util.cpp:



Namespaces

- dcapi

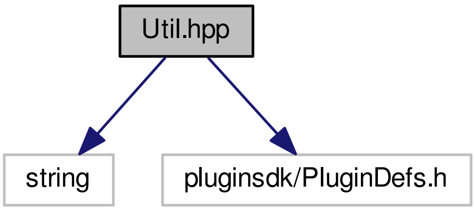
Constant Groups

- dcapi

6.29 Util.hpp File Reference

```
#include <string>
#include <pluginsdk/PluginDefs.h>
```

Include dependency graph for Util.hpp:



Classes

- class dcapi::Util

Namespaces

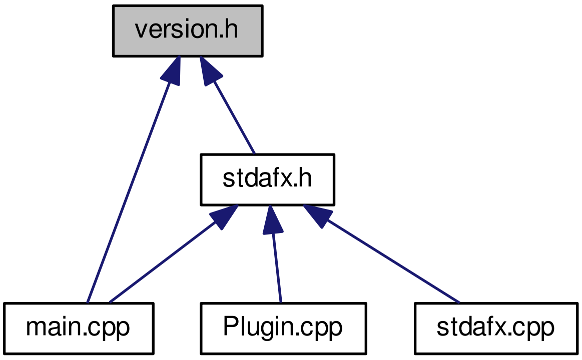
- dcapi

Constant Groups

- dcapi

6.30 version.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define PLUGIN_AUTHOR "..."
- #define PLUGIN_DESC "..."
- #define PLUGIN_GUID "..."
- #define PLUGIN_NAME "..."
- #define PLUGIN_VERSION_MAJOR 1
- #define PLUGIN_VERSION_MINOR 0
- #define PLUGIN_WEB "N/A"

6.30.1 Macro Definition Documentation

6.30.1.1 #define PLUGIN_AUTHOR "..."

Definition at line 33 of file version.h.

Referenced by pluginInit().

6.30.1.2 #define PLUGIN_DESC "..."

Definition at line 36 of file version.h.

Referenced by pluginInit().

6.30.1.3 #define PLUGIN_GUID "..."

Definition at line 27 of file version.h.

Referenced by Plugin::onLoad(), and pluginInit().

6.30.1.4 #define PLUGIN_NAME "..."

Definition at line 30 of file version.h.

Referenced by pluginInit().

6.30.1.5 #define PLUGIN_VERSION_MAJOR 1

Definition at line 39 of file version.h.

6.30.1.6 #define PLUGIN_VERSION_MINOR 0

Definition at line 40 of file version.h.

6.30.1.7 #define PLUGIN_WEB "N/A"

Definition at line 43 of file version.h.

Referenced by pluginInit().

Index