

DC++ plugin SDK (C++)

Generated by Doxygen 1.7.4

Tue Feb 5 2013 20:18:12

Contents

1	Directory Hierarchy	1
1.1	Directories	1
2	Namespace Index	3
2.1	Namespace List	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Directory Documentation	9
5.1	/cygdrive/h/Dev/DC++/pluginsdk/cpp/pluginsdk/ Directory Reference . . .	9
5.2	/cygdrive/h/Dev/DC++/pluginsdk/cpp/src/ Directory Reference	10
6	Namespace Documentation	11
6.1	dcapi Namespace Reference	11
7	Class Documentation	13
7.1	dcapi::Hooks::Chat Struct Reference	13
7.1.1	Detailed Description	13
7.1.2	Member Function Documentation	13
7.1.2.1	onIncomingChat	13
7.1.2.2	onIncomingPM	14
7.1.2.3	onOutgoingChat	14
7.1.2.4	onOutgoingPM	14
7.2	dcapi::Config Class Reference	15

7.2.1	Detailed Description	16
7.2.2	Member Function Documentation	16
7.2.2.1	freeCoreConfig	16
7.2.2.2	getBoolConfig	16
7.2.2.3	getConfig	16
7.2.2.4	getConfig	17
7.2.2.5	getCoreConfig	17
7.2.2.6	getInt64Config	17
7.2.2.7	getIntConfig	17
7.2.2.8	getPath	17
7.2.2.9	handle	17
7.2.2.10	init	18
7.2.2.11	init	18
7.2.2.12	setConfig	18
7.2.2.13	setConfig	18
7.2.2.14	setConfig	18
7.2.2.15	setConfig	18
7.2.2.16	setConfig	19
7.2.2.17	setConfig	19
7.2.3	Member Data Documentation	19
7.2.3.1	config	19
7.2.3.2	guid	20
7.3	dcapi::Connections Class Reference	20
7.3.1	Detailed Description	21
7.3.2	Member Function Documentation	21
7.3.2.1	handle	21
7.3.2.2	init	21
7.3.2.3	init	22
7.3.3	Member Data Documentation	22
7.3.3.1	connections	22
7.4	dcapi::Core Class Reference	22
7.4.1	Detailed Description	23
7.4.2	Member Function Documentation	23
7.4.2.1	handle	24

7.4.2.2	init	25
7.4.3	Member Data Documentation	26
7.4.3.1	appName	26
7.4.3.2	core	26
7.5	DCUI Struct Reference	26
7.5.1	Detailed Description	27
7.5.2	Member Function Documentation	27
7.5.2.1	void	27
7.5.2.2	void	27
7.5.2.3	void	27
7.5.3	Member Data Documentation	27
7.5.3.1	apiVersion	27
7.5.3.2	command	27
7.6	dcapi::Hooks Class Reference	27
7.6.1	Detailed Description	29
7.6.2	Member Typedef Documentation	29
7.6.2.1	Callback	29
7.6.3	Member Function Documentation	29
7.6.3.1	addEvent	29
7.6.3.2	clear	31
7.6.3.3	empty	31
7.6.3.4	eventCallback	31
7.6.3.5	handle	33
7.6.3.6	init	33
7.6.3.7	init	33
7.6.3.8	remove	33
7.6.4	Member Data Documentation	33
7.6.4.1	events	33
7.6.4.2	hooks	33
7.7	dcapi::Hubs Class Reference	34
7.7.1	Detailed Description	34
7.7.2	Member Function Documentation	34
7.7.2.1	handle	35
7.7.2.2	init	35

7.7.2.3	init	35
7.7.3	Member Data Documentation	36
7.7.3.1	hubs	36
7.8	dcapi::Hooks::Hubs Struct Reference	36
7.8.1	Detailed Description	36
7.8.2	Member Function Documentation	36
7.8.2.1	onOffline	36
7.8.2.2	onOnline	36
7.9	dcapi::Logger Class Reference	37
7.9.1	Detailed Description	38
7.9.2	Member Function Documentation	38
7.9.2.1	handle	38
7.9.2.2	init	38
7.9.2.3	init	38
7.9.2.4	log	38
7.9.3	Member Data Documentation	39
7.9.3.1	logger	39
7.10	dcapi::Hooks::Network Struct Reference	39
7.10.1	Detailed Description	39
7.10.2	Member Function Documentation	39
7.10.2.1	onClientDataIn	39
7.10.2.2	onClientDataOut	40
7.10.2.3	onHubDataIn	40
7.10.2.4	onHubDataOut	40
7.11	Plugin Class Reference	41
7.11.1	Detailed Description	41
7.11.2	Constructor & Destructor Documentation	41
7.11.2.1	Plugin	41
7.11.2.2	~Plugin	41
7.11.3	Member Function Documentation	42
7.11.3.1	main	42
7.11.3.2	onLoad	42
7.12	dcapi::Queue Class Reference	43
7.12.1	Detailed Description	44

7.12.2	Member Function Documentation	44
7.12.2.1	handle	44
7.12.2.2	init	44
7.12.2.3	init	45
7.12.3	Member Data Documentation	45
7.12.3.1	queue	45
7.13	dcapi::Hooks::Queue Struct Reference	45
7.13.1	Detailed Description	45
7.13.2	Member Function Documentation	46
7.13.2.1	onAdded	46
7.13.2.2	onFinished	46
7.13.2.3	onMoved	46
7.13.2.4	onRemoved	47
7.14	tagCommandData Struct Reference	47
7.14.1	Detailed Description	48
7.14.2	Member Data Documentation	48
7.14.2.1	command	48
7.14.2.2	params	48
7.15	tagConfigBool Struct Reference	48
7.15.1	Detailed Description	49
7.15.2	Member Data Documentation	49
7.15.2.1	type	49
7.15.2.2	value	49
7.16	tagConfigInt Struct Reference	49
7.16.1	Detailed Description	49
7.16.2	Member Data Documentation	50
7.16.2.1	type	50
7.16.2.2	value	50
7.17	tagConfigInt64 Struct Reference	50
7.17.1	Detailed Description	50
7.17.2	Member Data Documentation	50
7.17.2.1	type	51
7.17.2.2	value	51
7.18	tagConfigStr Struct Reference	51

7.18.1	Detailed Description	51
7.18.2	Member Data Documentation	51
7.18.2.1	type	51
7.18.2.2	value	52
7.19	tagConfigValue Struct Reference	52
7.19.1	Detailed Description	52
7.19.2	Member Data Documentation	52
7.19.2.1	type	52
7.20	tagConnectionData Struct Reference	52
7.20.1	Detailed Description	53
7.20.2	Member Data Documentation	53
7.20.2.1	ip	53
7.20.2.2	isManaged	53
7.20.2.3	isOp	53
7.20.2.4	isSecure	54
7.20.2.5	object	54
7.20.2.6	port	54
7.20.2.7	protocol	54
7.21	tagDCConfig Struct Reference	54
7.21.1	Detailed Description	55
7.21.2	Member Function Documentation	55
7.21.2.1	ConfigValuePtr	55
7.21.2.2	ConfigValuePtr	55
7.21.2.3	get_path	55
7.21.2.4	void	55
7.21.2.5	void	55
7.21.3	Member Data Documentation	56
7.21.3.1	apiVersion	56
7.21.3.2	setting	56
7.21.3.3	type	56
7.21.3.4	val	56
7.22	tagDCConnection Struct Reference	56
7.22.1	Detailed Description	57
7.22.2	Member Function Documentation	57

7.22.2.1	UserDataPtr	57
7.22.2.2	void	57
7.22.2.3	void	57
7.22.2.4	void	57
7.22.3	Member Data Documentation	57
7.22.3.1	apiVersion	57
7.22.3.2	cmd	57
7.22.3.3	data	57
7.22.3.4	graceless	57
7.22.3.5	n	58
7.22.3.6	port	58
7.23	tagDCCore Struct Reference	58
7.23.1	Detailed Description	59
7.23.2	Member Function Documentation	59
7.23.2.1	Bool	59
7.23.2.2	Bool	59
7.23.2.3	DCInterfacePtr	59
7.23.2.4	host_name	59
7.23.2.5	intfHandle	59
7.23.3	Member Data Documentation	59
7.23.3.1	apiVersion	59
7.23.3.2	pInterface	59
7.23.3.3	version	59
7.24	tagDCHooks Struct Reference	60
7.24.1	Detailed Description	60
7.24.2	Member Function Documentation	61
7.24.2.1	Bool	61
7.24.2.2	Bool	61
7.24.2.3	hookHandle	61
7.24.2.4	size_t	61
7.24.2.5	subsHandle	61
7.24.3	Member Data Documentation	61
7.24.3.1	apiVersion	61
7.24.3.2	defProc	61

7.24.3.3	hookProc	61
7.24.3.4	pCommon	61
7.24.3.5	pData	61
7.24.3.6	pObject	61
7.25	tagDCHub Struct Reference	61
7.25.1	Detailed Description	62
7.25.2	Member Function Documentation	63
7.25.2.1	Bool	63
7.25.2.2	HubDataPtr	63
7.25.2.3	HubDataPtr	63
7.25.2.4	HubDataPtr	63
7.25.2.5	UserDataPtr	63
7.25.2.6	UserDataPtr	63
7.25.2.7	void	63
7.25.2.8	void	63
7.25.2.9	void	63
7.25.2.10	void	63
7.25.2.11	void	63
7.25.2.12	void	63
7.25.2.13	void	63
7.25.3	Member Data Documentation	63
7.25.3.1	apiVersion	63
7.25.3.2	cmd	63
7.25.3.3	hubUrl	63
7.25.3.4	msg	63
7.25.3.5	nick	64
7.25.3.6	password	64
7.25.3.7	thirdPerson	64
7.25.3.8	type	64
7.26	tagDCInterface Struct Reference	64
7.26.1	Detailed Description	64
7.26.2	Member Data Documentation	65
7.26.2.1	apiVersion	65
7.27	tagDCLog Struct Reference	65

7.27.1	Detailed Description	65
7.27.2	Member Function Documentation	65
7.27.2.1	void	65
7.27.3	Member Data Documentation	66
7.27.3.1	apiVersion	66
7.28	tagDCQueue Struct Reference	66
7.28.1	Detailed Description	67
7.28.2	Member Function Documentation	67
7.28.2.1	Bool	67
7.28.2.2	QueueDataPtr	67
7.28.2.3	QueueDataPtr	67
7.28.2.4	QueueDataPtr	67
7.28.2.5	QueueDataPtr	67
7.28.2.6	void	67
7.28.2.7	void	67
7.28.2.8	void	67
7.28.3	Member Data Documentation	67
7.28.3.1	apiVersion	67
7.28.3.2	priority	67
7.28.3.3	silent	67
7.28.3.4	size	67
7.28.3.5	target	67
7.29	tagDCTagger Struct Reference	68
7.29.1	Detailed Description	68
7.29.2	Member Function Documentation	69
7.29.2.1	get_text	69
7.29.2.2	void	69
7.29.2.3	void	69
7.29.3	Member Data Documentation	69
7.29.3.1	apiVersion	69
7.29.3.2	attributes	69
7.29.3.3	end	69
7.29.3.4	id	69
7.29.3.5	replacement	69

7.29.3.6	start	69
7.30	tagDCUtils Struct Reference	69
7.30.1	Detailed Description	71
7.30.2	Member Function Documentation	71
7.30.2.1	size_t	71
7.30.2.2	size_t	71
7.30.2.3	size_t	71
7.30.2.4	size_t	71
7.30.2.5	size_t	71
7.30.2.6	size_t	71
7.30.3	Member Data Documentation	71
7.30.3.1	apiVersion	71
7.30.3.2	n	71
7.30.3.3	n	71
7.30.3.4	n	71
7.30.3.5	src	71
7.30.3.6	src	72
7.30.3.7	src	72
7.31	dcapi::Tagger Class Reference	72
7.31.1	Detailed Description	72
7.31.2	Member Function Documentation	73
7.31.2.1	handle	73
7.31.2.2	init	73
7.31.2.3	init	73
7.31.3	Member Data Documentation	74
7.31.3.1	tagger	74
7.32	tagHubData Struct Reference	74
7.32.1	Detailed Description	74
7.32.2	Member Data Documentation	75
7.32.2.1	ip	75
7.32.2.2	isManaged	75
7.32.2.3	isOp	75
7.32.2.4	isSecure	75
7.32.2.5	object	75

7.32.2.6	port	75
7.32.2.7	protocol	75
7.32.2.8	url	75
7.33	tagMetaData Struct Reference	75
7.33.1	Detailed Description	76
7.33.2	Member Data Documentation	76
7.33.2.1	apiVersion	76
7.33.2.2	author	77
7.33.2.3	dependencies	77
7.33.2.4	description	77
7.33.2.5	guid	77
7.33.2.6	name	77
7.33.2.7	numDependencies	77
7.33.2.8	version	77
7.33.2.9	web	77
7.34	tagQueueData Struct Reference	78
7.34.1	Detailed Description	78
7.34.2	Member Data Documentation	78
7.34.2.1	hash	78
7.34.2.2	isFileList	78
7.34.2.3	isManaged	79
7.34.2.4	location	79
7.34.2.5	object	79
7.34.2.6	size	79
7.34.2.7	target	79
7.35	tagStringData Struct Reference	79
7.35.1	Detailed Description	80
7.35.2	Member Data Documentation	80
7.35.2.1	in	80
7.35.2.2	out	80
7.36	tagTagData Struct Reference	80
7.36.1	Detailed Description	81
7.36.2	Member Data Documentation	81
7.36.2.1	isManaged	81

7.36.2.2	object	81
7.37	tagUserData Struct Reference	81
7.37.1	Detailed Description	82
7.37.2	Member Data Documentation	82
7.37.2.1	cid	82
7.37.2.2	hubHint	82
7.37.2.3	isManaged	82
7.37.2.4	isOp	82
7.37.2.5	nick	82
7.37.2.6	object	82
7.37.2.7	protocol	82
7.37.2.8	sid	82
7.38	dcapi::Hooks::Timer Struct Reference	82
7.38.1	Detailed Description	83
7.38.2	Member Function Documentation	83
7.38.2.1	onMinute	83
7.38.2.2	onSecond	83
7.39	dcapi::Hooks::UI Struct Reference	84
7.39.1	Detailed Description	84
7.39.2	Member Function Documentation	84
7.39.2.1	onChatCommand	84
7.39.2.2	onChatCommandPM	84
7.39.2.3	onChatDisplay	85
7.39.2.4	onChatTags	85
7.39.2.5	onCreated	85
7.40	dcapi::UI Class Reference	86
7.40.1	Detailed Description	87
7.40.2	Member Typedef Documentation	87
7.40.2.1	Command	87
7.40.3	Member Function Documentation	87
7.40.3.1	addCommand	87
7.40.3.2	commandCallback	88
7.40.3.3	handle	88
7.40.3.4	init	88

7.40.3.5	init	88
7.40.3.6	removeCommand	89
7.40.4	Member Data Documentation	89
7.40.4.1	commands	89
7.40.4.2	ui	89
7.41	dcapi::Hooks::Users Struct Reference	89
7.41.1	Detailed Description	90
7.41.2	Member Function Documentation	90
7.41.2.1	onOffline	90
7.41.2.2	onOnline	90
7.42	dcapi::Util Class Reference	90
7.42.1	Detailed Description	91
7.42.2	Member Function Documentation	91
7.42.2.1	fromUtf8	92
7.42.2.2	handle	92
7.42.2.3	init	92
7.42.2.4	init	92
7.42.2.5	toUtf8	92
7.42.3	Member Data Documentation	92
7.42.3.1	utils	92
8	File Documentation	95
8.1	Config.cpp File Reference	95
8.2	Config.h File Reference	96
8.3	Connections.cpp File Reference	97
8.4	Connections.h File Reference	98
8.5	Core.cpp File Reference	99
8.6	Core.h File Reference	99
8.7	Hooks.cpp File Reference	100
8.8	Hooks.h File Reference	101
8.9	Hubs.cpp File Reference	103
8.10	Hubs.h File Reference	103
8.11	Logger.cpp File Reference	105
8.12	Logger.h File Reference	105

8.13	main.cpp File Reference	107
8.13.1	Function Documentation	107
8.13.1.1	pluginInit	107
8.14	Plugin.cpp File Reference	108
8.15	Plugin.h File Reference	109
8.16	PluginDefs.h File Reference	109
8.16.1	Define Documentation	113
8.16.1.1	Bool	113
8.16.1.2	DCAPI	114
8.16.1.3	DCAPI_CORE_VER	114
8.16.1.4	DCEXP	114
8.16.1.5	DCIMP	114
8.16.1.6	DCINTF_CONFIG	114
8.16.1.7	DCINTF_CONFIG_VER	114
8.16.1.8	DCINTF_DCPP_CONNECTIONS	114
8.16.1.9	DCINTF_DCPP_CONNECTIONS_VER	114
8.16.1.10	DCINTF_DCPP_HUBS	114
8.16.1.11	DCINTF_DCPP_HUBS_VER	115
8.16.1.12	DCINTF_DCPP_QUEUE	115
8.16.1.13	DCINTF_DCPP_QUEUE_VER	115
8.16.1.14	DCINTF_DCPP_TAGGER	115
8.16.1.15	DCINTF_DCPP_TAGGER_VER	115
8.16.1.16	DCINTF_DCPP_UI	115
8.16.1.17	DCINTF_DCPP_UI_VER	115
8.16.1.18	DCINTF_DCPP_UTILS	115
8.16.1.19	DCINTF_DCPP_UTILS_VER	116
8.16.1.20	DCINTF_HOOKS	116
8.16.1.21	DCINTF_HOOKS_VER	116
8.16.1.22	DCINTF_LOGGING	116
8.16.1.23	DCINTF_LOGGING_VER	116
8.16.1.24	False	116
8.16.1.25	HOOK_CHAT_IN	116
8.16.1.26	HOOK_CHAT_OUT	116
8.16.1.27	HOOK_CHAT_PM_IN	117

8.16.1.28	HOOK_CHAT_PM_OUT	117
8.16.1.29	HOOK_HUB_OFFLINE	117
8.16.1.30	HOOK_HUB_ONLINE	117
8.16.1.31	HOOK_NETWORK_CONN_IN	117
8.16.1.32	HOOK_NETWORK_CONN_OUT	117
8.16.1.33	HOOK_NETWORK_HUB_IN	117
8.16.1.34	HOOK_NETWORK_HUB_OUT	117
8.16.1.35	HOOK_QUEUE_ADDED	118
8.16.1.36	HOOK_QUEUE_FINISHED	118
8.16.1.37	HOOK_QUEUE_MOVED	118
8.16.1.38	HOOK_QUEUE_REMOVED	118
8.16.1.39	HOOK_TIMER_MINUTE	118
8.16.1.40	HOOK_TIMER_SECOND	118
8.16.1.41	HOOK_UI_CHAT_COMMAND	118
8.16.1.42	HOOK_UI_CHAT_COMMAND_PM	118
8.16.1.43	HOOK_UI_CHAT_DISPLAY	119
8.16.1.44	HOOK_UI_CHAT_TAGS	119
8.16.1.45	HOOK_UI_CREATED	119
8.16.1.46	HOOK_USER_OFFLINE	119
8.16.1.47	HOOK_USER_ONLINE	119
8.16.1.48	True	119
8.16.2	Typedef Documentation	119
8.16.2.1	bBreak	119
8.16.2.2	CommandData	119
8.16.2.3	CommandDataPtr	119
8.16.2.4	ConfigBool	120
8.16.2.5	ConfigBoolPtr	120
8.16.2.6	ConfigInt	120
8.16.2.7	ConfigInt64	120
8.16.2.8	ConfigInt64Ptr	120
8.16.2.9	ConfigIntPtr	120
8.16.2.10	ConfigStr	120
8.16.2.11	ConfigStrPtr	120
8.16.2.12	ConfigType	120

8.16.2.13 ConfigValue	120
8.16.2.14 ConfigValuePtr	120
8.16.2.15 ConnectionData	120
8.16.2.16 ConnectionDataPtr	120
8.16.2.17 core	120
8.16.2.18 dcBool	120
8.16.2.19 DCConfig	120
8.16.2.20 DCConfigPtr	120
8.16.2.21 DCConnection	120
8.16.2.22 DCConnectionPtr	120
8.16.2.23 DCCore	120
8.16.2.24 DCCorePtr	120
8.16.2.25 DCHooks	120
8.16.2.26 DCHooksPtr	120
8.16.2.27 DCHub	121
8.16.2.28 DCHubPtr	121
8.16.2.29 DCInterface	121
8.16.2.30 DCInterfacePtr	121
8.16.2.31 DCLog	121
8.16.2.32 DCLogPtr	121
8.16.2.33 dcptr_t	121
8.16.2.34 DCQueue	121
8.16.2.35 DCQueuePtr	121
8.16.2.36 DCTagger	121
8.16.2.37 DCTaggerPtr	121
8.16.2.38 dctime_t	121
8.16.2.39 DCUI	121
8.16.2.40 DCUIPtr	121
8.16.2.41 DCUtils	121
8.16.2.42 DCUtilsPtr	121
8.16.2.43 hookHandle	121
8.16.2.44 HubData	121
8.16.2.45 HubDataPtr	121
8.16.2.46 intfHandle	122

8.16.2.47	MetaData	122
8.16.2.48	MetaDataPtr	122
8.16.2.49	MsgType	122
8.16.2.50	PathType	122
8.16.2.51	pCommon	122
8.16.2.52	pData	122
8.16.2.53	PluginState	122
8.16.2.54	ProtocolType	122
8.16.2.55	QueueData	122
8.16.2.56	QueueDataPtr	122
8.16.2.57	QueuePrio	122
8.16.2.58	StringData	122
8.16.2.59	StringDataPtr	122
8.16.2.60	subsHandle	122
8.16.2.61	TagData	122
8.16.2.62	TagDataPtr	122
8.16.2.63	UserData	122
8.16.2.64	UserDataPtr	122
8.16.3	Enumeration Type Documentation	123
8.16.3.1	tagConfigType	123
8.16.3.2	tagDCBool	123
8.16.3.3	tagMsgType	123
8.16.3.4	tagPathType	123
8.16.3.5	tagPluginState	124
8.16.3.6	tagProtocolType	124
8.16.3.7	tagQueuePrio	124
8.16.4	Function Documentation	124
8.16.4.1	Bool	124
8.16.4.2	void	124
8.17	Queue.cpp File Reference	124
8.18	Queue.h File Reference	125
8.19	resource.h File Reference	127
8.19.1	Define Documentation	127
8.19.1.1	VERSION_INFO	127

8.20	stdafx.cpp File Reference	127
8.21	stdafx.h File Reference	127
8.21.1	Typedef Documentation	128
8.21.1.1	tstring	128
8.22	Tagger.cpp File Reference	129
8.23	Tagger.h File Reference	129
8.24	UI.cpp File Reference	131
8.25	UI.h File Reference	131
8.26	Util.cpp File Reference	133
8.27	Util.h File Reference	133
8.28	version.h File Reference	135
8.28.1	Define Documentation	135
8.28.1.1	PLUGIN_AUTHOR	135
8.28.1.2	PLUGIN_DESC	135
8.28.1.3	PLUGIN_GUID	136
8.28.1.4	PLUGIN_NAME	136
8.28.1.5	PLUGIN_VERSION	136
8.28.1.6	PLUGIN_WEB	136

Chapter 1

Directory Hierarchy

1.1 Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

pluginsdk	9
src	10

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

dcapi	11
-----------------------------	----

Chapter 3

Class Index

3.1 Class List

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

dcapi::Hooks::Chat	13
dcapi::Config	15
dcapi::Connections	20
dcapi::Core	22
DCUI	26
dcapi::Hooks	27
dcapi::Hubs	34
dcapi::Hooks::Hubs	36
dcapi::Logger	37
dcapi::Hooks::Network	39
Plugin	41
dcapi::Queue	43
dcapi::Hooks::Queue	45
tagCommandData	47
tagConfigBool	48
tagConfigInt	49
tagConfigInt64	50
tagConfigStr	51
tagConfigValue	52
tagConnectionData	52
tagDCConfig	54
tagDCConnection	56
tagDCCore	58
tagDCHooks	60
tagDCHub	61
tagDCInterface	64
tagDCLog	65
tagDCQueue	66
tagDCTagger	68

tagDCUtils	69
dcapi::Tagger	72
tagHubData	74
tagMetaData	75
tagQueueData	78
tagStringData	79
tagTagData	80
tagUserData	81
dcapi::Hooks::Timer	82
dcapi::Hooks::UI	84
dcapi::UI	86
dcapi::Hooks::Users	89
dcapi::Util	90

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

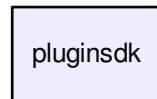
Config.cpp	95
Config.h	96
Connections.cpp	97
Connections.h	98
Core.cpp	99
Core.h	99
Hooks.cpp	100
Hooks.h	101
Hubs.cpp	103
Hubs.h	103
Logger.cpp	105
Logger.h	105
main.cpp	107
Plugin.cpp	108
Plugin.h	109
PluginDefs.h	109
Queue.cpp	124
Queue.h	125
resource.h	127
stdafx.cpp	127
stdafx.h	127
Tagger.cpp	129
Tagger.h	129
UI.cpp	131
UI.h	131
Util.cpp	133
Util.h	133
version.h	135

Chapter 5

Directory Documentation

5.1 /cygdrive/h/Dev/DC++/pluginsdk/cpp/pluginsdk/ Directory Reference

Directory dependency graph for /cygdrive/h/Dev/DC++/pluginsdk/cpp/pluginsdk/:



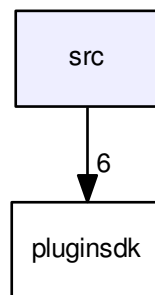
Files

- file [Config.cpp](#)
- file [Config.h](#)
- file [Connections.cpp](#)
- file [Connections.h](#)
- file [Core.cpp](#)
- file [Core.h](#)
- file [Hooks.cpp](#)
- file [Hooks.h](#)
- file [Hubs.cpp](#)
- file [Hubs.h](#)
- file [Logger.cpp](#)
- file [Logger.h](#)

- file [PluginDefs.h](#)
- file [Queue.cpp](#)
- file [Queue.h](#)
- file [Tagger.cpp](#)
- file [Tagger.h](#)
- file [UI.cpp](#)
- file [UI.h](#)
- file [Util.cpp](#)
- file [Util.h](#)

5.2 /cygdrive/h/Dev/DC++/pluginsdk/cpp/src/ Directory Reference

Directory dependency graph for /cygdrive/h/Dev/DC++/pluginsdk/cpp/src/:



Files

- file [main.cpp](#)
- file [Plugin.cpp](#)
- file [Plugin.h](#)
- file [resource.h](#)
- file [stdafx.cpp](#)
- file [stdafx.h](#)
- file [version.h](#)

Chapter 6

Namespace Documentation

6.1 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 7

Class Documentation

7.1 dcapi::Hooks::Chat Struct Reference

```
#include <Hooks.h>
```

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)

7.1.1 Detailed Description

Definition at line 54 of file Hooks.h.

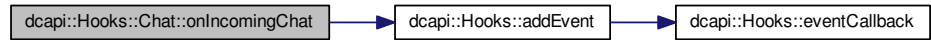
7.1.2 Member Function Documentation

7.1.2.1 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:

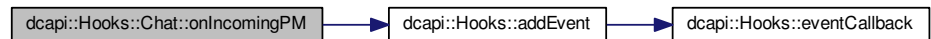


7.1.2.2 `void dcapi::Hooks::Chat::onIncomingPM (function< bool(UserDataPtr, char *, bool &)> f) [static]`

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:

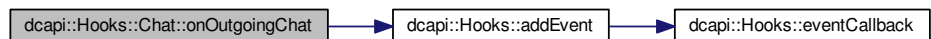


7.1.2.3 `void dcapi::Hooks::Chat::onOutgoingChat (function< bool(HubDataPtr, char *, bool &)> f) [static]`

Definition at line 42 of file Hooks.cpp.

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

Here is the call graph for this function:

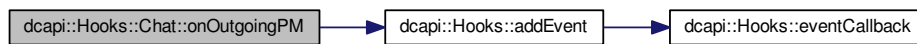


7.1.2.4 `void dcapi::Hooks::Chat::onOutgoingPM (function< bool(UserDataPtr, char *, bool &)> f) [static]`

Definition at line 50 of file Hooks.cpp.

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

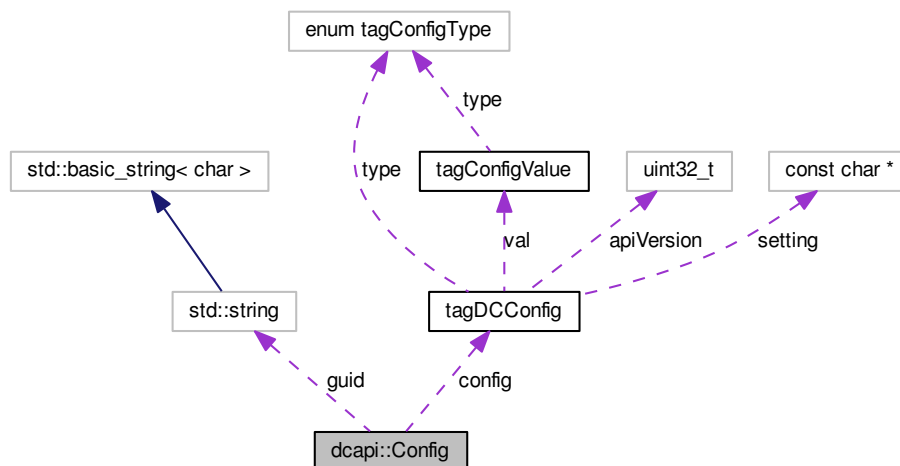
Here is the call graph for this function:



7.2 dcapi::Config Class Reference

```
#include <Config.h>
```

Collaboration diagram for `dcapi::Config`:



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 `int64_t` `getInt64Config` (const char *name)
- static `int32_t` `getIntConfig` (const char *name)

- static string [getPath](#) ([PathType](#) type)
- static [DCConfigPtr](#) [handle](#) ()
- static void [init](#) ([DCConfigPtr](#) coreConfig, string pluginGuid)
- static bool [init](#) (string pluginGuid)
- static void [setConfig](#) (const char *name, const string &value)
- static void [setConfig](#) (const char *name, int32_t value)
- static void [setConfig](#) (const char *name, const char *value)
- static void [setConfig](#) (const char *name, bool 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](#)

7.2.1 Detailed Description

Definition at line 32 of file Config.h.

7.2.2 Member Function Documentation

7.2.2.1 void dcapi::Config::freeCoreConfig ([ConfigValuePtr](#) *value*) [static]

Definition at line 50 of file Config.cpp.

References [config](#).

7.2.2.2 bool dcapi::Config::getBoolConfig (const char * *name*) [static]

Definition at line 45 of file Config.cpp.

References [CFG_TYPE_BOOL](#).

7.2.2.3 string dcapi::Config::getConfig (const char * *name*) [static]

Definition at line 44 of file Config.cpp.

References [CFG_TYPE_STRING](#).

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

Definition at line 69 of file Config.h.

References config, and guid.

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

Definition at line 49 of file Config.cpp.

References CFG_TYPE_UNKNOWN, and config.

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

Definition at line 47 of file Config.cpp.

References CFG_TYPE_INT64.

7.2.2.7 `int32_t dcapi::Config::getIntConfig (const char * name)` [static]

Definition at line 46 of file Config.cpp.

References CFG_TYPE_INT.

7.2.2.8 `string dcapi::Config::getPath (PathType type)` [static]

Definition at line 52 of file Config.cpp.

References config, and tagDCConfig::get_path().

Here is the call graph for this function:



7.2.2.9 `DConfigPtr dcapi::Config::handle ()` [static]

Definition at line 36 of file Config.cpp.

References config.

7.2.2.10 `void dcapi::Config::init (DCConfigPtr coreConfig, string pluginGuid)`
[static]

Definition at line 35 of file Config.cpp.

References config, and guid.

7.2.2.11 `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:



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

Definition at line 41 of file Config.cpp.

References CFG_TYPE_INT.

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

Definition at line 42 of file Config.cpp.

References CFG_TYPE_INT64.

7.2.2.14 `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.

7.2.2.15 `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:



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

Definition at line 64 of file Config.h.

References config, and guid.

7.2.2.17 `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:



7.2.3 Member Data Documentation

7.2.3.1 `DConfigPtr dcapi::Config::config [static, private]`

Definition at line 59 of file Config.h.

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

7.2.3.2 `string dcapi::Config::guid` [static, private]

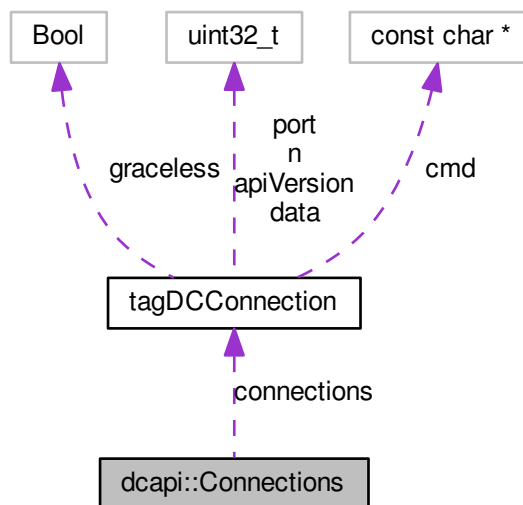
Definition at line 61 of file `Config.h`.

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

7.3 `dcapi::Connections` Class Reference

```
#include <Connections.h>
```

Collaboration diagram for `dcapi::Connections`:



Static Public Member Functions

- static `DCCConnectionPtr handle ()`
- static `bool init ()`
- static `void init (DCCConnectionPtr coreConnections)`

Static Private Attributes

- static [DCConnectionPtr connections](#)

7.3.1 Detailed Description

Definition at line 28 of file Connections.h.

7.3.2 Member Function Documentation

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

Definition at line 35 of file Connections.cpp.

References [connections](#).

7.3.2.2 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:



7.3.2.3 `void dcap::Connections::init (DCConnectionPtr coreConnections) [static]`

Definition at line 34 of file `Connections.cpp`.

References `connections`.

7.3.3 Member Data Documentation

7.3.3.1 `DCConnectionPtr dcap::Connections::connections [static, private]`

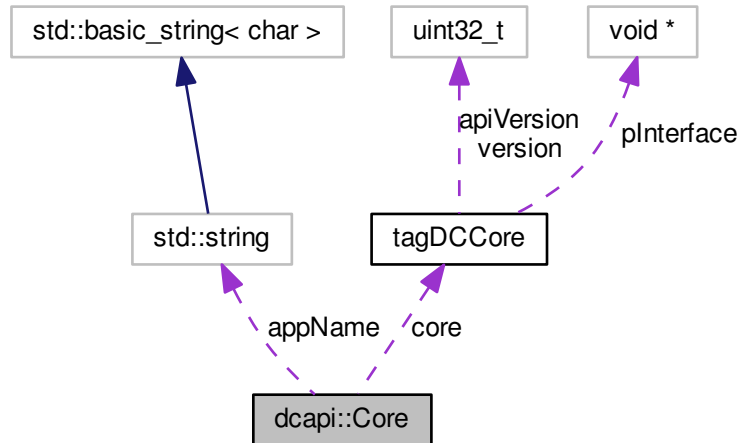
Definition at line 36 of file `Connections.h`.

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

7.4 dcap::Core Class Reference

```
#include <Core.h>
```

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](#)

7.4.1 Detailed Description

Definition at line 32 of file Core.h.

7.4.2 Member Function Documentation

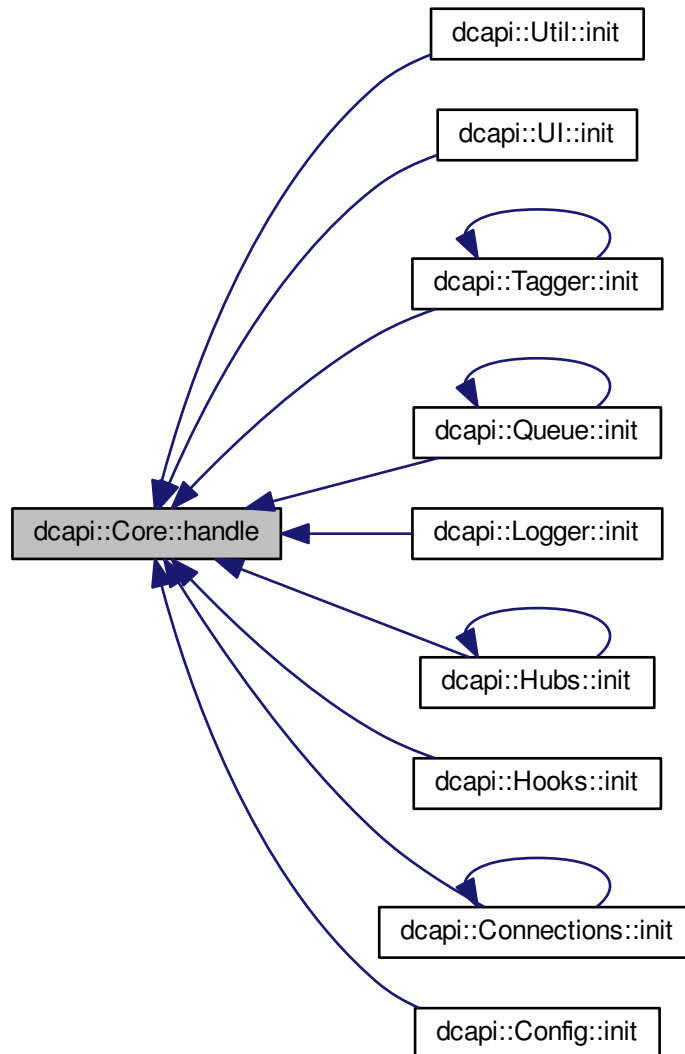
7.4.2.1 DCCorePtr dcapi::Core::handle () [static]

Definition at line 33 of file Core.cpp.

References core.

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

Here is the caller graph for this function:

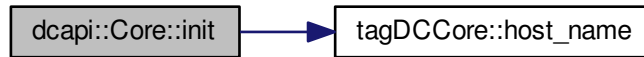


7.4.2.2 void dcapi::Core::init (DCCorePtr corePtr) [static]

Definition at line 28 of file Core.cpp.

References `appName`, and `tagDCCore::host_name()`.

Here is the call graph for this function:



7.4.3 Member Data Documentation

7.4.3.1 `string dcap::Core::appName` [static]

Definition at line 38 of file Core.h.

Referenced by `init()`.

7.4.3.2 `DCCorePtr dcap::Core::core` [static, private]

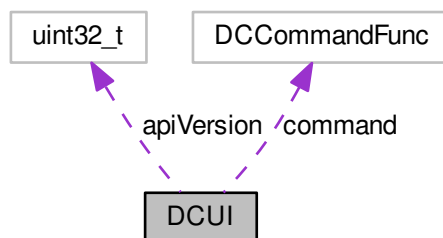
Definition at line 41 of file Core.h.

Referenced by `handle()`.

7.5 DCUI Struct Reference

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

Collaboration diagram for DCUI:



Public Member Functions

- [void](#) (DCAPI *add_command)(const char *name
- [void](#) (DCAPI *play_sound)(const char *path)
- [void](#) (DCAPI *remove_command)(const char *name)

Public Attributes

- uint32_t [apiVersion](#)
- DCCCommandFunc [command](#)

7.5.1 Detailed Description

Definition at line 437 of file PluginDefs.h.

7.5.2 Member Function Documentation

7.5.2.1 DCUI::void (DCAPI * *add_command*) const

7.5.2.2 DCUI::void (DCAPI * *play_sound*) const

7.5.2.3 DCUI::void (DCAPI * *remove_command*) const

7.5.3 Member Data Documentation

7.5.3.1 uint32_t DCUI::apiVersion

Definition at line 439 of file PluginDefs.h.

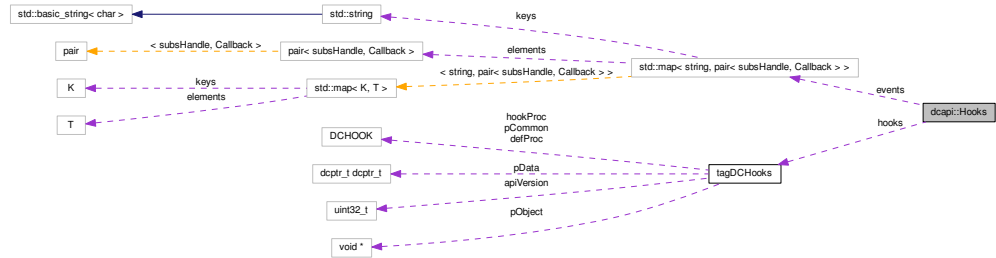
7.5.3.2 DCCCommandFunc DCUI::command

Definition at line 441 of file PluginDefs.h.

7.6 dcapi::Hooks Class Reference

```
#include <Hooks.h>
```

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 void [init](#) ([DCHooksPtr](#) coreHooks)
- static bool [init](#) ()
- 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

7.6.1 Detailed Description

Definition at line 38 of file Hooks.h.

7.6.2 Member Typedef Documentation

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

Definition at line 103 of file Hooks.h.

7.6.3 Member Function Documentation

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

Definition at line 155 of file Hooks.cpp.

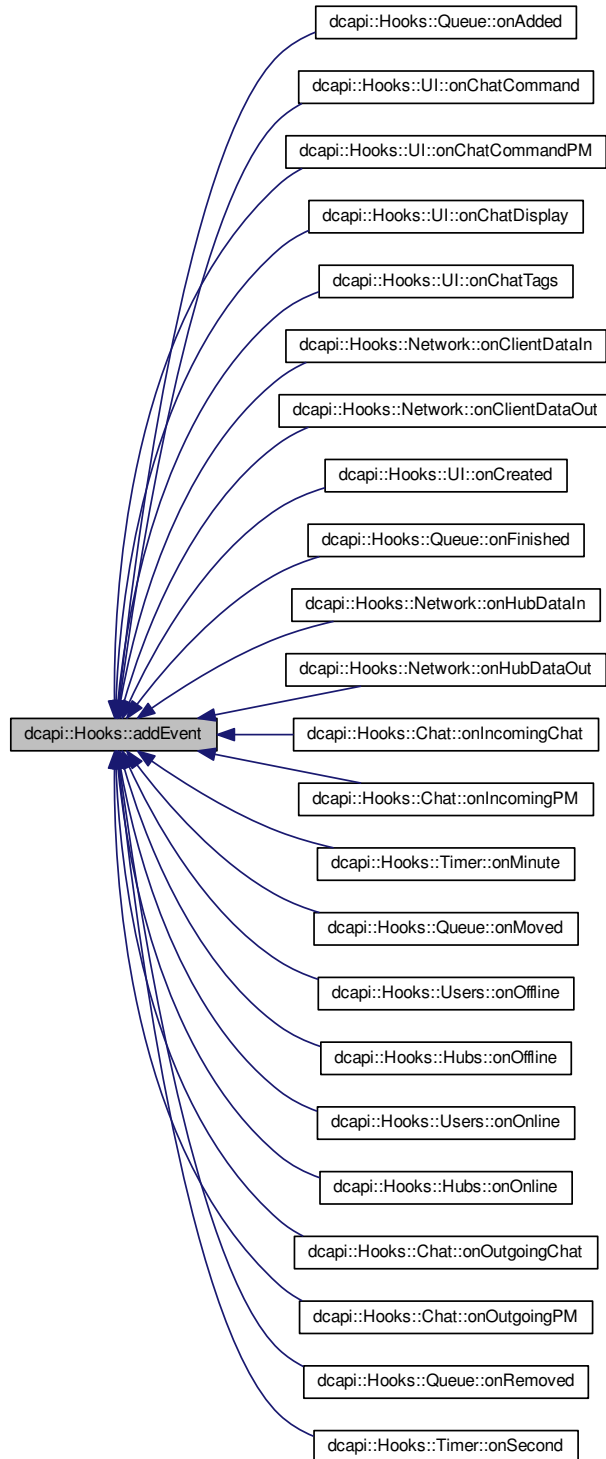
References eventCallback(), events, and hooks.

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::Users::onOffline(), dcapi::Hooks::Hubs::onOffline(), dcapi::Hooks::Users::onOnline(), dcapi::Hooks::Hubs::onOnline(), dcapi::Hooks::Chat::onOutgoingChat(), dcapi::Hooks::Chat::onOutgoingPM(), dcapi::Hooks::Queue::onRemoved(), and dcapi::Hooks::Timer::onSecond().

Here is the call graph for this function:



Here is the caller graph for this function:



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

Definition at line 141 of file Hooks.cpp.

References events, and hooks.

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

Definition at line 137 of file Hooks.cpp.

References events.

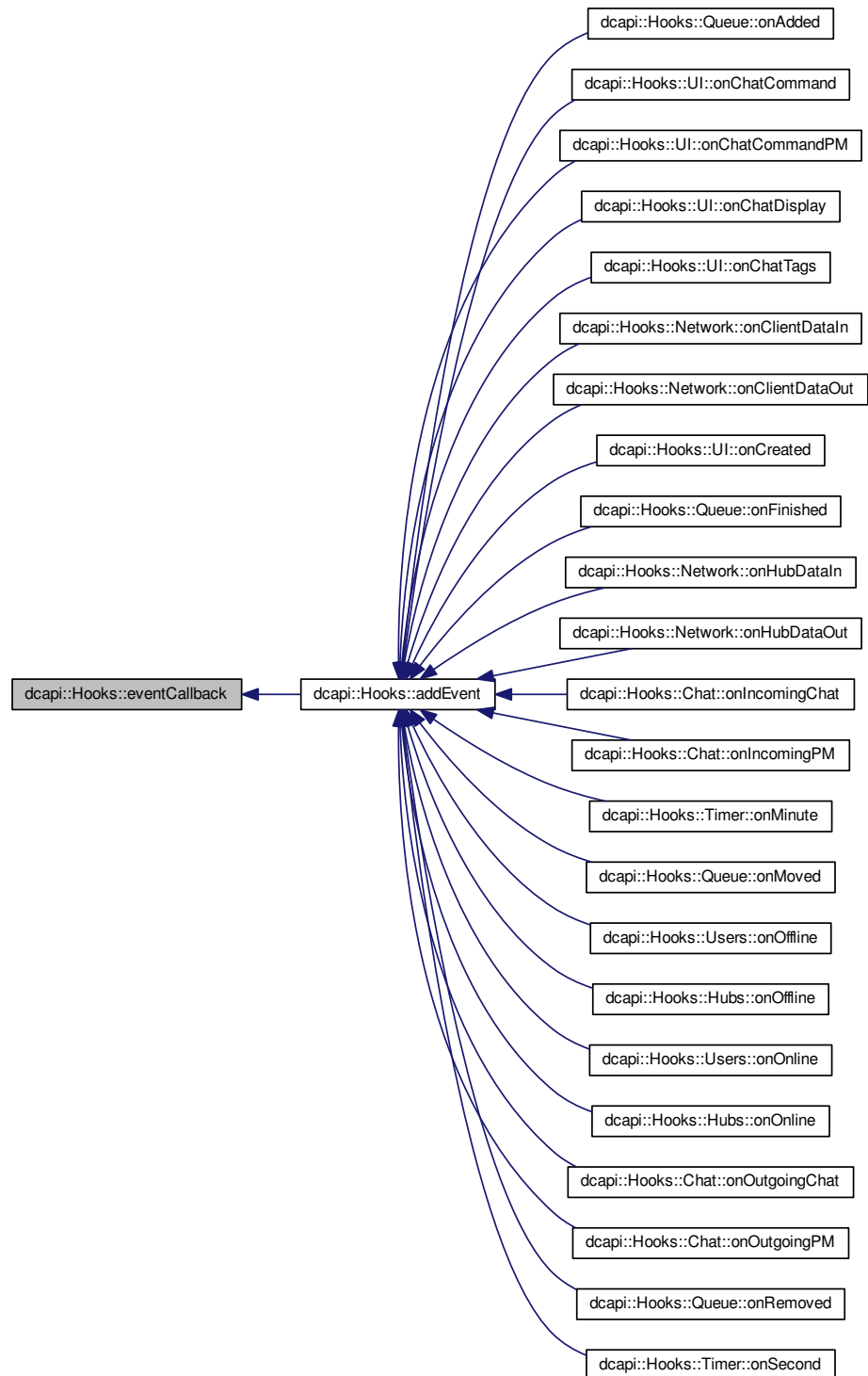
7.6.3.4 Bool DCAPI dcapi::Hooks::eventCallback (dcptr_t *pObject*, dcptr_t *pData*, dcptr_t *pCommon*, Bool * *bBreak*) [static, private]

Definition at line 161 of file Hooks.cpp.

References events, False, and True.

Referenced by addEvent().

Here is the caller graph for this function:



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

Definition at line 36 of file Hooks.cpp.

References hooks.

7.6.3.6 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:



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

Definition at line 35 of file Hooks.cpp.

References hooks.

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

Definition at line 147 of file Hooks.cpp.

References events, and hooks.

7.6.4 Member Data Documentation

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

Definition at line 109 of file Hooks.h.

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

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

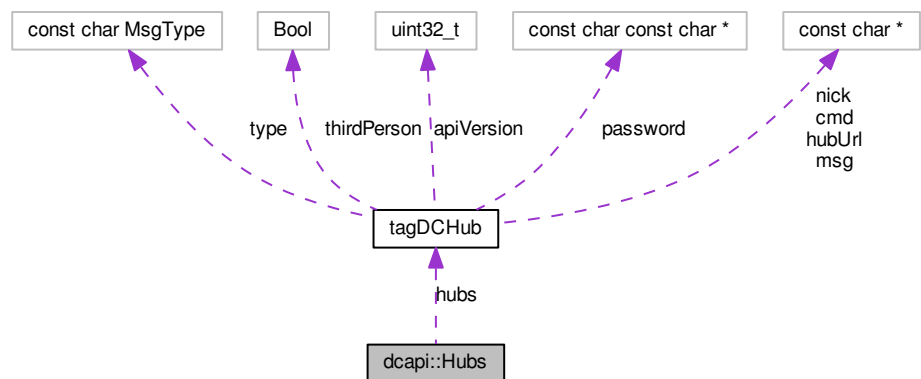
Definition at line 107 of file Hooks.h.

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

7.7 dcapi::Hubs Class Reference

```
#include <Hubs.h>
```

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`

7.7.1 Detailed Description

Definition at line 28 of file `Hubs.h`.

7.7.2 Member Function Documentation

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

Definition at line 35 of file Hubs.cpp.

References hubs.

7.7.2.2 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:



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

Definition at line 34 of file Hubs.cpp.

References hubs.

7.7.3 Member Data Documentation

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

Definition at line 36 of file Hubs.h.

Referenced by handle(), and init().

7.8 dcapi::Hooks::Hubs Struct Reference

```
#include <Hooks.h>
```

Static Public Member Functions

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

7.8.1 Detailed Description

Definition at line 66 of file Hooks.h.

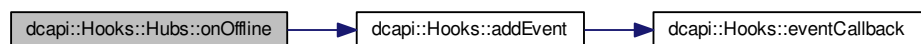
7.8.2 Member Function Documentation

7.8.2.1 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:

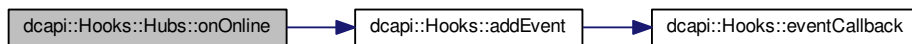


7.8.2.2 void dcapi::Hooks::Hubs::onOnline (function< bool([HubDataPtr](#), bool &)> f) [static]

Definition at line 64 of file Hooks.cpp.

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

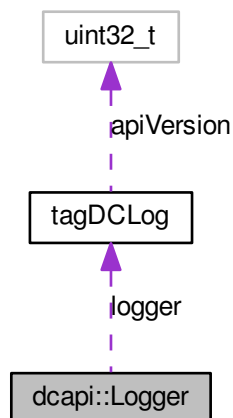
Here is the call graph for this function:



7.9 dcapi::Logger Class Reference

```
#include <Logger.h>
```

Collaboration diagram for `dcapi::Logger`:



Static Public Member Functions

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

Static Private Attributes

- static [DCLogPtr logger](#)

7.9.1 Detailed Description

Definition at line 32 of file `Logger.h`.

7.9.2 Member Function Documentation

7.9.2.1 `DCLogPtr dcapi::Logger::handle()` [static]

Definition at line 35 of file `Logger.cpp`.

References `logger`.

7.9.2.2 `void dcapi::Logger::init(DCLogPtr coreLogger)` [static]

Definition at line 34 of file `Logger.cpp`.

References `logger`.

7.9.2.3 `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:



7.9.2.4 `void dcapi::Logger::log(const string & message)` [static]

Definition at line 37 of file `Logger.cpp`.

References `logger`.

7.9.3 Member Data Documentation

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

Definition at line 42 of file Logger.h.

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

7.10 dcapi::Hooks::Network Struct Reference

```
#include <Hooks.h>
```

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)

7.10.1 Detailed Description

Definition at line 76 of file Hooks.h.

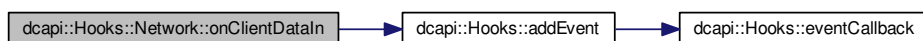
7.10.2 Member Function Documentation

7.10.2.1 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:

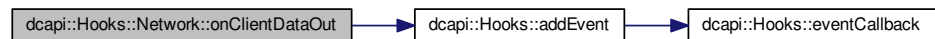


7.10.2.2 `void dcapi::Hooks::Network::onClientDataOut (function< bool(ConnectionDataPtr, char *, bool &)> f) [static]`

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:

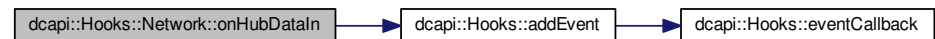


7.10.2.3 `void dcapi::Hooks::Network::onHubDataIn (function< bool(HubDataPtr, char *, bool &)> f) [static]`

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:

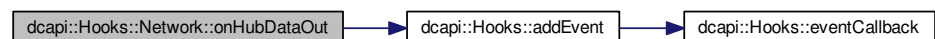


7.10.2.4 `void dcapi::Hooks::Network::onHubDataOut (function< bool(HubDataPtr, char *, bool &)> f) [static]`

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:



7.11 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)
- [Plugin](#) ()
- [~Plugin](#) ()

7.11.1 Detailed Description

Definition at line 24 of file [Plugin.h](#).

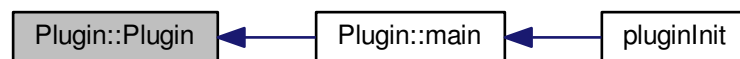
7.11.2 Constructor & Destructor Documentation

7.11.2.1 [Plugin::Plugin](#) () [[private](#)]

Definition at line 37 of file [Plugin.cpp](#).

Referenced by [main\(\)](#).

Here is the caller graph for this function:



7.11.2.2 [Plugin::~~Plugin](#) () [[private](#)]

Definition at line 40 of file [Plugin.cpp](#).

7.11.3 Member Function Documentation

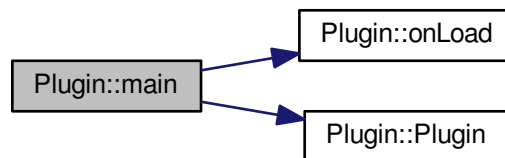
7.11.3.1 Bool DCAPI Plugin::main (PluginState *state*, DCCorePtr *core*, dcptr_t) [static]

Definition at line 44 of file Plugin.cpp.

References False, ON_INSTALL, ON_LOAD, ON_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:



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

Definition at line 70 of file Plugin.cpp.

References PLUGIN_GUID.

Referenced by main().

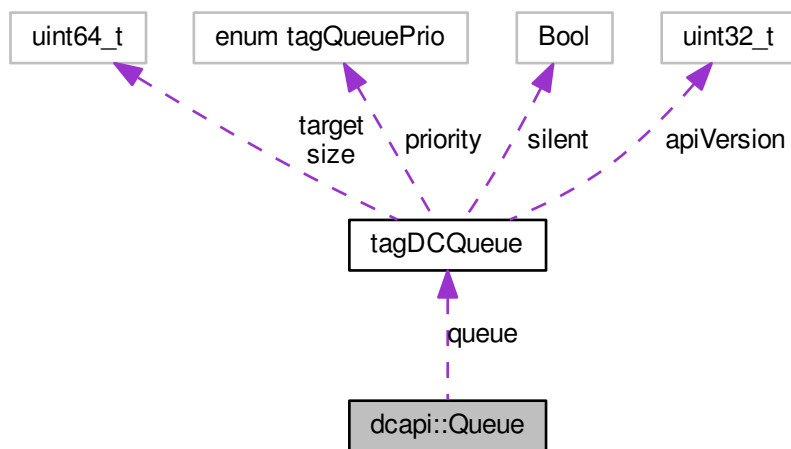
Here is the caller graph for this function:



7.12 dcapi::Queue Class Reference

```
#include <Queue.h>
```

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](#)

7.12.1 Detailed Description

Definition at line 28 of file Queue.h.

7.12.2 Member Function Documentation

7.12.2.1 `DCQueuePtr dcapi::Queue::handle () [static]`

Definition at line 35 of file Queue.cpp.

References [queue](#).

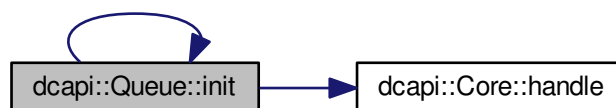
7.12.2.2 `bool dcapi::Queue::init () [static]`

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:



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

Definition at line 34 of file Queue.cpp.

References queue.

7.12.3 Member Data Documentation

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

Definition at line 36 of file Queue.h.

Referenced by handle(), and init().

7.13 dcapi::Hooks::Queue Struct Reference

```
#include <Hooks.h>
```

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)

7.13.1 Detailed Description

Definition at line 83 of file Hooks.h.

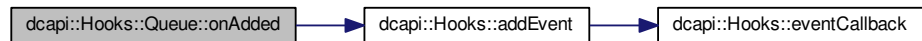
7.13.2 Member Function Documentation

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

Definition at line 99 of file Hooks.cpp.

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

Here is the call graph for this function:

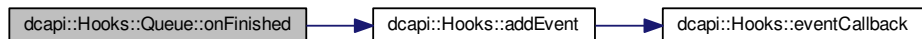


7.13.2.2 `void dcapi::Hooks::Queue::onFinished (function< bool(QueueDataPtr, bool &)> f)`
[static]

Definition at line 111 of file Hooks.cpp.

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

Here is the call graph for this function:

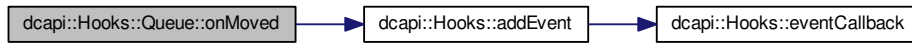


7.13.2.3 `void dcapi::Hooks::Queue::onMoved (function< bool(QueueDataPtr, bool &)> f)`
[static]

Definition at line 103 of file Hooks.cpp.

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

Here is the call graph for this function:

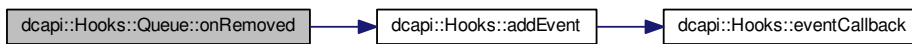


7.13.2.4 `void dcapi::Hooks::Queue::onRemoved (function< bool(QueueDataPtr, bool &> f) [static]`

Definition at line 107 of file Hooks.cpp.

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

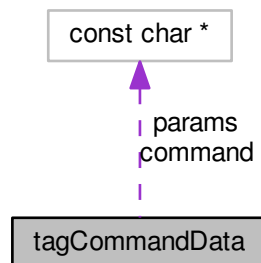
Here is the call graph for this function:



7.14 tagCommandData Struct Reference

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

Collaboration diagram for tagCommandData:



Public Attributes

- const char * [command](#)
- const char * [params](#)

7.14.1 Detailed Description

Definition at line 209 of file PluginDefs.h.

7.14.2 Member Data Documentation

7.14.2.1 const char* tagCommandData::command

Definition at line 210 of file PluginDefs.h.

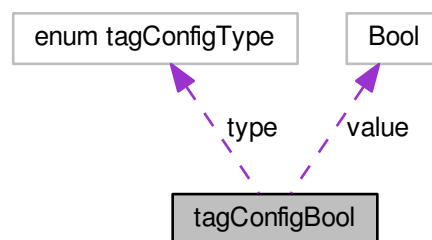
7.14.2.2 const char* tagCommandData::params

Definition at line 211 of file PluginDefs.h.

7.15 tagConfigBool Struct Reference

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

Collaboration diagram for tagConfigBool:



Public Attributes

- [ConfigType](#) type
- [Bool](#) value

7.15.1 Detailed Description

Definition at line 191 of file PluginDefs.h.

7.15.2 Member Data Documentation

7.15.2.1 ConfigType tagConfigBool::type

Definition at line 192 of file PluginDefs.h.

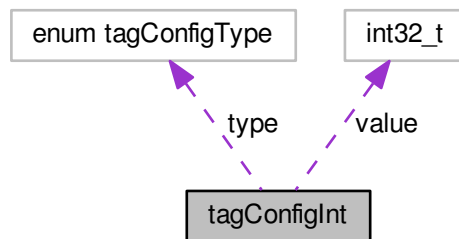
7.15.2.2 Bool tagConfigBool::value

Definition at line 193 of file PluginDefs.h.

7.16 tagConfigInt Struct Reference

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

Collaboration diagram for tagConfigInt:



Public Attributes

- [ConfigType type](#)
- [int32_t value](#)

7.16.1 Detailed Description

Definition at line 185 of file PluginDefs.h.

7.16.2 Member Data Documentation

7.16.2.1 ConfigType tagConfigInt::type

Definition at line 186 of file PluginDefs.h.

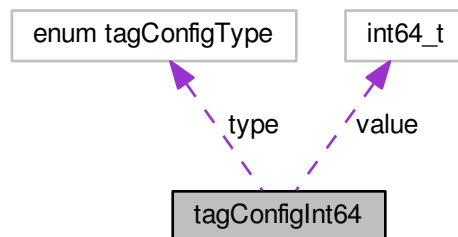
7.16.2.2 int32_t tagConfigInt::value

Definition at line 187 of file PluginDefs.h.

7.17 tagConfigInt64 Struct Reference

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

Collaboration diagram for tagConfigInt64:



Public Attributes

- [ConfigType type](#)
- [int64_t value](#)

7.17.1 Detailed Description

Definition at line 197 of file PluginDefs.h.

7.17.2 Member Data Documentation

7.17.2.1 ConfigType tagConfigInt64::type

Definition at line 198 of file PluginDefs.h.

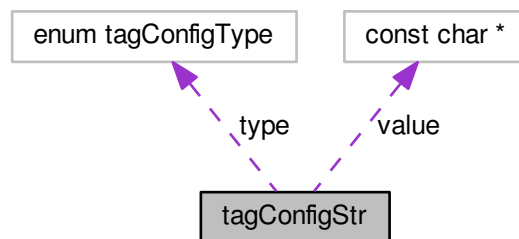
7.17.2.2 int64_t tagConfigInt64::value

Definition at line 199 of file PluginDefs.h.

7.18 tagConfigStr Struct Reference

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

Collaboration diagram for tagConfigStr:



Public Attributes

- [ConfigType type](#)
- `const char *` [value](#)

7.18.1 Detailed Description

Definition at line 179 of file PluginDefs.h.

7.18.2 Member Data Documentation

7.18.2.1 ConfigType tagConfigStr::type

Definition at line 180 of file PluginDefs.h.

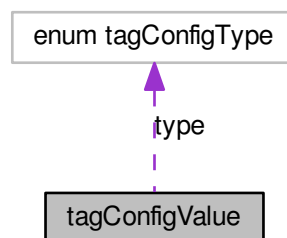
7.18.2.2 `const char* tagConfigStr::value`

Definition at line 181 of file PluginDefs.h.

7.19 tagConfigValue Struct Reference

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

Collaboration diagram for tagConfigValue:



Public Attributes

- [ConfigType type](#)

7.19.1 Detailed Description

Definition at line 174 of file PluginDefs.h.

7.19.2 Member Data Documentation

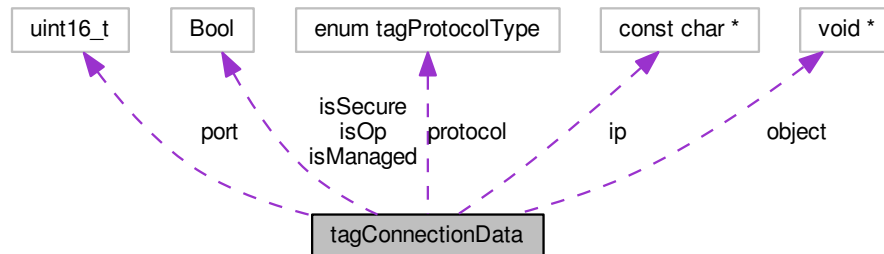
7.19.2.1 `ConfigType tagConfigValue::type`

Definition at line 175 of file PluginDefs.h.

7.20 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`

7.20.1 Detailed Description

Definition at line 239 of file `PluginDefs.h`.

7.20.2 Member Data Documentation

7.20.2.1 `const char* tagConnectionData::ip`

Definition at line 240 of file `PluginDefs.h`.

7.20.2.2 `Bool tagConnectionData::isManaged`

Definition at line 246 of file `PluginDefs.h`.

7.20.2.3 `Bool tagConnectionData::isOp`

Definition at line 244 of file `PluginDefs.h`.

7.20.2.4 Bool tagConnectionData::isSecure

Definition at line 245 of file PluginDefs.h.

7.20.2.5 dcptr_t tagConnectionData::object

Definition at line 241 of file PluginDefs.h.

7.20.2.6 uint16_t tagConnectionData::port

Definition at line 242 of file PluginDefs.h.

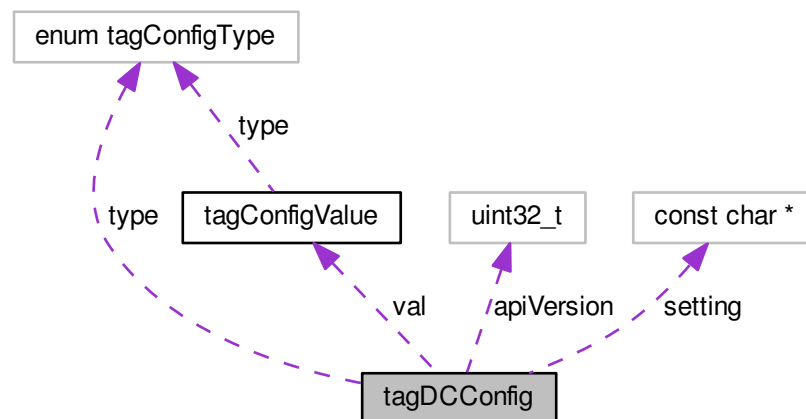
7.20.2.7 ProtocolType tagConnectionData::protocol

Definition at line 243 of file PluginDefs.h.

7.21 tagDCConfig Struct Reference

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

Collaboration diagram for tagDCConfig:



Public Member Functions

- [ConfigValuePtr](#) (DCAPI *get_cfg)(const char *guid
- [ConfigValuePtr](#) (DCAPI *copy)(const [ConfigValuePtr](#) val)
- const char *DCAPI [get_path](#) ([PathType](#) type)
- 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

7.21.1 Detailed Description

Definition at line 328 of file PluginDefs.h.

7.21.2 Member Function Documentation

7.21.2.1 `tagDCConfig::ConfigValuePtr (DCAPI * get_cfg) const`

7.21.2.2 `tagDCConfig::ConfigValuePtr (DCAPI * copy) const`

7.21.2.3 `const char* DCAPI tagDCConfig::get_path (PathType type)`

Referenced by `dcapi::Config::getPath()`.

Here is the caller graph for this function:



7.21.2.4 `tagDCConfig::void (DCAPI * set_cfg) const`

7.21.2.5 `tagDCConfig::void (DCAPI * release)`

7.21.3 Member Data Documentation

7.21.3.1 `uint32_t tagDCConfig::apiVersion`

Definition at line 330 of file PluginDefs.h.

7.21.3.2 `const char * tagDCConfig::setting`

Definition at line 334 of file PluginDefs.h.

7.21.3.3 `const char ConfigType tagDCConfig::type`

Definition at line 335 of file PluginDefs.h.

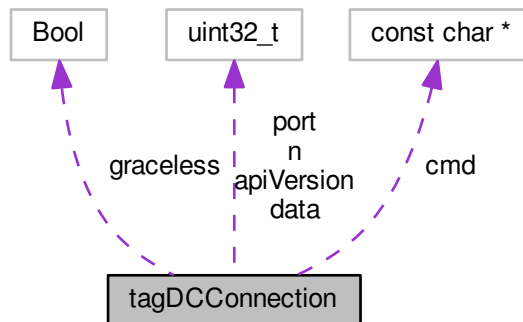
7.21.3.4 `const char ConfigValuePtr tagDCConfig::val`

Definition at line 334 of file PluginDefs.h.

7.22 tagDCConnection Struct Reference

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

Collaboration diagram for tagDCConnection:



Public Member Functions

- `UserDataPtr` (DCAPI `*get_user`)(`ConnectionDataPtr` hConn)

- `void` (DCAPI *send_protocol_cmd)(ConnectionDataPtr hConn
- `void` (DCAPI *terminate_conn)(ConnectionDataPtr hConn
- `void` (DCAPI *send_udp_data)(const char *ip

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

7.22.1 Detailed Description

Definition at line 352 of file PluginDefs.h.

7.22.2 Member Function Documentation

7.22.2.1 `tagDCConnection::UserDataPtr` (DCAPI * *get_user*)

7.22.2.2 `tagDCConnection::void` (DCAPI * *send_protocol_cmd*)

7.22.2.3 `tagDCConnection::void` (DCAPI * *terminate_conn*)

7.22.2.4 `tagDCConnection::void` (DCAPI * *send_udp_data*) const

7.22.3 Member Data Documentation

7.22.3.1 `uint32_t` tagDCConnection::apiVersion

Definition at line 354 of file PluginDefs.h.

7.22.3.2 `const char*` tagDCConnection::cmd

Definition at line 357 of file PluginDefs.h.

7.22.3.3 `uint32_t` dcptr_t tagDCConnection::data

Definition at line 356 of file PluginDefs.h.

7.22.3.4 `Bool` tagDCConnection::graceless

Definition at line 358 of file PluginDefs.h.

7.22.3.5 uint32_t dcptr_t size_t tagDCConnection::n

Definition at line 356 of file PluginDefs.h.

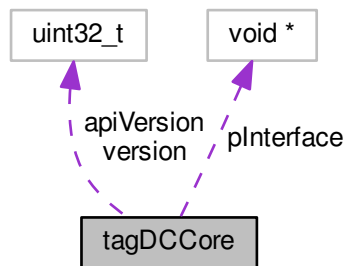
7.22.3.6 uint32_t tagDCConnection::port

Definition at line 356 of file PluginDefs.h.

7.23 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](#)

7.23.1 Detailed Description

Definition at line 286 of file PluginDefs.h.

7.23.2 Member Function Documentation

7.23.2.1 tagDCCore::Bool (DCAPI * *release_interface*)

7.23.2.2 tagDCCore::Bool (DCAPI * *has_plugin*) const

7.23.2.3 tagDCCore::DCInterfacePtr (DCAPI * *query_interface*) const

7.23.2.4 const char* DCAPI tagDCCore::host_name ()

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

Here is the caller graph for this function:



7.23.2.5 tagDCCore::intfHandle (DCAPI * *register_interface*) const

7.23.3 Member Data Documentation

7.23.3.1 uint32_t tagDCCore::apiVersion

Definition at line 288 of file PluginDefs.h.

7.23.3.2 dcptr_t tagDCCore::pInterface

Definition at line 291 of file PluginDefs.h.

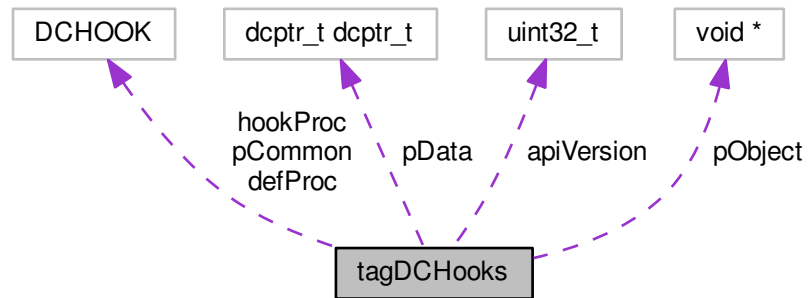
7.23.3.3 uint32_t tagDCCore::version

Definition at line 292 of file PluginDefs.h.

7.24 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](#)

7.24.1 Detailed Description

Definition at line 311 of file PluginDefs.h.

7.24.2 Member Function Documentation

7.24.2.1 tagDCHooks::Bool (DCAPI * *destroy_hook*)

7.24.2.2 tagDCHooks::Bool (DCAPI * *run_hook*)

7.24.2.3 tagDCHooks::hookHandle (DCAPI * *create_hook*) const

7.24.2.4 tagDCHooks::size_t (DCAPI * *release_hook*)

7.24.2.5 tagDCHooks::subsHandle (DCAPI * *bind_hook*) const

7.24.3 Member Data Documentation

7.24.3.1 uint32_t tagDCHooks::apiVersion

Definition at line 313 of file PluginDefs.h.

7.24.3.2 DHOOK tagDCHooks::defProc

Definition at line 316 of file PluginDefs.h.

7.24.3.3 DHOOK tagDCHooks::hookProc

Definition at line 320 of file PluginDefs.h.

7.24.3.4 DHOOK void* tagDCHooks::pCommon

Definition at line 320 of file PluginDefs.h.

7.24.3.5 dcptr_t dcptr_t tagDCHooks::pData

Definition at line 321 of file PluginDefs.h.

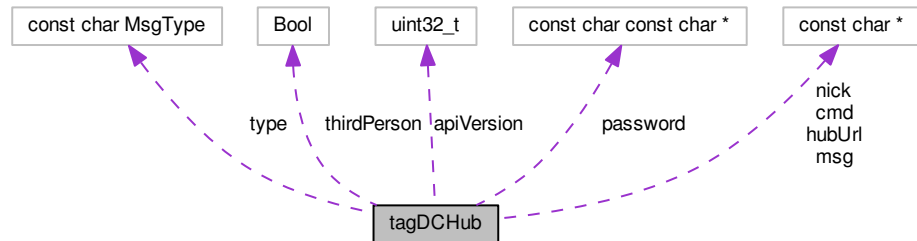
7.24.3.6 dcptr_t tagDCHooks::pObject

Definition at line 321 of file PluginDefs.h.

7.25 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 *emulate_protocol_cmd)(`HubDataPtr` hHub)
- `void` (DCAPI *local_message)(`HubDataPtr` hHub)
- `void` (DCAPI *remove_hub)(`HubDataPtr` hHub)
- `void` (DCAPI *send_protocol_cmd)(`HubDataPtr` hHub)
- `void` (DCAPI *send_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`

7.25.1 Detailed Description

Definition at line 365 of file `PluginDefs.h`.

7.25.2 Member Function Documentation

7.25.2.1 tagDCHub::Bool (DCAPI * *send_private_message*)

7.25.2.2 tagDCHub::HubDataPtr (DCAPI * *add_hub*) const

7.25.2.3 tagDCHub::HubDataPtr (DCAPI * *find_hub*) const

7.25.2.4 tagDCHub::HubDataPtr (DCAPI * *copy*) const

7.25.2.5 tagDCHub::UserDataPtr (DCAPI * *find_user*) const

7.25.2.6 tagDCHub::UserDataPtr (DCAPI * *copy_user*) const

7.25.2.7 tagDCHub::void (DCAPI * *emulate_protocol_cmd*)

7.25.2.8 tagDCHub::void (DCAPI * *local_message*)

7.25.2.9 tagDCHub::void (DCAPI * *remove_hub*)

7.25.2.10 tagDCHub::void (DCAPI * *send_protocol_cmd*)

7.25.2.11 tagDCHub::void (DCAPI * *send_message*)

7.25.2.12 tagDCHub::void (DCAPI * *release_user*)

7.25.2.13 tagDCHub::void (DCAPI * *release*)

7.25.3 Member Data Documentation

7.25.3.1 uint32_t tagDCHub::apiVersion

Definition at line 367 of file PluginDefs.h.

7.25.3.2 const char * tagDCHub::cmd

Definition at line 373 of file PluginDefs.h.

7.25.3.3 const char* tagDCHub::hubUrl

Definition at line 380 of file PluginDefs.h.

7.25.3.4 const char * tagDCHub::msg

Definition at line 376 of file PluginDefs.h.

7.25.3.5 `const char* tagDCHub::nick`

Definition at line 369 of file PluginDefs.h.

7.25.3.6 `const char const char* tagDCHub::password`

Definition at line 369 of file PluginDefs.h.

7.25.3.7 `const char Bool tagDCHub::thirdPerson`

Definition at line 376 of file PluginDefs.h.

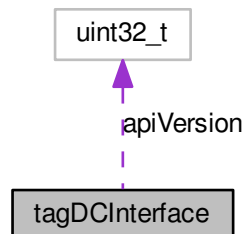
7.25.3.8 `const char MsgType tagDCHub::type`

Definition at line 377 of file PluginDefs.h.

7.26 tagDCInterface Struct Reference

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

Collaboration diagram for tagDCInterface:



Public Attributes

- `uint32_t apiVersion`

7.26.1 Detailed Description

Definition at line 280 of file PluginDefs.h.

7.26.2 Member Data Documentation

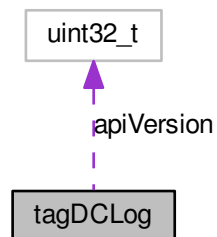
7.26.2.1 uint32_t tagDCInterface::apiVersion

Definition at line 282 of file PluginDefs.h.

7.27 tagDCLog Struct Reference

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

Collaboration diagram for tagDCLog:



Public Member Functions

- `void (DCAPI *log)(const char *msg)`

Public Attributes

- `uint32_t apiVersion`

7.27.1 Detailed Description

Definition at line 342 of file PluginDefs.h.

7.27.2 Member Function Documentation

7.27.2.1 tagDCLog::void (DCAPI * log) const

7.27.3 Member Data Documentation

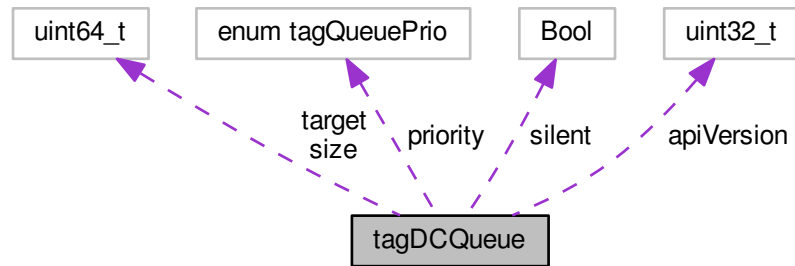
7.27.3.1 uint32_t tagDCLog::apiVersion

Definition at line 344 of file PluginDefs.h.

7.28 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 *copy)(const `QueueDataPtr` hItem)
- `QueueDataPtr` (DCAPI *find_download)(const char *target)
- `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

7.28.1 Detailed Description

Definition at line 389 of file PluginDefs.h.

7.28.2 Member Function Documentation

7.28.2.1 tagDCQueue::Bool (DCAPI * *pause*)

7.28.2.2 tagDCQueue::QueueDataPtr (DCAPI * *add_list*)

7.28.2.3 tagDCQueue::QueueDataPtr (DCAPI * *add_download*) const

7.28.2.4 tagDCQueue::QueueDataPtr (DCAPI * *copy*) const

7.28.2.5 tagDCQueue::QueueDataPtr (DCAPI * *find_download*) const

7.28.2.6 tagDCQueue::void (DCAPI * *remove_download*)

7.28.2.7 tagDCQueue::void (DCAPI * *set_priority*)

7.28.2.8 tagDCQueue::void (DCAPI * *release*)

7.28.3 Member Data Documentation

7.28.3.1 uint32_t tagDCQueue::apiVersion

Definition at line 391 of file PluginDefs.h.

7.28.3.2 QueuePrio tagDCQueue::priority

Definition at line 398 of file PluginDefs.h.

7.28.3.3 Bool tagDCQueue::silent

Definition at line 393 of file PluginDefs.h.

7.28.3.4 uint64_t tagDCQueue::size

Definition at line 394 of file PluginDefs.h.

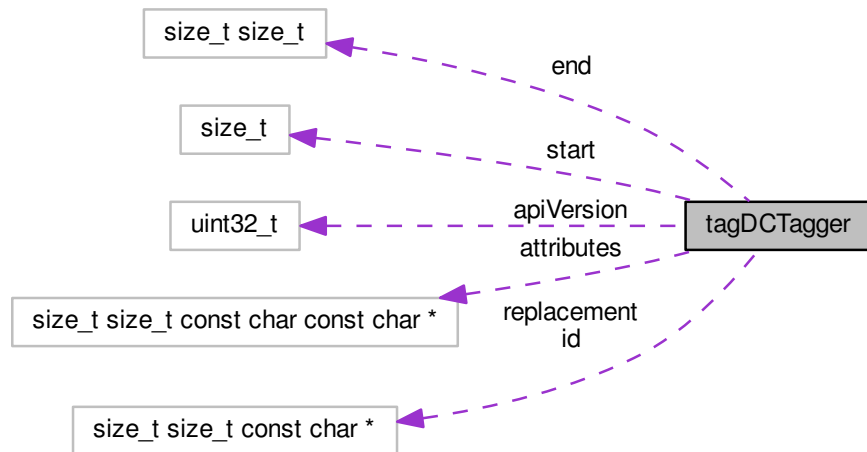
7.28.3.5 uint64_t const char* tagDCQueue::target

Definition at line 394 of file PluginDefs.h.

7.29 tagDCTagger Struct Reference

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

Collaboration diagram for tagDCTagger:



Public Member Functions

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

7.29.1 Detailed Description

Definition at line 423 of file `PluginDefs.h`.

7.29.2 Member Function Documentation

7.29.2.1 `const char* DCAPI tagDCTagger::get_text (TagDataPtr hTags)`

7.29.2.2 `tagDCTagger::void (DCAPI * replace_text)`

7.29.2.3 `tagDCTagger::void (DCAPI * add_tag)`

7.29.3 Member Data Documentation

7.29.3.1 `uint32_t tagDCTagger::apiVersion`

Definition at line 425 of file PluginDefs.h.

7.29.3.2 `size_t size_t const char const char* tagDCTagger::attributes`

Definition at line 427 of file PluginDefs.h.

7.29.3.3 `size_t size_t tagDCTagger::end`

Definition at line 427 of file PluginDefs.h.

7.29.3.4 `size_t size_t const char* tagDCTagger::id`

Definition at line 427 of file PluginDefs.h.

7.29.3.5 `size_t size_t const char* tagDCTagger::replacement`

Definition at line 431 of file PluginDefs.h.

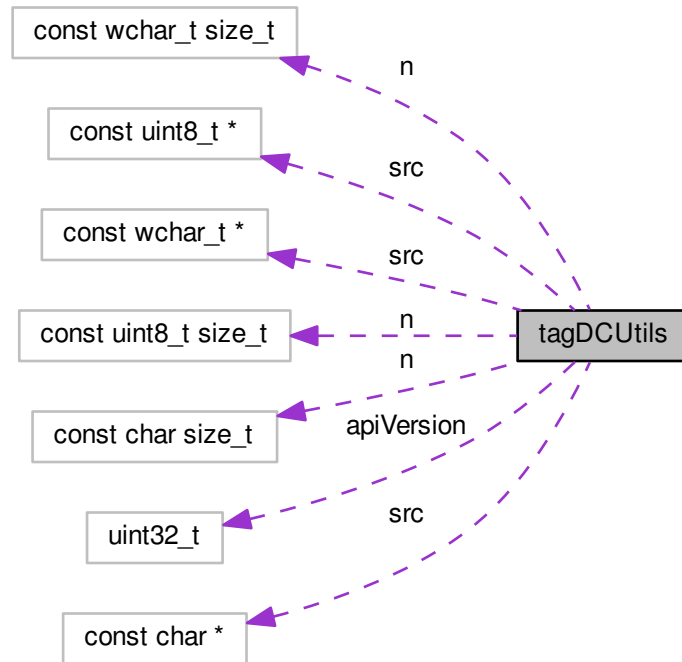
7.29.3.6 `size_t tagDCTagger::start`

Definition at line 427 of file PluginDefs.h.

7.30 tagDCUtils Struct Reference

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

Collaboration diagram for tagDCUtils:



Public Member Functions

- [size_t](#) (DCAPI *to_utf8)(char *dst
- [size_t](#) (DCAPI *from_base32)(uint8_t *dst
- [size_t](#) (DCAPI *to_base32)(char *dst
- [size_t](#) (DCAPI *wcs_to_utf8)(char *dst
- [size_t](#) (DCAPI *utf8_to_wcs)(wchar_t *dst
- [size_t](#) (DCAPI *from_utf8)(char *dst

Public Attributes

- `uint32_t` [apiVersion](#)
- `const uint8_t size_t` [n](#)
- `const wchar_t size_t` [n](#)
- `const char size_t` [n](#)
- `const wchar_t *` [src](#)

- `const char * src`
- `const uint8_t * src`

7.30.1 Detailed Description

Definition at line 408 of file PluginDefs.h.

7.30.2 Member Function Documentation

7.30.2.1 `tagDCUtils::size_t (DCAPI * to_utf8)`

7.30.2.2 `tagDCUtils::size_t (DCAPI * from_base32)`

7.30.2.3 `tagDCUtils::size_t (DCAPI * to_base32)`

7.30.2.4 `tagDCUtils::size_t (DCAPI * wcs_to_utf8)`

7.30.2.5 `tagDCUtils::size_t (DCAPI * utf8_to_wcs)`

7.30.2.6 `tagDCUtils::size_t (DCAPI * from_utf8)`

7.30.3 Member Data Documentation

7.30.3.1 `uint32_t tagDCUtils::apiVersion`

Definition at line 410 of file PluginDefs.h.

7.30.3.2 `const uint8_t size_t tagDCUtils::n`

Definition at line 418 of file PluginDefs.h.

7.30.3.3 `const wchar_t size_t tagDCUtils::n`

Definition at line 416 of file PluginDefs.h.

7.30.3.4 `const char size_t tagDCUtils::n`

Definition at line 412 of file PluginDefs.h.

7.30.3.5 `const wchar_t* tagDCUtils::src`

Definition at line 416 of file PluginDefs.h.

7.30.3.6 `const char * tagDCUtils::src`

Definition at line 412 of file PluginDefs.h.

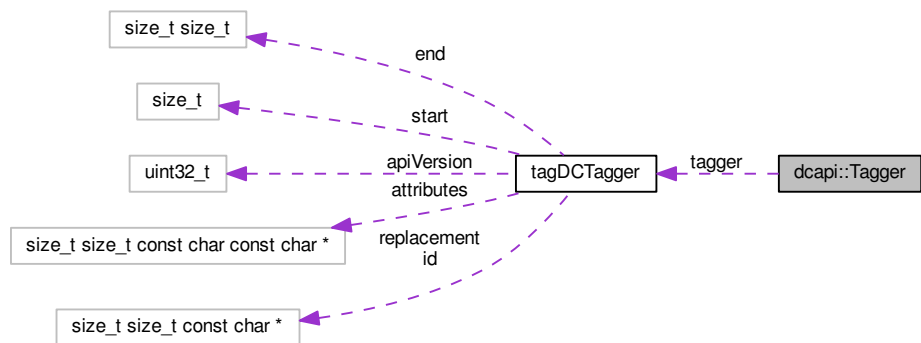
7.30.3.7 `const uint8_t* tagDCUtils::src`

Definition at line 418 of file PluginDefs.h.

7.31 `dcapi::Tagger` Class Reference

```
#include <Tagger.h>
```

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`

7.31.1 Detailed Description

Definition at line 28 of file Tagger.h.

7.31.2 Member Function Documentation

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

Definition at line 35 of file Tagger.cpp.

References tagger.

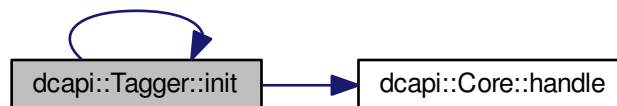
7.31.2.2 bool dcapi::Tagger::init () [static]

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:



7.31.2.3 void dcapi::Tagger::init (DCTaggerPtr coreTagger) [static]

Definition at line 34 of file Tagger.cpp.

References tagger.

7.31.3 Member Data Documentation

7.31.3.1 DCTaggerPtr dcapi::Tagger::tagger [static, private]

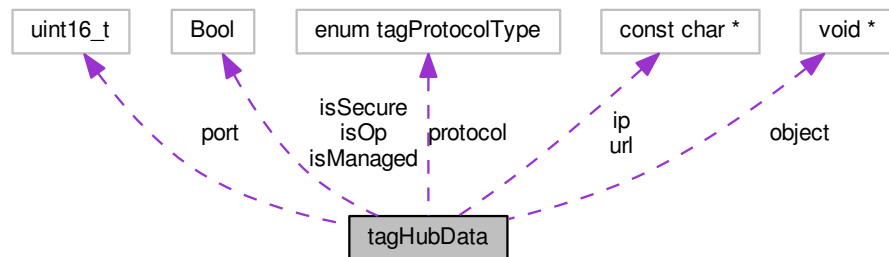
Definition at line 36 of file Tagger.h.

Referenced by handle(), and init().

7.32 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`

7.32.1 Detailed Description

Definition at line 227 of file PluginDefs.h.

7.32.2 Member Data Documentation

7.32.2.1 const char* tagHubData::ip

Definition at line 229 of file PluginDefs.h.

7.32.2.2 Bool tagHubData::isManaged

Definition at line 235 of file PluginDefs.h.

7.32.2.3 Bool tagHubData::isOp

Definition at line 233 of file PluginDefs.h.

7.32.2.4 Bool tagHubData::isSecure

Definition at line 234 of file PluginDefs.h.

7.32.2.5 dcptr_t tagHubData::object

Definition at line 230 of file PluginDefs.h.

7.32.2.6 uint16_t tagHubData::port

Definition at line 231 of file PluginDefs.h.

7.32.2.7 ProtocolType tagHubData::protocol

Definition at line 232 of file PluginDefs.h.

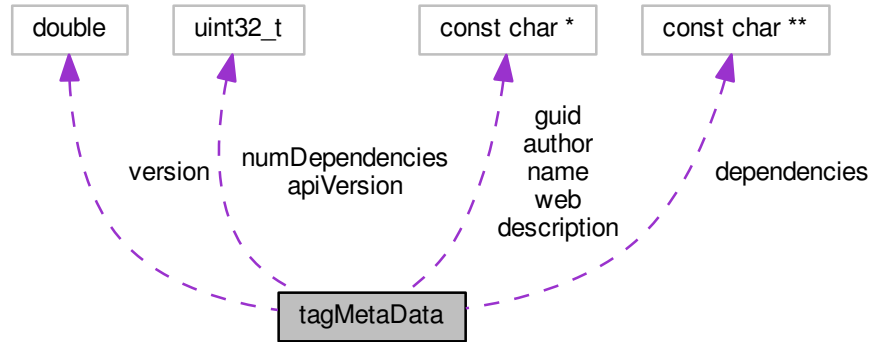
7.32.2.8 const char* tagHubData::url

Definition at line 228 of file PluginDefs.h.

7.33 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](#)

7.33.1 Detailed Description

Definition at line 267 of file `PluginDefs.h`.

7.33.2 Member Data Documentation

7.33.2.1 `uint32_t tagMetaData::apiVersion`

Definition at line 275 of file `PluginDefs.h`.

Referenced by `pluginInit()`.

7.33.2.2 const char* tagMetaData::author

Definition at line 269 of file PluginDefs.h.

Referenced by pluginInit().

7.33.2.3 const char tagMetaData::dependencies**

Definition at line 273 of file PluginDefs.h.

7.33.2.4 const char* tagMetaData::description

Definition at line 270 of file PluginDefs.h.

Referenced by pluginInit().

7.33.2.5 const char* tagMetaData::guid

Definition at line 272 of file PluginDefs.h.

Referenced by pluginInit().

7.33.2.6 const char* tagMetaData::name

Definition at line 268 of file PluginDefs.h.

Referenced by pluginInit().

7.33.2.7 uint32_t tagMetaData::numDependencies

Definition at line 274 of file PluginDefs.h.

7.33.2.8 double tagMetaData::version

Definition at line 276 of file PluginDefs.h.

Referenced by pluginInit().

7.33.2.9 const char* tagMetaData::web

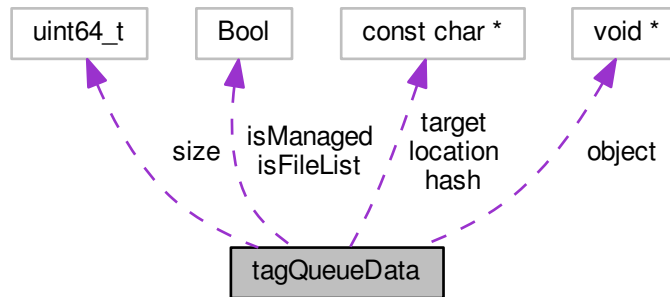
Definition at line 271 of file PluginDefs.h.

Referenced by pluginInit().

7.34 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](#)

7.34.1 Detailed Description

Definition at line 250 of file PluginDefs.h.

7.34.2 Member Data Documentation

7.34.2.1 const char* tagQueueData::hash

Definition at line 253 of file PluginDefs.h.

7.34.2.2 Bool tagQueueData::isFileList

Definition at line 256 of file PluginDefs.h.

7.34.2.3 Bool tagQueueData::isManaged

Definition at line 257 of file PluginDefs.h.

7.34.2.4 const char* tagQueueData::location

Definition at line 252 of file PluginDefs.h.

7.34.2.5 dcptr_t tagQueueData::object

Definition at line 254 of file PluginDefs.h.

7.34.2.6 uint64_t tagQueueData::size

Definition at line 255 of file PluginDefs.h.

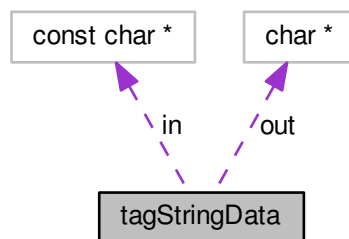
7.34.2.7 const char* tagQueueData::target

Definition at line 251 of file PluginDefs.h.

7.35 tagStringData Struct Reference

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

Collaboration diagram for tagStringData:



Public Attributes

- const char * [in](#)

- char * [out](#)

7.35.1 Detailed Description

Definition at line 203 of file PluginDefs.h.

7.35.2 Member Data Documentation

7.35.2.1 const char* [tagStringData::in](#)

Definition at line 204 of file PluginDefs.h.

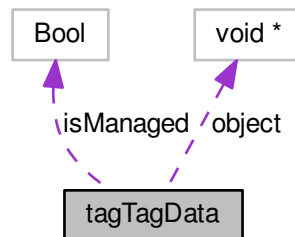
7.35.2.2 char* [tagStringData::out](#)

Definition at line 205 of file PluginDefs.h.

7.36 tagTagData Struct Reference

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

Collaboration diagram for tagTagData:



Public Attributes

- Bool [isManaged](#)
- [dcptr_t](#) [object](#)

7.36.1 Detailed Description

Definition at line 261 of file PluginDefs.h.

7.36.2 Member Data Documentation

7.36.2.1 Bool tagTagData::isManaged

Definition at line 263 of file PluginDefs.h.

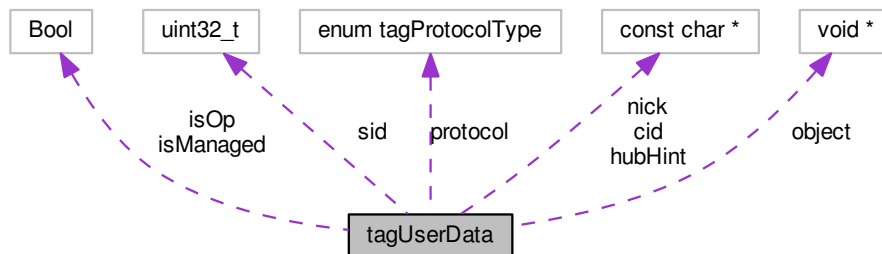
7.36.2.2 dcptr_t tagTagData::object

Definition at line 262 of file PluginDefs.h.

7.37 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`

7.37.1 Detailed Description

Definition at line 215 of file PluginDefs.h.

7.37.2 Member Data Documentation

7.37.2.1 `const char*` `tagUserData::cid`

Definition at line 218 of file PluginDefs.h.

7.37.2.2 `const char*` `tagUserData::hubHint`

Definition at line 217 of file PluginDefs.h.

7.37.2.3 `Bool` `tagUserData::isManaged`

Definition at line 223 of file PluginDefs.h.

7.37.2.4 `Bool` `tagUserData::isOp`

Definition at line 222 of file PluginDefs.h.

7.37.2.5 `const char*` `tagUserData::nick`

Definition at line 216 of file PluginDefs.h.

7.37.2.6 `dcptr_t` `tagUserData::object`

Definition at line 219 of file PluginDefs.h.

7.37.2.7 `ProtocolType` `tagUserData::protocol`

Definition at line 221 of file PluginDefs.h.

7.37.2.8 `uint32_t` `tagUserData::sid`

Definition at line 220 of file PluginDefs.h.

7.38 `dcapi::Hooks::Timer` Struct Reference

```
#include <Hooks.h>
```

Static Public Member Functions

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

7.38.1 Detailed Description

Definition at line 61 of file Hooks.h.

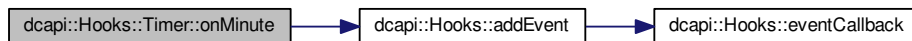
7.38.2 Member Function Documentation

7.38.2.1 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:

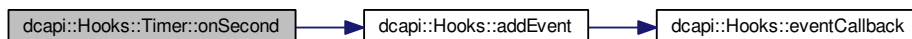


7.38.2.2 void `dcapi::Hooks::Timer::onSecond` (function< bool(uint64_t, bool &)> *f*)
[static]

Definition at line 55 of file Hooks.cpp.

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

Here is the call graph for this function:



7.39 dcapi::Hooks::UI Struct Reference

```
#include <Hooks.h>
```

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)

7.39.1 Detailed Description

Definition at line 90 of file Hooks.h.

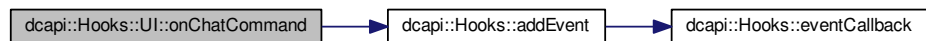
7.39.2 Member Function Documentation

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

Definition at line 128 of file Hooks.cpp.

References [dcapi::Hooks::addEvent\(\)](#), and [HOOK_UI_CHAT_COMMAND](#).

Here is the call graph for this function:

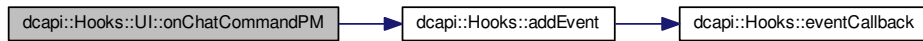


7.39.2.2 void [dcapi::Hooks::UI::onChatCommandPM](#) (function< bool([UserDataPtr](#), [CommandDataPtr](#), bool &)> f) [static]

Definition at line 132 of file Hooks.cpp.

References [dcapi::Hooks::addEvent\(\)](#), and [HOOK_UI_CHAT_COMMAND_PM](#).

Here is the call graph for this function:

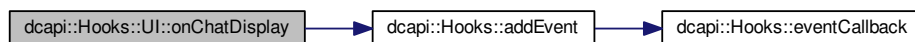


7.39.2.3 void dcapi::Hooks::UI::onChatDisplay (function< bool(UserDataPtr, StringDataPtr, bool &)> f) [static]

Definition at line 124 of file Hooks.cpp.

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

Here is the call graph for this function:

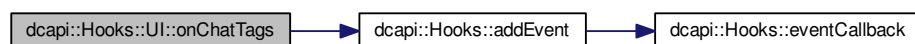


7.39.2.4 void dcapi::Hooks::UI::onChatTags (function< bool(UserDataPtr, TagDataPtr, bool &)> f) [static]

Definition at line 120 of file Hooks.cpp.

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

Here is the call graph for this function:

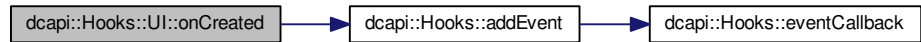


7.39.2.5 void dcapi::Hooks::UI::onCreated (function< bool(dcptr_t, bool &)> f) [static]

Definition at line 116 of file Hooks.cpp.

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

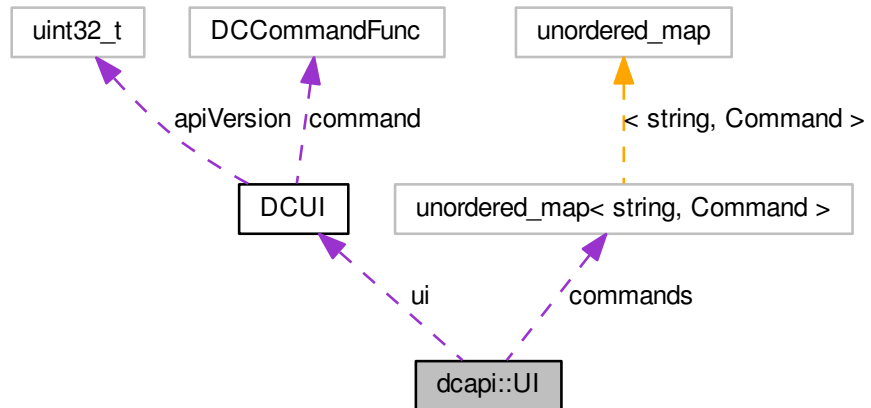
Here is the call graph for this function:



7.40 dcapi::UI Class Reference

```
#include <UI.h>
```

Collaboration diagram for `dcapi::UI`:



Public Types

- typedef `function< void()>` [Command](#)

Static Public Member Functions

- static void [addCommand](#) (string name, [Command](#) command)
- static [DCUIPtr](#) [handle](#) ()

- static void [init](#) ([DCUIPtr](#) coreUI)
- static bool [init](#) ()
- static void [removeCommand](#) (const string &name)

Static Private Member Functions

- static void DCAPI [commandCallback](#) (const char *name)

Static Private Attributes

- static unordered_map< string, [Command](#) > [commands](#)
- static [DCUIPtr](#) [ui](#)

7.40.1 Detailed Description

Definition at line 36 of file UI.h.

7.40.2 Member Typedef Documentation

7.40.2.1 `typedef function<void ()> dcapi::UI::Command`

Definition at line 43 of file UI.h.

7.40.3 Member Function Documentation

7.40.3.1 `void dcapi::UI::addCommand (string name, Command command) [static]`

Definition at line 38 of file UI.cpp.

References [commandCallback\(\)](#), [commands](#), and [ui](#).

Here is the call graph for this function:



7.40.3.2 `void DCAPIDcapi::UI::commandCallback (const char * name) [static, private]`

Definition at line 47 of file UI.cpp.

References `commands`.

Referenced by `addCommand()`.

Here is the caller graph for this function:



7.40.3.3 `DCUIPtr Dcapi::UI::handle () [static]`

Definition at line 36 of file UI.cpp.

References `ui`.

7.40.3.4 `void Dcapi::UI::init (DCUIPtr coreUI) [static]`

Definition at line 35 of file UI.cpp.

References `ui`.

7.40.3.5 `bool Dcapi::UI::init () [static]`

Definition at line 30 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:



7.40.3.6 `void dcapi::UI::removeCommand (const string & name) [static]`

Definition at line 42 of file UI.cpp.

References `commands`, and `ui`.

7.40.4 Member Data Documentation

7.40.4.1 `unordered_map< string, UI::Command > dcapi::UI::commands [static, private]`

Definition at line 52 of file UI.h.

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

7.40.4.2 `DCUIPtr dcapi::UI::ui [static, private]`

Definition at line 50 of file UI.h.

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

7.41 dcapi::Hooks::Users Struct Reference

```
#include <Hooks.h>
```

Static Public Member Functions

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

7.41.1 Detailed Description

Definition at line 71 of file Hooks.h.

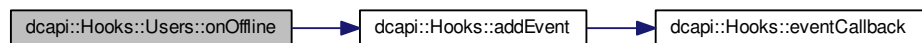
7.41.2 Member Function Documentation

7.41.2.1 `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:

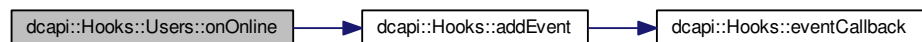


7.41.2.2 `void dcapi::Hooks::Users::onOnline (function< bool(UserDataPtr, bool &)> f)`
[static]

Definition at line 73 of file Hooks.cpp.

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

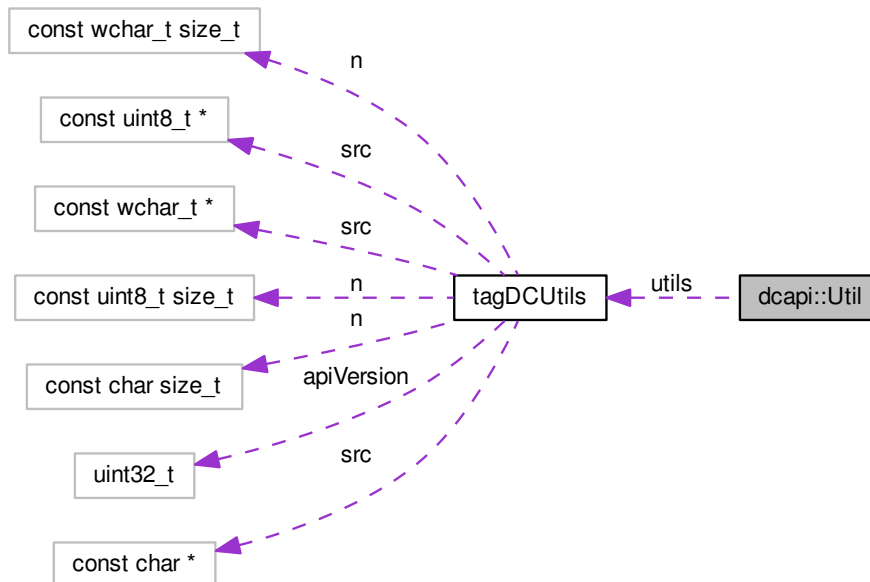
Here is the call graph for this function:



7.42 dcapi::Util Class Reference

```
#include <Util.h>
```

Collaboration diagram for dcapi::Util:



Static Public Member Functions

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

Static Private Attributes

- static [DCUtilsPtr utils](#)

7.42.1 Detailed Description

Definition at line 35 of file Util.h.

7.42.2 Member Function Documentation

7.42.2.1 `string dcapi::Util::fromUtf8 (const string & str) [static]`

Definition at line 60 of file Util.cpp.

References `utils`.

7.42.2.2 `DCUtilsPtr dcapi::Util::handle () [static]`

Definition at line 35 of file Util.cpp.

References `utils`.

7.42.2.3 `void dcapi::Util::init (DCUtilsPtr coreUtils) [static]`

Definition at line 34 of file Util.cpp.

References `utils`.

7.42.2.4 `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:



7.42.2.5 `string dcapi::Util::toUtf8 (const string & str) [static]`

Definition at line 69 of file Util.cpp.

References `utils`.

7.42.3 Member Data Documentation

7.42.3.1 `DCUtilsPtr dcapi::Util::utils [static, private]`

Definition at line 55 of file Util.h.

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

Chapter 8

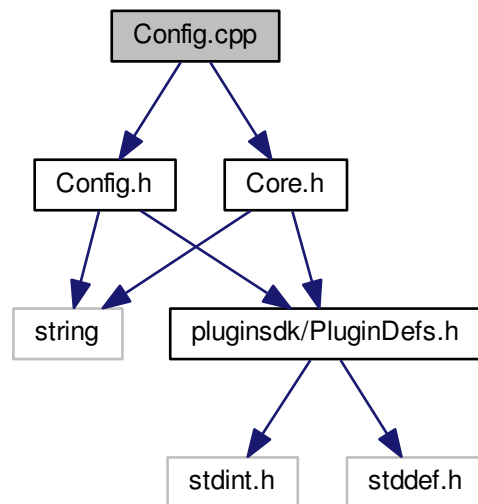
File Documentation

8.1 Config.cpp File Reference

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

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

Include dependency graph for Config.cpp:



Namespaces

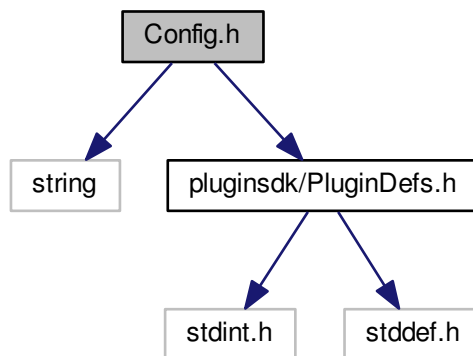
- namespace [dcapi](#)

8.2 Config.h File Reference

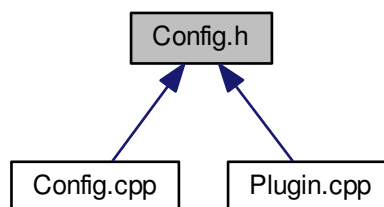
```
#include <string>
```

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

Include dependency graph for Config.h:



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



Classes

- class [dcapi::Config](#)

Namespaces

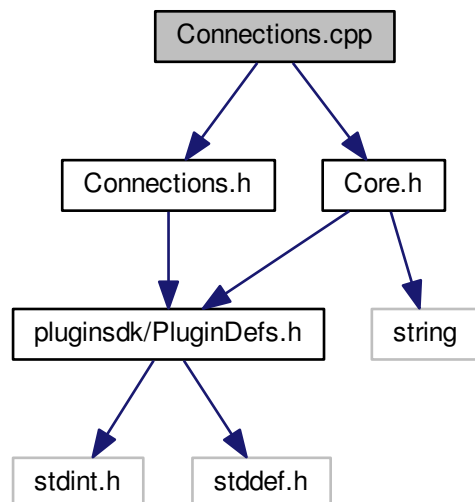
- namespace [dcapi](#)

8.3 Connections.cpp File Reference

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

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

Include dependency graph for Connections.cpp:



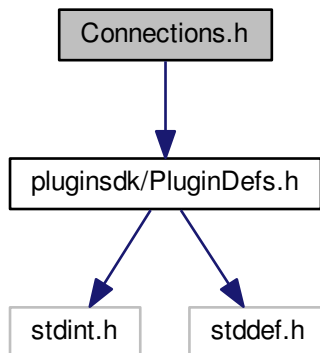
Namespaces

- namespace [dcapi](#)

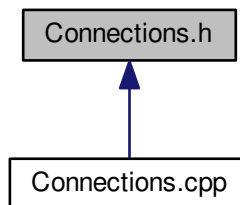
8.4 Connections.h File Reference

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

Include dependency graph for Connections.h:



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



Classes

- class [dcapi::Connections](#)

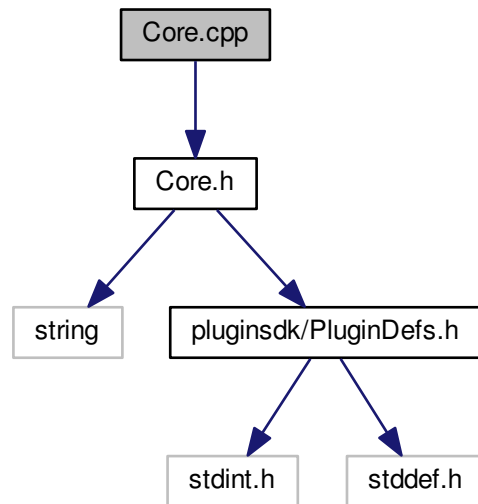
Namespaces

- namespace [dcapi](#)

8.5 Core.cpp File Reference

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

Include dependency graph for Core.cpp:



Namespaces

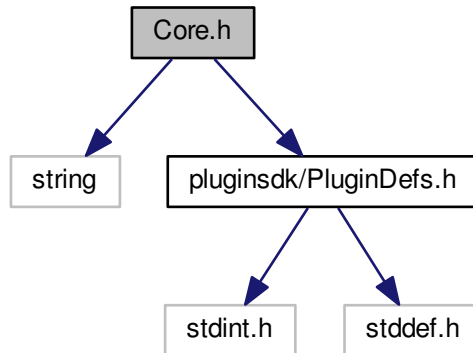
- namespace [dcapi](#)

8.6 Core.h File Reference

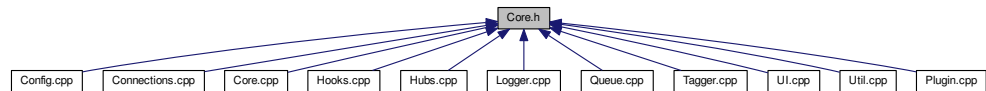
```
#include <string>
```

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

Include dependency graph for Core.h:



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



Classes

- class [dcapi::Core](#)

Namespaces

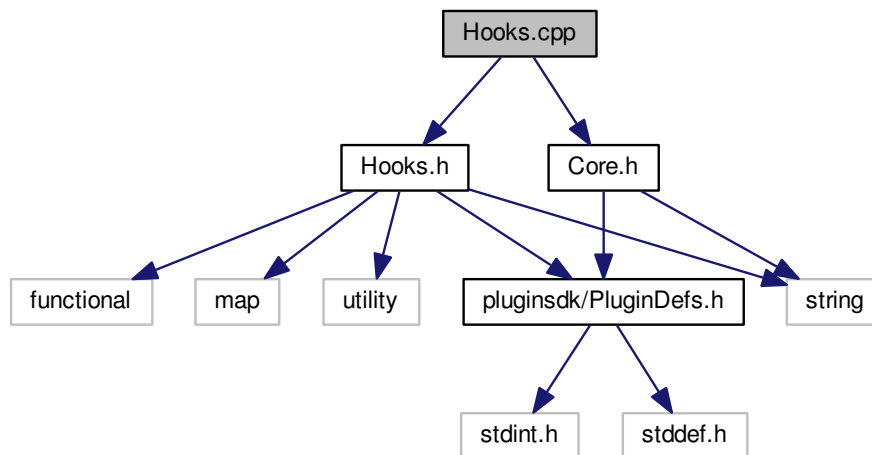
- namespace [dcapi](#)

8.7 Hooks.cpp File Reference

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

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


Include dependency graph for Hooks.cpp:



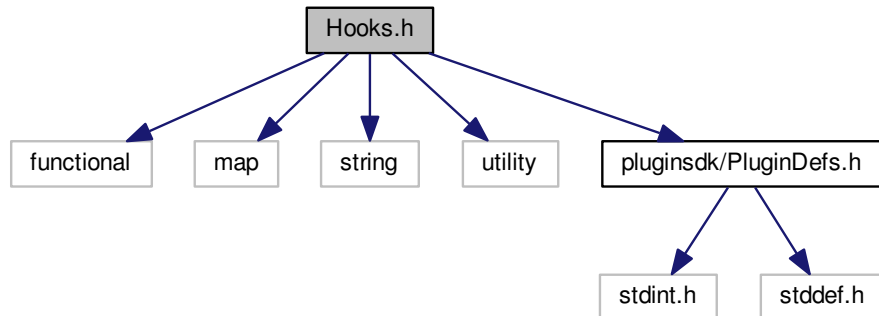
Namespaces

- namespace [dcapi](#)

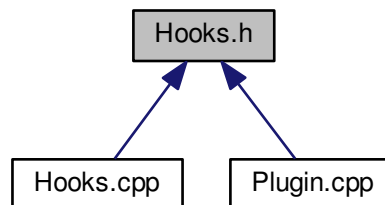
8.8 Hooks.h File Reference

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

Include dependency graph for Hooks.h:



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



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

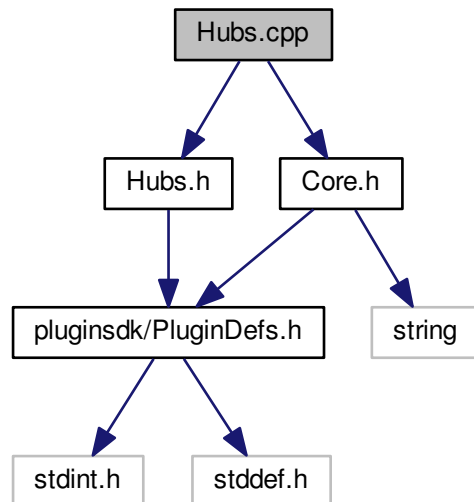
- namespace [dcapi](#)

8.9 Hubs.cpp File Reference

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

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

Include dependency graph for Hubs.cpp:



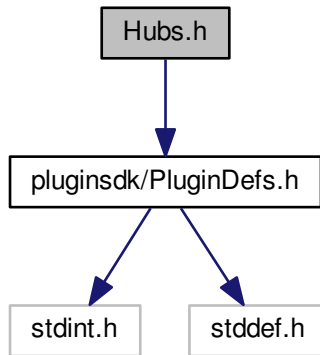
Namespaces

- namespace [dcapi](#)

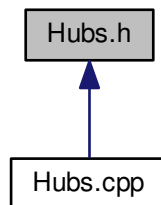
8.10 Hubs.h File Reference

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

Include dependency graph for Hubs.h:



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



Classes

- class [dcapi::Hubs](#)

Namespaces

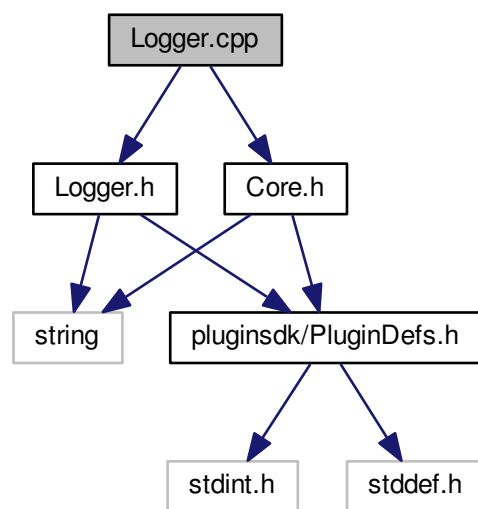
- namespace [dcapi](#)

8.11 Logger.cpp File Reference

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

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

Include dependency graph for Logger.cpp:



Namespaces

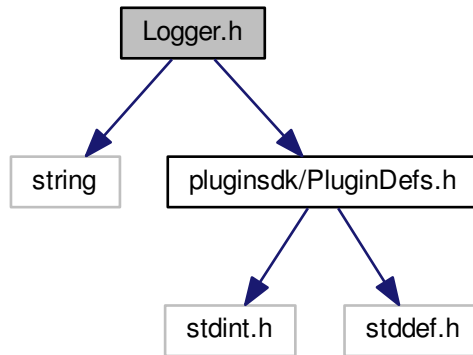
- namespace [dcapi](#)

8.12 Logger.h File Reference

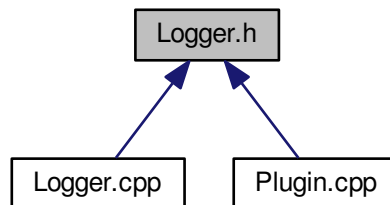
```
#include <string>
```

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

Include dependency graph for `Logger.h`:



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



Classes

- class `dcapi::Logger`

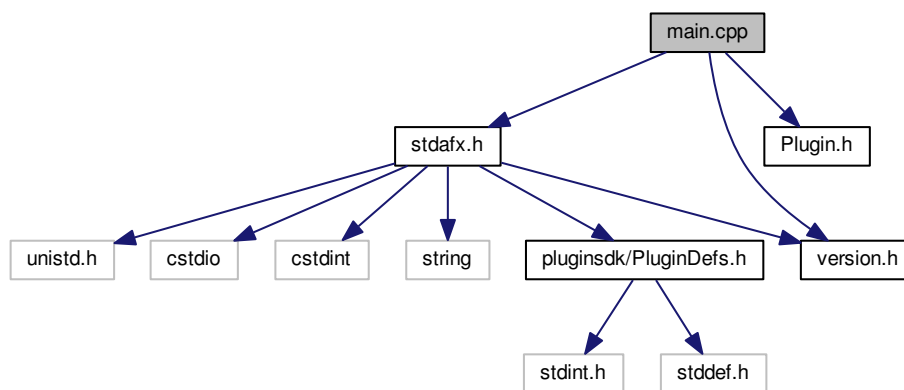
Namespaces

- namespace `dcapi`

8.13 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)

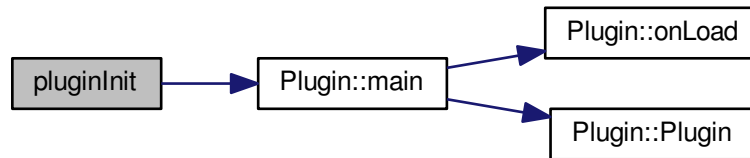
8.13.1 Function Documentation

8.13.1.1 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:

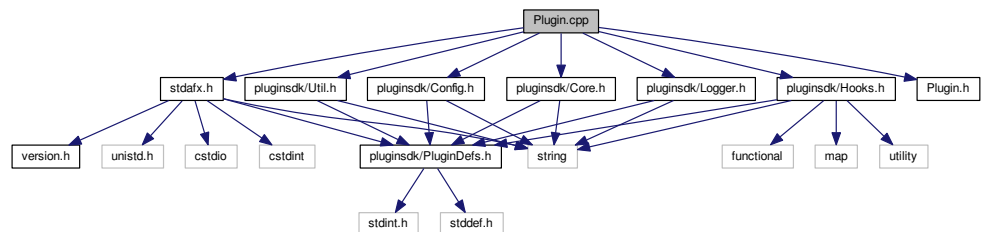


8.14 Plugin.cpp File Reference

```

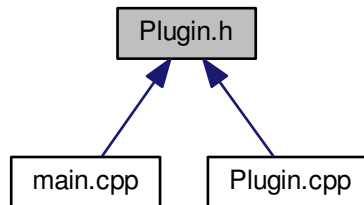
#include "stdafx.h"
#include "Plugin.h"
#include <pluginsdk/Config.h>
#include <pluginsdk/Core.h>
#include <pluginsdk/Hooks.h>
#include <pluginsdk/Logger.h>
#include <pluginsdk/Util.h>
  
```

Include dependency graph for Plugin.cpp:



8.15 Plugin.h File Reference

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



Classes

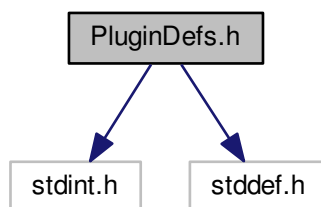
- class [Plugin](#)

8.16 PluginDefs.h File Reference

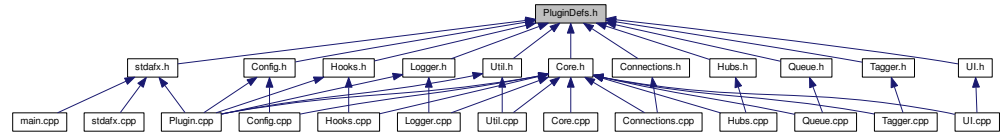
```
#include <stdint.h>
```

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

Include dependency graph for `PluginDefs.h`:



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



Classes

- struct [DCUI](#)
- struct [tagCommandData](#)
- struct [tagConfigBool](#)
- struct [tagConfigInt](#)
- struct [tagConfigInt64](#)
- struct [tagConfigStr](#)
- struct [tagConfigValue](#)
- struct [tagConnectionData](#)
- struct [tagDCCConfig](#)
- struct [tagDCCConnection](#)
- 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 [tagUserData](#)

Defines

- `#define Bool dcBool`
- `#define DCAPI`
- `#define DCAPI_CORE_VER 6`
- `#define DCEXP __attribute__((visibility("default")))`
- `#define DCIMP __attribute__((visibility("default")))`
- `#define DCINTF_CONFIG "generic.plugins.DCCconfig"`

- #define `DCINTF_CONFIG_VER` 1
- #define `DCINTF_DCPP_CONNECTIONS` "dcpp.network.DCConnection"
- #define `DCINTF_DCPP_CONNECTIONS_VER` 2
- #define `DCINTF_DCPP_HUBS` "dcpp.network.DCHub"
- #define `DCINTF_DCPP_HUBS_VER` 1
- #define `DCINTF_DCPP_QUEUE` "dcpp.queue.DCQueue"
- #define `DCINTF_DCPP_QUEUE_VER` 2
- #define `DCINTF_DCPP_TAGGER` "dcpp.xml.DCTagger"
- #define `DCINTF_DCPP_TAGGER_VER` 2
- #define `DCINTF_DCPP_UI` "dcpp.ui.DCUI"
- #define `DCINTF_DCPP_UI_VER` 1
- #define `DCINTF_DCPP_UTILS` "dcpp.utils.DCUtils"
- #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"
- #define `DCINTF_LOGGING_VER` 1
- #define `False` dcFalse
- #define `HOOK_CHAT_IN` "dcpp.chat.onIncomingChat"
- #define `HOOK_CHAT_OUT` "dcpp.chat.onOutgoingChat"
- #define `HOOK_CHAT_PM_IN` "dcpp.chat.onIncomingPM"
- #define `HOOK_CHAT_PM_OUT` "dcpp.chat.onOutgoingPM"
- #define `HOOK_HUB_OFFLINE` "dcpp.hubs.onOffline"
- #define `HOOK_HUB_ONLINE` "dcpp.hubs.onOnline"
- #define `HOOK_NETWORK_CONN_IN` "dcpp.network.onClientDataIn"
- #define `HOOK_NETWORK_CONN_OUT` "dcpp.network.onClientDataOut"
- #define `HOOK_NETWORK_HUB_IN` "dcpp.network.onHubDataIn"
- #define `HOOK_NETWORK_HUB_OUT` "dcpp.network.onHubDataOut"
- #define `HOOK_QUEUE_ADDED` "dcpp.queue.onAdded"
- #define `HOOK_QUEUE_FINISHED` "dcpp.queue.onFinished"
- #define `HOOK_QUEUE_MOVED` "dcpp.queue.onMoved"
- #define `HOOK_QUEUE_REMOVED` "dcpp.queue.onRemoved"
- #define `HOOK_TIMER_MINUTE` "dcpp.timer.onMinute"
- #define `HOOK_TIMER_SECOND` "dcpp.timer.onSecond"
- #define `HOOK_UI_CHAT_COMMAND` "dcpp.ui.onChatCommand"
- #define `HOOK_UI_CHAT_COMMAND_PM` "dcpp.ui.onChatCommandPM"
- #define `HOOK_UI_CHAT_DISPLAY` "dcpp.ui.onChatDisplay"
- #define `HOOK_UI_CHAT_TAGS` "dcpp.ui.onChatTags"
- #define `HOOK_UI_CREATED` "dcpp.ui.onCreated"
- #define `HOOK_USER_OFFLINE` "dcpp.users.onOffline"
- #define `HOOK_USER_ONLINE` "dcpp.users.onOnline"
- #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 [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_UNINSTALL](#), [ON_LOAD](#), [ON_UNLOAD](#), [ON_CONFIGURE](#) }
- 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)

8.16.1 Define Documentation

8.16.1.1 #define Bool dcBool

Definition at line 169 of file [PluginDefs.h](#).

8.16.1.2 #define DCAPI

Definition at line 49 of file PluginDefs.h.

8.16.1.3 #define DCAPI_CORE_VER 6

Definition at line 35 of file PluginDefs.h.

Referenced by pluginInit().

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

Definition at line 51 of file PluginDefs.h.

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

Definition at line 52 of file PluginDefs.h.

8.16.1.6 #define DCINTF_CONFIG "generic.plugins.DCConfig"

Definition at line 60 of file PluginDefs.h.

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

8.16.1.7 #define DCINTF_CONFIG_VER 1

Definition at line 61 of file PluginDefs.h.

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

8.16.1.8 #define DCINTF_DCPP_CONNECTIONS "dcpp.network.DCConnection"

Definition at line 67 of file PluginDefs.h.

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

8.16.1.9 #define DCINTF_DCPP_CONNECTIONS_VER 2

Definition at line 68 of file PluginDefs.h.

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

8.16.1.10 #define DCINTF_DCPP_HUBS "dcpp.network.DCHub"

Definition at line 70 of file PluginDefs.h.

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

8.16.1.11 #define DCINTF_DCPP_HUBS_VER 1

Definition at line 71 of file PluginDefs.h.

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

8.16.1.12 #define DCINTF_DCPP_QUEUE "dcpp.queue.DCQueue"

Definition at line 73 of file PluginDefs.h.

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

8.16.1.13 #define DCINTF_DCPP_QUEUE_VER 2

Definition at line 74 of file PluginDefs.h.

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

8.16.1.14 #define DCINTF_DCPP_TAGGER "dcpp.xml.DCTagger"

Definition at line 79 of file PluginDefs.h.

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

8.16.1.15 #define DCINTF_DCPP_TAGGER_VER 2

Definition at line 80 of file PluginDefs.h.

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

8.16.1.16 #define DCINTF_DCPP_UI "dcpp.ui.DCUI"

Definition at line 82 of file PluginDefs.h.

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

8.16.1.17 #define DCINTF_DCPP_UI_VER 1

Definition at line 83 of file PluginDefs.h.

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

8.16.1.18 #define DCINTF_DCPP_UTILS "dcpp.utils.DCUtils"

Definition at line 76 of file PluginDefs.h.

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

8.16.1.19 #define DCINTF_DCPP_UTILS_VER 1

Definition at line 77 of file PluginDefs.h.

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

8.16.1.20 #define DCINTF_HOOKS "generic.plugins.DCHooks"

Definition at line 56 of file PluginDefs.h.

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

8.16.1.21 #define DCINTF_HOOKS_VER DCAPI_CORE_VER

Definition at line 57 of file PluginDefs.h.

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

8.16.1.22 #define DCINTF_LOGGING "generic.plugins.DCLog"

Definition at line 63 of file PluginDefs.h.

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

8.16.1.23 #define DCINTF_LOGGING_VER 1

Definition at line 64 of file PluginDefs.h.

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

8.16.1.24 #define False dcFalse

Definition at line 171 of file PluginDefs.h.

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

8.16.1.25 #define HOOK_CHAT_IN "dcpp.chat.onIncomingChat"

Definition at line 86 of file PluginDefs.h.

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

8.16.1.26 #define HOOK_CHAT_OUT "dcpp.chat.onOutgoingChat"

Definition at line 87 of file PluginDefs.h.

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

8.16.1.27 #define HOOK_CHAT_PM_IN "dcpp.chat.onIncomingPM"

Definition at line 88 of file PluginDefs.h.

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

8.16.1.28 #define HOOK_CHAT_PM_OUT "dcpp.chat.onOutgoingPM"

Definition at line 89 of file PluginDefs.h.

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

8.16.1.29 #define HOOK_HUB_OFFLINE "dcpp.hubs.onOffline"

Definition at line 95 of file PluginDefs.h.

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

8.16.1.30 #define HOOK_HUB_ONLINE "dcpp.hubs.onOnline"

Definition at line 94 of file PluginDefs.h.

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

8.16.1.31 #define HOOK_NETWORK_CONN_IN "dcpp.network.onClientDataIn"

Definition at line 101 of file PluginDefs.h.

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

8.16.1.32 #define HOOK_NETWORK_CONN_OUT "dcpp.network.onClientDataOut"

Definition at line 102 of file PluginDefs.h.

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

8.16.1.33 #define HOOK_NETWORK_HUB_IN "dcpp.network.onHubDataIn"

Definition at line 99 of file PluginDefs.h.

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

8.16.1.34 #define HOOK_NETWORK_HUB_OUT "dcpp.network.onHubDataOut"

Definition at line 100 of file PluginDefs.h.

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

8.16.1.35 #define HOOK_QUEUE_ADDED "dcpp.queue.onAdded"

Definition at line 104 of file PluginDefs.h.

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

8.16.1.36 #define HOOK_QUEUE_FINISHED "dcpp.queue.onFinished"

Definition at line 107 of file PluginDefs.h.

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

8.16.1.37 #define HOOK_QUEUE_MOVED "dcpp.queue.onMoved"

Definition at line 105 of file PluginDefs.h.

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

8.16.1.38 #define HOOK_QUEUE_REMOVED "dcpp.queue.onRemoved"

Definition at line 106 of file PluginDefs.h.

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

8.16.1.39 #define HOOK_TIMER_MINUTE "dcpp.timer.onMinute"

Definition at line 92 of file PluginDefs.h.

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

8.16.1.40 #define HOOK_TIMER_SECOND "dcpp.timer.onSecond"

Definition at line 91 of file PluginDefs.h.

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

8.16.1.41 #define HOOK_UI_CHAT_COMMAND "dcpp.ui.onChatCommand"

Definition at line 112 of file PluginDefs.h.

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

8.16.1.42 #define HOOK_UI_CHAT_COMMAND_PM "dcpp.ui.onChatCommandPM"

Definition at line 113 of file PluginDefs.h.

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

8.16.1.43 #define HOOK_UI_CHAT_DISPLAY "dcpp.ui.onChatDisplay"

Definition at line 111 of file PluginDefs.h.

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

8.16.1.44 #define HOOK_UI_CHAT_TAGS "dcpp.ui.onChatTags"

Definition at line 110 of file PluginDefs.h.

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

8.16.1.45 #define HOOK_UI_CREATED "dcpp.ui.onCreated"

Definition at line 109 of file PluginDefs.h.

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

8.16.1.46 #define HOOK_USER_OFFLINE "dcpp.users.onOffline"

Definition at line 97 of file PluginDefs.h.

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

8.16.1.47 #define HOOK_USER_ONLINE "dcpp.users.onOnline"

Definition at line 96 of file PluginDefs.h.

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

8.16.1.48 #define True dcTrue

Definition at line 170 of file PluginDefs.h.

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

8.16.2 Typedef Documentation**8.16.2.1 typedef dcptr_t dcptr_t Bool* bBreak**

Definition at line 308 of file PluginDefs.h.

8.16.2.2 typedef struct tagCommandData CommandData**8.16.2.3 typedef struct tagCommandData * CommandDataPtr**

- 8.16.2.4 `typedef struct tagConfigBool ConfigBool`
 - 8.16.2.5 `typedef struct tagConfigBool * ConfigBoolPtr`
 - 8.16.2.6 `typedef struct tagConfigInt ConfigInt`
 - 8.16.2.7 `typedef struct tagConfigInt64 ConfigInt64`
 - 8.16.2.8 `typedef struct tagConfigInt64 * ConfigInt64Ptr`
 - 8.16.2.9 `typedef struct tagConfigInt * ConfigIntPtr`
 - 8.16.2.10 `typedef struct tagConfigStr ConfigStr`
 - 8.16.2.11 `typedef struct tagConfigStr * ConfigStrPtr`
 - 8.16.2.12 `typedef enum tagConfigType ConfigType`
 - 8.16.2.13 `typedef struct tagConfigValue ConfigValue`
 - 8.16.2.14 `typedef struct tagConfigValue * ConfigValuePtr`
 - 8.16.2.15 `typedef struct tagConnectionData ConnectionData`
 - 8.16.2.16 `typedef struct tagConnectionData * ConnectionDataPtr`
 - 8.16.2.17 `typedef DCCorePtr core`
- Definition at line 303 of file PluginDefs.h.
- 8.16.2.18 `typedef enum tagDCBool dcBool`
 - 8.16.2.19 `typedef struct tagDCConfig DCConfig`
 - 8.16.2.20 `typedef struct tagDCConfig * DCConfigPtr`
 - 8.16.2.21 `typedef struct tagDCConnection DCConnection`
 - 8.16.2.22 `typedef struct tagDCConnection * DCConnectionPtr`
 - 8.16.2.23 `typedef struct tagDCCore DCCore`
 - 8.16.2.24 `typedef struct tagDCCore * DCCorePtr`
 - 8.16.2.25 `typedef struct tagDCHooks DCHooks`
 - 8.16.2.26 `typedef struct tagDCHooks * DCHooksPtr`

8.16.2.27 `typedef struct tagDCHub DCHub`

8.16.2.28 `typedef struct tagDCHub * DCHubPtr`

8.16.2.29 `typedef struct tagDCInterface DCInterface`

8.16.2.30 `typedef struct tagDCInterface * DCInterfacePtr`

8.16.2.31 `typedef struct tagDCLog DCLog`

8.16.2.32 `typedef struct tagDCLog * DCLogPtr`

8.16.2.33 `typedef void * dcptr_t`

Definition at line 164 of file PluginDefs.h.

8.16.2.34 `typedef struct tagDCQueue DCQueue`

8.16.2.35 `typedef struct tagDCQueue * DCQueuePtr`

8.16.2.36 `typedef struct tagDCTagger DCTagger`

8.16.2.37 `typedef struct tagDCTagger * DCTaggerPtr`

8.16.2.38 `typedef uint64_t dctime_t`

Definition at line 166 of file PluginDefs.h.

8.16.2.39 `typedef struct DCUI DCUI`

8.16.2.40 `typedef struct DCUI * DCUIPtr`

8.16.2.41 `typedef struct tagDCUtils DCUtils`

8.16.2.42 `typedef struct tagDCUtils * DCUtilsPtr`

8.16.2.43 `typedef void* hookHandle`

Definition at line 164 of file PluginDefs.h.

8.16.2.44 `typedef struct tagHubData HubData`

8.16.2.45 `typedef struct tagHubData * HubDataPtr`

8.16.2.46 `typedef void * intfHandle`

Definition at line 164 of file PluginDefs.h.

8.16.2.47 `typedef struct tagMetaData MetaData`

8.16.2.48 `typedef struct tagMetaData * MetaDataPtr`

8.16.2.49 `typedef enum tagMsgType MsgType`

8.16.2.50 `typedef enum tagPathType PathType`

8.16.2.51 `typedef dcptr_t dcptr_t pCommon`

Definition at line 308 of file PluginDefs.h.

8.16.2.52 `typedef dcptr_t pData`

Definition at line 303 of file PluginDefs.h.

8.16.2.53 `typedef enum tagPluginState PluginState`

8.16.2.54 `typedef enum tagProtocolType ProtocolType`

8.16.2.55 `typedef struct tagQueueData QueueData`

8.16.2.56 `typedef struct tagQueueData * QueueDataPtr`

8.16.2.57 `typedef enum tagQueuePrio QueuePrio`

8.16.2.58 `typedef struct tagStringData StringData`

8.16.2.59 `typedef struct tagStringData * StringDataPtr`

8.16.2.60 `typedef void * subsHandle`

Definition at line 164 of file PluginDefs.h.

8.16.2.61 `typedef struct tagTagData TagData`

8.16.2.62 `typedef struct tagTagData * TagDataPtr`

8.16.2.63 `typedef struct tagUserData UserData`

8.16.2.64 `typedef struct tagUserData * UserDataPtr`

8.16.3 Enumeration Type Documentation

8.16.3.1 enum tagConfigType

Enumerator:

CFG_TYPE_UNKNOWN
CFG_TYPE_REMOVE
CFG_TYPE_STRING
CFG_TYPE_INT
CFG_TYPE_BOOL
CFG_TYPE_INT64

Definition at line 125 of file PluginDefs.h.

8.16.3.2 enum tagDCBool

Enumerator:

dcFalse
dcTrue

Definition at line 165 of file PluginDefs.h.

8.16.3.3 enum tagMsgType

Enumerator:

MSG_CLIENT
MSG_STATUS
MSG_SYSTEM

Definition at line 148 of file PluginDefs.h.

8.16.3.4 enum tagPathType

Enumerator:

PATH_GLOBAL_CONFIG
PATH_USER_CONFIG
PATH_USER_LOCAL
PATH_RESOURCES
PATH_LOCALE

Definition at line 140 of file PluginDefs.h.

8.16.3.5 enum tagPluginState

Enumerator:

ON_INSTALL
ON_UNINSTALL
ON_LOAD
ON_UNLOAD
ON_CONFIGURE

Definition at line 116 of file PluginDefs.h.

8.16.3.6 enum tagProtocolType

Enumerator:

PROTOCOL_ADC
PROTOCOL_NMDC
PROTOCOL_DHT

Definition at line 134 of file PluginDefs.h.

8.16.3.7 enum tagQueuePrio

Enumerator:

PRIO_DEFAULT
PRIO_LOWEST
PRIO_LOW
PRIO_NORMAL
PRIO_HIGH
PRIO_HIGHEST

Definition at line 154 of file PluginDefs.h.

8.16.4 Function Documentation

8.16.4.1 typedef Bool (DCAPI * *DCMAIN*)

8.16.4.2 typedef void (DCAPI * *DCCommandFunc*) const

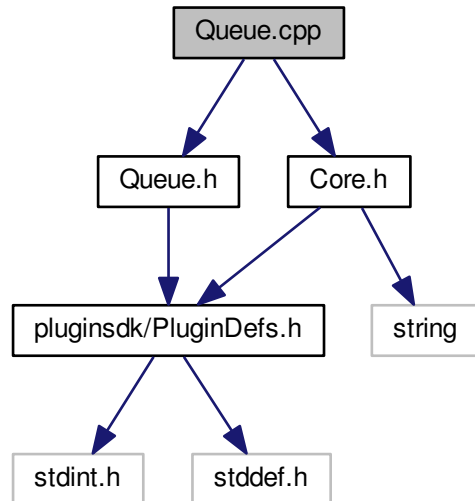
8.17 Queue.cpp File Reference

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



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

Include dependency graph for Queue.cpp:



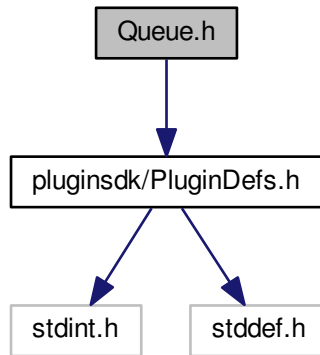
Namespaces

- namespace [dcapi](#)

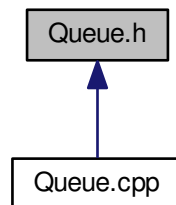
8.18 Queue.h File Reference

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

Include dependency graph for Queue.h:



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



Classes

- class [dcapi::Queue](#)

Namespaces

- namespace [dcapi](#)

8.19 resource.h File Reference

Defines

- #define [VERSION_INFO](#) 1

8.19.1 Define Documentation

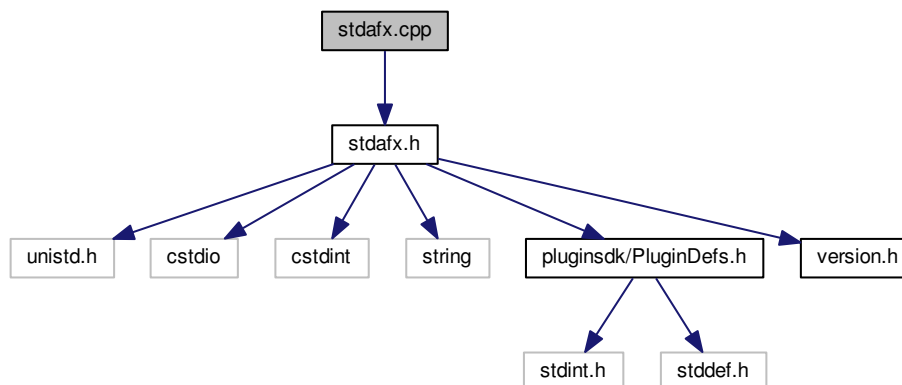
8.19.1.1 #define VERSION_INFO 1

Definition at line 5 of file resource.h.

8.20 stdafx.cpp File Reference

```
#include "stdafx.h"
```

Include dependency graph for stdafx.cpp:



8.21 stdafx.h File Reference

```
#include <unistd.h>
```

```
#include <cstdio>
```

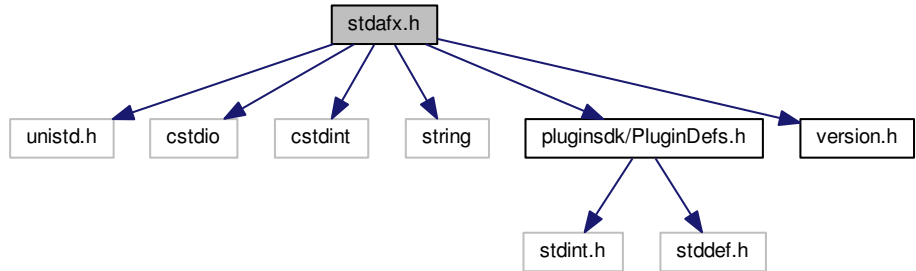
```
#include <stdint>
```

```
#include <string>
```

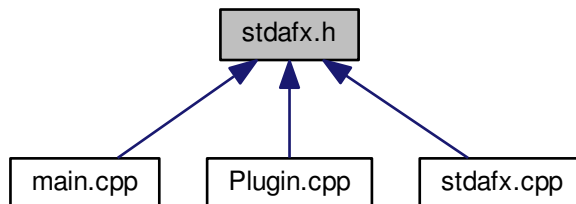
```
#include <pluginsdk/PluginDefs.h>
```

```
#include "version.h"
```

Include dependency graph for stdafx.h:



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



Typedefs

- typedef `std::string` `tstring`

8.21.1 Typedef Documentation

8.21.1.1 typedef `std::string` `tstring`

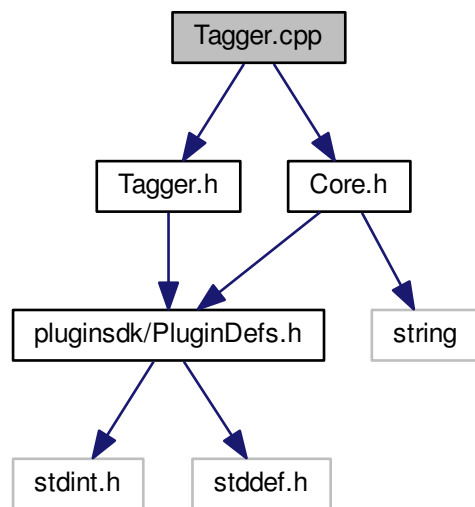
Definition at line 44 of file `stdafx.h`.

8.22 Tagger.cpp File Reference

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

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

Include dependency graph for Tagger.cpp:



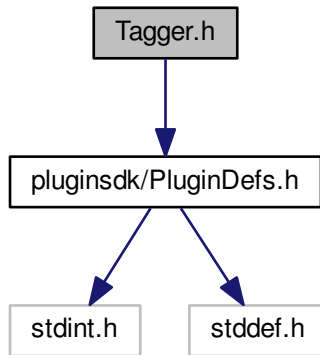
Namespaces

- namespace [dcapi](#)

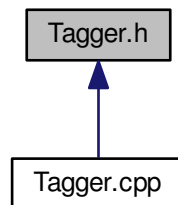
8.23 Tagger.h File Reference

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

Include dependency graph for Tagger.h:



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



Classes

- class [dcapi::Tagger](#)

Namespaces

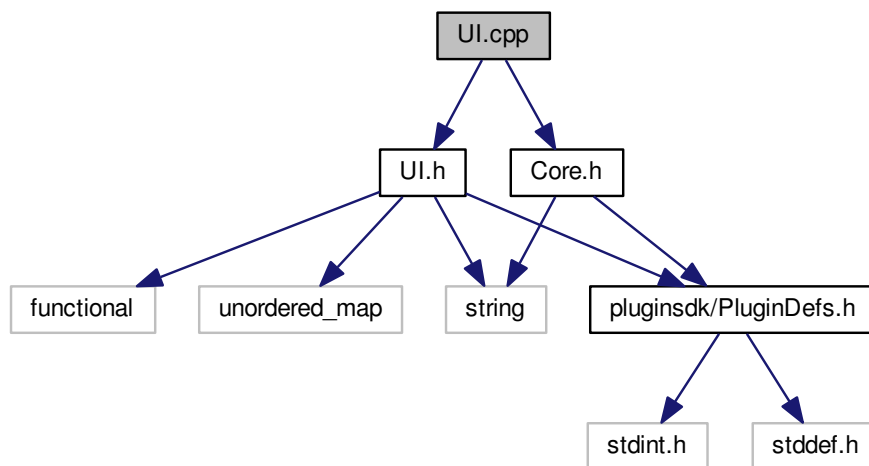
- namespace [dcapi](#)

8.24 UI.cpp File Reference

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

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

Include dependency graph for UI.cpp:



Namespaces

- namespace [dcapi](#)

8.25 UI.h File Reference

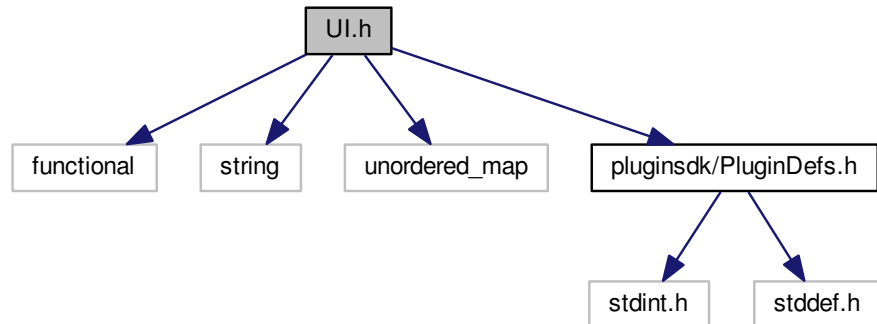
```
#include <functional>
```

```
#include <string>
```

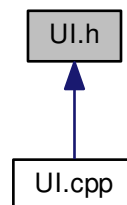
```
#include <unordered_map>
```

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

Include dependency graph for UI.h:



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



Classes

- class [dcapi::UI](#)

Namespaces

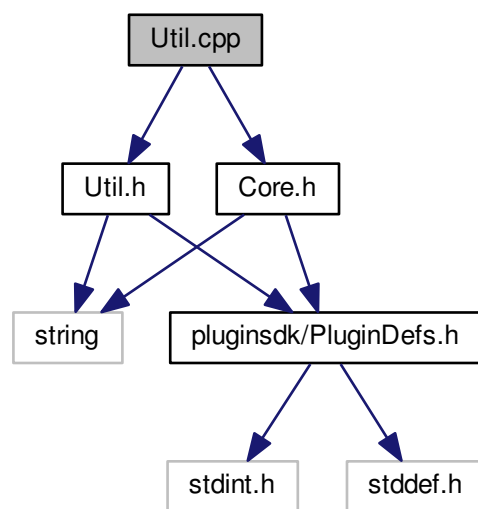
- namespace [dcapi](#)

8.26 Util.cpp File Reference

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

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

Include dependency graph for Util.cpp:



Namespaces

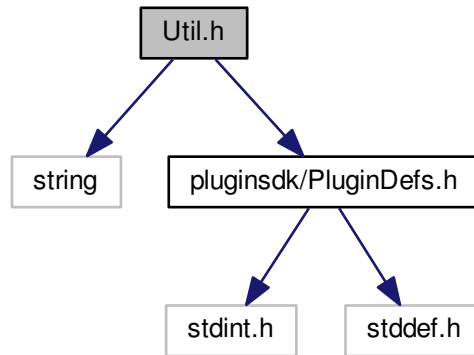
- namespace [dcapi](#)

8.27 Util.h File Reference

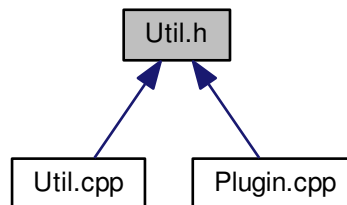
```
#include <string>
```

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

Include dependency graph for Util.h:



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



Classes

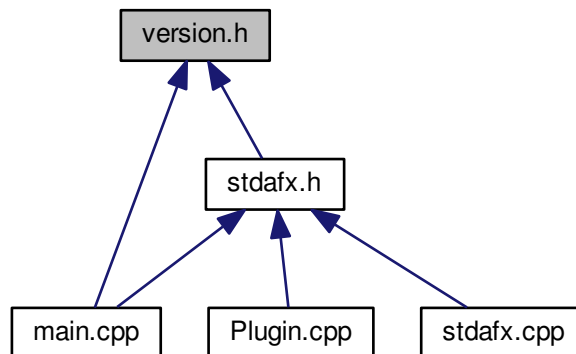
- class [dcapi::Util](#)

Namespaces

- namespace [dcapi](#)

8.28 version.h File Reference

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



Defines

- #define `PLUGIN_AUTHOR` "..."
- #define `PLUGIN_DESC` "..."
- #define `PLUGIN_GUID` "..."
- #define `PLUGIN_NAME` "..."
- #define `PLUGIN_VERSION` 1.0
- #define `PLUGIN_WEB` "N/A"

8.28.1 Define Documentation

8.28.1.1 #define `PLUGIN_AUTHOR` "..."

Definition at line 33 of file version.h.

Referenced by pluginInit().

8.28.1.2 #define `PLUGIN_DESC` "..."

Definition at line 36 of file version.h.

Referenced by pluginInit().

8.28.1.3 #define PLUGIN_GUID "..."

Definition at line 27 of file version.h.

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

8.28.1.4 #define PLUGIN_NAME "..."

Definition at line 30 of file version.h.

Referenced by pluginInit().

8.28.1.5 #define PLUGIN_VERSION 1.0

Definition at line 39 of file version.h.

Referenced by pluginInit().

8.28.1.6 #define PLUGIN_WEB "N/A"

Definition at line 42 of file version.h.

Referenced by pluginInit().

Index

- ~Plugin
 - Plugin, [41](#)
 - /cygdrive/h/Dev/DC++/pluginsdk/cpp/pluginsdk/
Directory Reference, [9](#)
 - /cygdrive/h/Dev/DC++/pluginsdk/cpp/src/ Di-
rectory Reference, [10](#)
- addCommand
 - dcapi::UI, [87](#)
- addEvent
 - dcapi::Hooks, [29](#)
- apiVersion
 - DCUI, [27](#)
 - tagDCConfig, [56](#)
 - tagDCConnection, [57](#)
 - tagDCCore, [59](#)
 - tagDCHooks, [61](#)
 - tagDCHub, [63](#)
 - tagDCInterface, [65](#)
 - tagDCLog, [66](#)
 - tagDCQueue, [67](#)
 - tagDCTagger, [69](#)
 - tagDCUtils, [71](#)
 - tagMetaData, [76](#)
- appName
 - dcapi::Core, [26](#)
- attributes
 - tagDCTagger, [69](#)
- author
 - tagMetaData, [76](#)
- bBreak
 - PluginDefs.h, [119](#)
- Bool
 - PluginDefs.h, [113](#), [124](#)
 - tagDCCore, [59](#)
 - tagDCHooks, [61](#)
 - tagDCHub, [63](#)
 - tagDCQueue, [67](#)
- Callback
 - dcapi::Hooks, [29](#)
- CFG_TYPE_BOOL
 - PluginDefs.h, [123](#)
- CFG_TYPE_INT
 - PluginDefs.h, [123](#)
- CFG_TYPE_INT64
 - PluginDefs.h, [123](#)
- CFG_TYPE_REMOVE
 - PluginDefs.h, [123](#)
- CFG_TYPE_STRING
 - PluginDefs.h, [123](#)
- CFG_TYPE_UNKNOWN
 - PluginDefs.h, [123](#)
- cid
 - tagUserData, [82](#)
- clear
 - dcapi::Hooks, [31](#)
- cmd
 - tagDCConnection, [57](#)
 - tagDCHub, [63](#)
- Command
 - dcapi::UI, [87](#)
- command
 - DCUI, [27](#)
 - tagCommandData, [48](#)
- commandCallback
 - dcapi::UI, [87](#)
- CommandData
 - PluginDefs.h, [119](#)
- CommandDataPtr
 - PluginDefs.h, [119](#)
- commands
 - dcapi::UI, [89](#)
- config
 - dcapi::Config, [19](#)
- Config.cpp, [95](#)
- Config.h, [96](#)
- ConfigBool
 - PluginDefs.h, [119](#)
- ConfigBoolPtr
 - PluginDefs.h, [120](#)

- ConfigInt
 - PluginDefs.h, 120
- ConfigInt64
 - PluginDefs.h, 120
- ConfigInt64Ptr
 - PluginDefs.h, 120
- ConfigIntPtr
 - PluginDefs.h, 120
- ConfigStr
 - PluginDefs.h, 120
- ConfigStrPtr
 - PluginDefs.h, 120
- ConfigType
 - PluginDefs.h, 120
- ConfigValue
 - PluginDefs.h, 120
- ConfigValuePtr
 - PluginDefs.h, 120
 - tagDCConfig, 55
- ConnectionData
 - PluginDefs.h, 120
- ConnectionDataPtr
 - PluginDefs.h, 120
- connections
 - dcapi::Connections, 22
- Connections.cpp, 97
- Connections.h, 98
- core
 - dcapi::Core, 26
 - PluginDefs.h, 120
- Core.cpp, 99
- Core.h, 99
- data
 - tagDCConnection, 57
- DCAPI
 - PluginDefs.h, 113
- dcapi, 11
- dcapi::Config, 15
 - config, 19
 - freeCoreConfig, 16
 - getBoolConfig, 16
 - getConfig, 16
 - getCoreConfig, 17
 - getInt64Config, 17
 - getIntConfig, 17
 - getPath, 17
 - guid, 20
 - handle, 17
 - init, 17, 18
 - setConfig, 18, 19
- dcapi::Connections, 20
 - connections, 22
 - handle, 21
 - init, 21, 22
- dcapi::Core, 22
 - appName, 26
 - core, 26
 - handle, 23
 - init, 25
- dcapi::Hooks, 27
 - addEvent, 29
 - Callback, 29
 - clear, 31
 - empty, 31
 - eventCallback, 31
 - events, 33
 - handle, 33
 - hooks, 33
 - init, 33
 - remove, 33
- dcapi::Hooks::Chat, 13
 - onIncomingChat, 13
 - onIncomingPM, 14
 - onOutgoingChat, 14
 - onOutgoingPM, 14
- dcapi::Hooks::Hubs, 36
 - onOffline, 36
 - onOnline, 36
- dcapi::Hooks::Network, 39
 - onClientDataIn, 39
 - onClientDataOut, 39
 - onHubDataIn, 40
 - onHubDataOut, 40
- dcapi::Hooks::Queue, 45
 - onAdded, 46
 - onFinished, 46
 - onMoved, 46
 - onRemoved, 47
- dcapi::Hooks::Timer, 82
 - onMinute, 83
 - onSecond, 83
- dcapi::Hooks::UI, 84
 - onChatCommand, 84
 - onChatCommandPM, 84
 - onChatDisplay, 85
 - onChatTags, 85
 - onCreated, 85
- dcapi::Hooks::Users, 89
 - onOffline, 90

- onOnline, 90
- dcapi::Hubs, 34
 - handle, 34
 - hubs, 36
 - init, 35
- dcapi::Logger, 37
 - handle, 38
 - init, 38
 - log, 38
 - logger, 39
- dcapi::Queue, 43
 - handle, 44
 - init, 44, 45
 - queue, 45
- dcapi::Tagger, 72
 - handle, 73
 - init, 73
 - tagger, 74
- dcapi::UI, 86
 - addCommand, 87
 - Command, 87
 - commandCallback, 87
 - commands, 89
 - handle, 88
 - init, 88
 - removeCommand, 89
 - ui, 89
- dcapi::Util, 90
 - fromUtf8, 91
 - handle, 92
 - init, 92
 - toUtf8, 92
 - utils, 92
- DCAPI_CORE_VER
 - PluginDefs.h, 114
- dcBool
 - PluginDefs.h, 120
- DCConfig
 - PluginDefs.h, 120
- DCConfigPtr
 - PluginDefs.h, 120
- DCConnection
 - PluginDefs.h, 120
- DCConnectionPtr
 - PluginDefs.h, 120
- DCCore
 - PluginDefs.h, 120
- DCCorePtr
 - PluginDefs.h, 120
- DCEXP
 - PluginDefs.h, 114
- dcFalse
 - PluginDefs.h, 123
- DCHooks
 - PluginDefs.h, 120
- DCHooksPtr
 - PluginDefs.h, 120
- DCHub
 - PluginDefs.h, 120
- DCHubPtr
 - PluginDefs.h, 121
- DCIMP
 - PluginDefs.h, 114
- DCInterface
 - PluginDefs.h, 121
- DCInterfacePtr
 - PluginDefs.h, 121
 - tagDCCore, 59
- DCINTF_CONFIG
 - PluginDefs.h, 114
- DCINTF_CONFIG_VER
 - PluginDefs.h, 114
- DCINTF_DCPP_CONNECTIONS
 - PluginDefs.h, 114
- DCINTF_DCPP_CONNECTIONS_VER
 - PluginDefs.h, 114
- DCINTF_DCPP_HUBS
 - PluginDefs.h, 114
- DCINTF_DCPP_HUBS_VER
 - PluginDefs.h, 114
- DCINTF_DCPP_QUEUE
 - PluginDefs.h, 115
- DCINTF_DCPP_QUEUE_VER
 - PluginDefs.h, 115
- DCINTF_DCPP_TAGGER
 - PluginDefs.h, 115
- DCINTF_DCPP_TAGGER_VER
 - PluginDefs.h, 115
- DCINTF_DCPP_UI
 - PluginDefs.h, 115
- DCINTF_DCPP_UI_VER
 - PluginDefs.h, 115
- DCINTF_DCPP_UTILS
 - PluginDefs.h, 115
- DCINTF_DCPP_UTILS_VER
 - PluginDefs.h, 115
- DCINTF_HOOKS
 - PluginDefs.h, 116
- DCINTF_HOOKS_VER
 - PluginDefs.h, 116

- DCINTF_LOGGING
 - PluginDefs.h, [116](#)
- DCINTF_LOGGING_VER
 - PluginDefs.h, [116](#)
- DCLog
 - PluginDefs.h, [121](#)
- DCLogPtr
 - PluginDefs.h, [121](#)
- dcptr_t
 - PluginDefs.h, [121](#)
- DCQueue
 - PluginDefs.h, [121](#)
- DCQueuePtr
 - PluginDefs.h, [121](#)
- DCTagger
 - PluginDefs.h, [121](#)
- DCTaggerPtr
 - PluginDefs.h, [121](#)
- dctime_t
 - PluginDefs.h, [121](#)
- dcTrue
 - PluginDefs.h, [123](#)
- DCUI, [26](#)
 - apiVersion, [27](#)
 - command, [27](#)
 - PluginDefs.h, [121](#)
 - void, [27](#)
- DCUIPtr
 - PluginDefs.h, [121](#)
- DCUtils
 - PluginDefs.h, [121](#)
- DCUtilsPtr
 - PluginDefs.h, [121](#)
- defProc
 - tagDCHooks, [61](#)
- dependencies
 - tagMetaData, [77](#)
- description
 - tagMetaData, [77](#)
- empty
 - dcapi::Hooks, [31](#)
- end
 - tagDCTagger, [69](#)
- eventCallback
 - dcapi::Hooks, [31](#)
- events
 - dcapi::Hooks, [33](#)
- False
 - PluginDefs.h, [116](#)
- freeCoreConfig
 - dcapi::Config, [16](#)
- fromUtf8
 - dcapi::Util, [91](#)
- get_path
 - tagDCConfig, [55](#)
- get_text
 - tagDCTagger, [69](#)
- getBoolConfig
 - dcapi::Config, [16](#)
- getConfig
 - dcapi::Config, [16](#)
- getCoreConfig
 - dcapi::Config, [17](#)
- getInt64Config
 - dcapi::Config, [17](#)
- getIntConfig
 - dcapi::Config, [17](#)
- getPath
 - dcapi::Config, [17](#)
- graceless
 - tagDCConnection, [57](#)
- guid
 - dcapi::Config, [20](#)
 - tagMetaData, [77](#)
- handle
 - dcapi::Config, [17](#)
 - dcapi::Connections, [21](#)
 - dcapi::Core, [23](#)
 - dcapi::Hooks, [33](#)
 - dcapi::Hubs, [34](#)
 - dcapi::Logger, [38](#)
 - dcapi::Queue, [44](#)
 - dcapi::Tagger, [73](#)
 - dcapi::UI, [88](#)
 - dcapi::Util, [92](#)
- hash
 - tagQueueData, [78](#)
- HOOK_CHAT_IN
 - PluginDefs.h, [116](#)
- HOOK_CHAT_OUT
 - PluginDefs.h, [116](#)
- HOOK_CHAT_PM_IN
 - PluginDefs.h, [116](#)
- HOOK_CHAT_PM_OUT
 - PluginDefs.h, [117](#)
- HOOK_HUB_OFFLINE

- PluginDefs.h, [117](#)
- HOOK_HUB_ONLINE
 - PluginDefs.h, [117](#)
- HOOK_NETWORK_CONN_IN
 - PluginDefs.h, [117](#)
- HOOK_NETWORK_CONN_OUT
 - PluginDefs.h, [117](#)
- HOOK_NETWORK_HUB_IN
 - PluginDefs.h, [117](#)
- HOOK_NETWORK_HUB_OUT
 - PluginDefs.h, [117](#)
- HOOK_QUEUE_ADDED
 - PluginDefs.h, [117](#)
- HOOK_QUEUE_FINISHED
 - PluginDefs.h, [118](#)
- HOOK_QUEUE_MOVED
 - PluginDefs.h, [118](#)
- HOOK_QUEUE_REMOVED
 - PluginDefs.h, [118](#)
- HOOK_TIMER_MINUTE
 - PluginDefs.h, [118](#)
- HOOK_TIMER_SECOND
 - PluginDefs.h, [118](#)
- HOOK_UI_CHAT_COMMAND
 - PluginDefs.h, [118](#)
- HOOK_UI_CHAT_COMMAND_PM
 - PluginDefs.h, [118](#)
- HOOK_UI_CHAT_DISPLAY
 - PluginDefs.h, [118](#)
- HOOK_UI_CHAT_TAGS
 - PluginDefs.h, [119](#)
- HOOK_UI_CREATED
 - PluginDefs.h, [119](#)
- HOOK_USER_OFFLINE
 - PluginDefs.h, [119](#)
- HOOK_USER_ONLINE
 - PluginDefs.h, [119](#)
- hookHandle
 - PluginDefs.h, [121](#)
 - tagDCHooks, [61](#)
- hookProc
 - tagDCHooks, [61](#)
- hooks
 - dcapi::Hooks, [33](#)
- Hooks.cpp, [100](#)
- Hooks.h, [101](#)
- host_name
 - tagDCCore, [59](#)
- HubData
 - PluginDefs.h, [121](#)
- HubDataPtr
 - PluginDefs.h, [121](#)
 - tagDCHub, [63](#)
- hubHint
 - tagUserData, [82](#)
- hubs
 - dcapi::Hubs, [36](#)
- Hubs.cpp, [103](#)
- Hubs.h, [103](#)
- hubUrl
 - tagDCHub, [63](#)
- id
 - tagDCTagger, [69](#)
- in
 - tagStringData, [80](#)
- init
 - dcapi::Config, [17, 18](#)
 - dcapi::Connections, [21, 22](#)
 - dcapi::Core, [25](#)
 - dcapi::Hooks, [33](#)
 - dcapi::Hubs, [35](#)
 - dcapi::Logger, [38](#)
 - dcapi::Queue, [44, 45](#)
 - dcapi::Tagger, [73](#)
 - dcapi::UI, [88](#)
 - dcapi::Util, [92](#)
- intfHandle
 - PluginDefs.h, [121](#)
 - tagDCCore, [59](#)
- ip
 - tagConnectionData, [53](#)
 - tagHubData, [75](#)
- isFileList
 - tagQueueData, [78](#)
- isManaged
 - tagConnectionData, [53](#)
 - tagHubData, [75](#)
 - tagQueueData, [78](#)
 - tagTagData, [81](#)
 - tagUserData, [82](#)
- isOp
 - tagConnectionData, [53](#)
 - tagHubData, [75](#)
 - tagUserData, [82](#)
- isSecure
 - tagConnectionData, [53](#)
 - tagHubData, [75](#)
- location

- tagQueueData, 79
- log
 - dcapi::Logger, 38
- logger
 - dcapi::Logger, 39
- Logger.cpp, 105
- Logger.h, 105
- main
 - Plugin, 42
- main.cpp, 107
- pluginInit, 107
- MetaData
 - PluginDefs.h, 122
- MetaDataPtr
 - PluginDefs.h, 122
- msg
 - tagDCHub, 63
- MSG_CLIENT
 - PluginDefs.h, 123
- MSG_STATUS
 - PluginDefs.h, 123
- MSG_SYSTEM
 - PluginDefs.h, 123
- MsgType
 - PluginDefs.h, 122
- n
 - tagDCCConnection, 57
 - tagDCUtils, 71
- name
 - tagMetaData, 77
- nick
 - tagDCHub, 63
 - tagUserData, 82
- numDependencies
 - tagMetaData, 77
- object
 - tagConnectionData, 54
 - tagHubData, 75
 - tagQueueData, 79
 - tagTagData, 81
 - tagUserData, 82
- ON_CONFIGURE
 - PluginDefs.h, 124
- ON_INSTALL
 - PluginDefs.h, 124
- ON_LOAD
 - PluginDefs.h, 124
- ON_UNINSTALL
 - PluginDefs.h, 124
- ON_UNLOAD
 - PluginDefs.h, 124
- onAdded
 - dcapi::Hooks::Queue, 46
- onChatCommand
 - dcapi::Hooks::UI, 84
- onChatCommandPM
 - dcapi::Hooks::UI, 84
- onChatDisplay
 - dcapi::Hooks::UI, 85
- onChatTags
 - dcapi::Hooks::UI, 85
- onClientDataIn
 - dcapi::Hooks::Network, 39
- onClientDataOut
 - dcapi::Hooks::Network, 39
- onCreated
 - dcapi::Hooks::UI, 85
- onFinished
 - dcapi::Hooks::Queue, 46
- onHubDataIn
 - dcapi::Hooks::Network, 40
- onHubDataOut
 - dcapi::Hooks::Network, 40
- onIncomingChat
 - dcapi::Hooks::Chat, 13
- onIncomingPM
 - dcapi::Hooks::Chat, 14
- onLoad
 - Plugin, 42
- onMinute
 - dcapi::Hooks::Timer, 83
- onMoved
 - dcapi::Hooks::Queue, 46
- onOffline
 - dcapi::Hooks::Hubs, 36
 - dcapi::Hooks::Users, 90
- onOnline
 - dcapi::Hooks::Hubs, 36
 - dcapi::Hooks::Users, 90
- onOutgoingChat
 - dcapi::Hooks::Chat, 14
- onOutgoingPM
 - dcapi::Hooks::Chat, 14
- onRemoved
 - dcapi::Hooks::Queue, 47
- onSecond
 - dcapi::Hooks::Timer, 83

- out
 - tagStringData, [80](#)
- params
 - tagCommandData, [48](#)
- password
 - tagDCHub, [64](#)
- PATH_GLOBAL_CONFIG
 - PluginDefs.h, [123](#)
- PATH_LOCALE
 - PluginDefs.h, [123](#)
- PATH_RESOURCES
 - PluginDefs.h, [123](#)
- PATH_USER_CONFIG
 - PluginDefs.h, [123](#)
- PATH_USER_LOCAL
 - PluginDefs.h, [123](#)
- PathType
 - PluginDefs.h, [122](#)
- pCommon
 - PluginDefs.h, [122](#)
 - tagDCHooks, [61](#)
- pData
 - PluginDefs.h, [122](#)
 - tagDCHooks, [61](#)
- plInterface
 - tagDCCore, [59](#)
- Plugin, [41](#)
 - ~Plugin, [41](#)
 - main, [42](#)
 - onLoad, [42](#)
 - Plugin, [41](#)
- Plugin.cpp, [108](#)
- Plugin.h, [109](#)
- PLUGIN_AUTHOR
 - version.h, [135](#)
- PLUGIN_DESC
 - version.h, [135](#)
- PLUGIN_GUID
 - version.h, [135](#)
- PLUGIN_NAME
 - version.h, [136](#)
- PLUGIN_VERSION
 - version.h, [136](#)
- PLUGIN_WEB
 - version.h, [136](#)
- PluginDefs.h, [109](#)
 - bBreak, [119](#)
 - Bool, [113](#), [124](#)
 - CFG_TYPE_BOOL, [123](#)
 - CFG_TYPE_INT, [123](#)
 - CFG_TYPE_INT64, [123](#)
 - CFG_TYPE_REMOVE, [123](#)
 - CFG_TYPE_STRING, [123](#)
 - CFG_TYPE_UNKNOWN, [123](#)
 - CommandData, [119](#)
 - CommandDataPtr, [119](#)
 - ConfigBool, [119](#)
 - ConfigBoolPtr, [120](#)
 - ConfigInt, [120](#)
 - ConfigInt64, [120](#)
 - ConfigInt64Ptr, [120](#)
 - ConfigIntPtr, [120](#)
 - ConfigStr, [120](#)
 - ConfigStrPtr, [120](#)
 - ConfigType, [120](#)
 - ConfigValue, [120](#)
 - ConfigValuePtr, [120](#)
 - ConnectionData, [120](#)
 - ConnectionDataPtr, [120](#)
 - core, [120](#)
 - DCAPI, [113](#)
 - DCAPI_CORE_VER, [114](#)
 - dcBool, [120](#)
 - DCConfig, [120](#)
 - DCConfigPtr, [120](#)
 - DCConnection, [120](#)
 - DCConnectionPtr, [120](#)
 - DCCore, [120](#)
 - DCCorePtr, [120](#)
 - DCEXP, [114](#)
 - dcFalse, [123](#)
 - DCHooks, [120](#)
 - DCHooksPtr, [120](#)
 - DCHub, [120](#)
 - DCHubPtr, [121](#)
 - DCIMP, [114](#)
 - DCInterface, [121](#)
 - DCInterfacePtr, [121](#)
 - DCINTF_CONFIG, [114](#)
 - DCINTF_CONFIG_VER, [114](#)
 - DCINTF_DCPP_CONNECTIONS, [114](#)
 - DCINTF_DCPP_CONNECTIONS_VER, [114](#)
 - DCINTF_DCPP_HUBS, [114](#)
 - DCINTF_DCPP_HUBS_VER, [114](#)
 - DCINTF_DCPP_QUEUE, [115](#)
 - DCINTF_DCPP_QUEUE_VER, [115](#)
 - DCINTF_DCPP_TAGGER, [115](#)
 - DCINTF_DCPP_TAGGER_VER, [115](#)

DCINTF_DCPP_UI, 115
 DCINTF_DCPP_UI_VER, 115
 DCINTF_DCPP_UTILS, 115
 DCINTF_DCPP_UTILS_VER, 115
 DCINTF_HOOKS, 116
 DCINTF_HOOKS_VER, 116
 DCINTF_LOGGING, 116
 DCINTF_LOGGING_VER, 116
 DCLog, 121
 DCLogPtr, 121
 dcptr_t, 121
 DCQueue, 121
 DCQueuePtr, 121
 DCTagger, 121
 DCTaggerPtr, 121
 dctime_t, 121
 dcTrue, 123
 DCUI, 121
 DCUIPtr, 121
 DCUtils, 121
 DCUtilsPtr, 121
 False, 116
 HOOK_CHAT_IN, 116
 HOOK_CHAT_OUT, 116
 HOOK_CHAT_PM_IN, 116
 HOOK_CHAT_PM_OUT, 117
 HOOK_HUB_OFFLINE, 117
 HOOK_HUB_ONLINE, 117
 HOOK_NETWORK_CONN_IN, 117
 HOOK_NETWORK_CONN_OUT, 117
 HOOK_NETWORK_HUB_IN, 117
 HOOK_NETWORK_HUB_OUT, 117
 HOOK_QUEUE_ADDED, 117
 HOOK_QUEUE_FINISHED, 118
 HOOK_QUEUE_MOVED, 118
 HOOK_QUEUE_REMOVED, 118
 HOOK_TIMER_MINUTE, 118
 HOOK_TIMER_SECOND, 118
 HOOK_UI_CHAT_COMMAND, 118
 HOOK_UI_CHAT_COMMAND_PM, 118
 HOOK_UI_CHAT_DISPLAY, 118
 HOOK_UI_CHAT_TAGS, 119
 HOOK_UI_CREATED, 119
 HOOK_USER_OFFLINE, 119
 HOOK_USER_ONLINE, 119
 hookHandle, 121
 HubData, 121
 HubDataPtr, 121
 intfHandle, 121
 MetaData, 122
 MetaDataPtr, 122
 MSG_CLIENT, 123
 MSG_STATUS, 123
 MSG_SYSTEM, 123
 MsgType, 122
 ON_CONFIGURE, 124
 ON_INSTALL, 124
 ON_LOAD, 124
 ON_UNINSTALL, 124
 ON_UNLOAD, 124
 PATH_GLOBAL_CONFIG, 123
 PATH_LOCALE, 123
 PATH_RESOURCES, 123
 PATH_USER_CONFIG, 123
 PATH_USER_LOCAL, 123
 PathType, 122
 pCommon, 122
 pData, 122
 PluginState, 122
 PRIO_DEFAULT, 124
 PRIO_HIGH, 124
 PRIO_HIGHEST, 124
 PRIO_LOW, 124
 PRIO_LOWEST, 124
 PRIO_NORMAL, 124
 PROTOCOL_ADC, 124
 PROTOCOL_DHT, 124
 PROTOCOL_NMDC, 124
 ProtocolType, 122
 QueueData, 122
 QueueDataPtr, 122
 QueuePrio, 122
 StringData, 122
 StringDataPtr, 122
 subsHandle, 122
 tagConfigType, 123
 TagData, 122
 TagDataPtr, 122
 tagDCBool, 123
 tagMsgType, 123
 tagPathType, 123
 tagPluginState, 123
 tagProtocolType, 124
 tagQueuePrio, 124
 True, 119
 UserData, 122
 UserDataPtr, 122
 void, 124
 pluginInit
 main.cpp, 107

- PluginState
 - PluginDefs.h, [122](#)
- pObject
 - tagDCHooks, [61](#)
- port
 - tagConnectionData, [54](#)
 - tagDCConnection, [58](#)
 - tagHubData, [75](#)
- PRIO_DEFAULT
 - PluginDefs.h, [124](#)
- PRIO_HIGH
 - PluginDefs.h, [124](#)
- PRIO_HIGHEST
 - PluginDefs.h, [124](#)
- PRIO_LOW
 - PluginDefs.h, [124](#)
- PRIO_LOWEST
 - PluginDefs.h, [124](#)
- PRIO_NORMAL
 - PluginDefs.h, [124](#)
- priority
 - tagDCQueue, [67](#)
- protocol
 - tagConnectionData, [54](#)
 - tagHubData, [75](#)
 - tagUserData, [82](#)
- PROTOCOL_ADC
 - PluginDefs.h, [124](#)
- PROTOCOL_DHT
 - PluginDefs.h, [124](#)
- PROTOCOL_NMDC
 - PluginDefs.h, [124](#)
- ProtocolType
 - PluginDefs.h, [122](#)
- queue
 - dcapi::Queue, [45](#)
- Queue.cpp, [124](#)
- Queue.h, [125](#)
- QueueData
 - PluginDefs.h, [122](#)
- QueueDataPtr
 - PluginDefs.h, [122](#)
 - tagDCQueue, [67](#)
- QueuePrio
 - PluginDefs.h, [122](#)
- remove
 - dcapi::Hooks, [33](#)
- removeCommand
 - dcapi::UI, [89](#)
- replacement
 - tagDCTagger, [69](#)
- resource.h, [127](#)
 - VERSION_INFO, [127](#)
- setConfig
 - dcapi::Config, [18, 19](#)
- setting
 - tagDCConfig, [56](#)
- sid
 - tagUserData, [82](#)
- silent
 - tagDCQueue, [67](#)
- size
 - tagDCQueue, [67](#)
 - tagQueueData, [79](#)
- size_t
 - tagDCHooks, [61](#)
 - tagDCUtils, [71](#)
- src
 - tagDCUtils, [71, 72](#)
- start
 - tagDCTagger, [69](#)
- stdafx.cpp, [127](#)
- stdafx.h, [127](#)
 - tstring, [128](#)
- StringData
 - PluginDefs.h, [122](#)
- StringDataPtr
 - PluginDefs.h, [122](#)
- subsHandle
 - PluginDefs.h, [122](#)
 - tagDCHooks, [61](#)
- tagCommandData, [47](#)
 - command, [48](#)
 - params, [48](#)
- tagConfigBool, [48](#)
 - type, [49](#)
 - value, [49](#)
- tagConfigInt, [49](#)
 - type, [50](#)
 - value, [50](#)
- tagConfigInt64, [50](#)
 - type, [50](#)
 - value, [51](#)
- tagConfigStr, [51](#)
 - type, [51](#)
 - value, [51](#)

- tagConfigType
 - PluginDefs.h, 123
- tagConfigValue, 52
 - type, 52
- tagConnectionData, 52
 - ip, 53
 - isManaged, 53
 - isOp, 53
 - isSecure, 53
 - object, 54
 - port, 54
 - protocol, 54
- TagData
 - PluginDefs.h, 122
- TagDataPtr
 - PluginDefs.h, 122
- tagDCBool
 - PluginDefs.h, 123
- tagDCConfig, 54
 - apiVersion, 56
 - ConfigValuePtr, 55
 - get_path, 55
 - setting, 56
 - type, 56
 - val, 56
 - void, 55
- tagDCConnection, 56
 - apiVersion, 57
 - cmd, 57
 - data, 57
 - graceless, 57
 - n, 57
 - port, 58
 - UserDataPtr, 57
 - void, 57
- tagDCCore, 58
 - apiVersion, 59
 - Bool, 59
 - DCInterfacePtr, 59
 - host_name, 59
 - intfHandle, 59
 - plInterface, 59
 - version, 59
- tagDCHooks, 60
 - apiVersion, 61
 - Bool, 61
 - defProc, 61
 - hookHandle, 61
 - hookProc, 61
 - pCommon, 61
 - pData, 61
 - pObject, 61
 - size_t, 61
 - subsHandle, 61
- tagDCHub, 61
 - apiVersion, 63
 - Bool, 63
 - cmd, 63
 - HubDataPtr, 63
 - hubUrl, 63
 - msg, 63
 - nick, 63
 - password, 64
 - thirdPerson, 64
 - type, 64
 - UserDataPtr, 63
 - void, 63
- tagDCInterface, 64
 - apiVersion, 65
- tagDCLog, 65
 - apiVersion, 66
 - void, 65
- tagDCQueue, 66
 - apiVersion, 67
 - Bool, 67
 - priority, 67
 - QueueDataPtr, 67
 - silent, 67
 - size, 67
 - target, 67
 - void, 67
- tagDCTagger, 68
 - apiVersion, 69
 - attributes, 69
 - end, 69
 - get_text, 69
 - id, 69
 - replacement, 69
 - start, 69
 - void, 69
- tagDCUtils, 69
 - apiVersion, 71
 - n, 71
 - size_t, 71
 - src, 71, 72
- tagger
 - dcapi::Tagger, 74
- Tagger.cpp, 129
- Tagger.h, 129
- tagHubData, 74

- ip, [75](#)
- isManaged, [75](#)
- isOp, [75](#)
- isSecure, [75](#)
- object, [75](#)
- port, [75](#)
- protocol, [75](#)
- url, [75](#)
- tagMetaData, [75](#)
 - apiVersion, [76](#)
 - author, [76](#)
 - dependencies, [77](#)
 - description, [77](#)
 - guid, [77](#)
 - name, [77](#)
 - numDependencies, [77](#)
 - version, [77](#)
 - web, [77](#)
- tagMsgType
 - PluginDefs.h, [123](#)
- tagPathType
 - PluginDefs.h, [123](#)
- tagPluginState
 - PluginDefs.h, [123](#)
- tagProtocolType
 - PluginDefs.h, [124](#)
- tagQueueData, [78](#)
 - hash, [78](#)
 - isFileList, [78](#)
 - isManaged, [78](#)
 - location, [79](#)
 - object, [79](#)
 - size, [79](#)
 - target, [79](#)
- tagQueuePrio
 - PluginDefs.h, [124](#)
- tagStringData, [79](#)
 - in, [80](#)
 - out, [80](#)
- tagTagData, [80](#)
 - isManaged, [81](#)
 - object, [81](#)
- tagUserData, [81](#)
 - cid, [82](#)
 - hubHint, [82](#)
 - isManaged, [82](#)
 - isOp, [82](#)
 - nick, [82](#)
 - object, [82](#)
 - protocol, [82](#)
 - sid, [82](#)
- target
 - tagDCQueue, [67](#)
 - tagQueueData, [79](#)
- thirdPerson
 - tagDCHub, [64](#)
- toUtf8
 - dcapi::Util, [92](#)
- True
 - PluginDefs.h, [119](#)
- tstring
 - stdafx.h, [128](#)
- type
 - tagConfigBool, [49](#)
 - tagConfigInt, [50](#)
 - tagConfigInt64, [50](#)
 - tagConfigStr, [51](#)
 - tagConfigValue, [52](#)
 - tagDCCConfig, [56](#)
 - tagDCHub, [64](#)
- ui
 - dcapi::UI, [89](#)
- UI.cpp, [131](#)
- UI.h, [131](#)
- url
 - tagHubData, [75](#)
- UserData
 - PluginDefs.h, [122](#)
- UserDataPtr
 - PluginDefs.h, [122](#)
 - tagDCConnection, [57](#)
 - tagDCHub, [63](#)
- Util.cpp, [133](#)
- Util.h, [133](#)
- utils
 - dcapi::Util, [92](#)
- val
 - tagDCCConfig, [56](#)
- value
 - tagConfigBool, [49](#)
 - tagConfigInt, [50](#)
 - tagConfigInt64, [51](#)
 - tagConfigStr, [51](#)
- version
 - tagDCCore, [59](#)
 - tagMetaData, [77](#)
- version.h, [135](#)
 - PLUGIN_AUTHOR, [135](#)

- PLUGIN_DESC, [135](#)
- PLUGIN_GUID, [135](#)
- PLUGIN_NAME, [136](#)
- PLUGIN_VERSION, [136](#)
- PLUGIN_WEB, [136](#)
- VERSION_INFO
 - resource.h, [127](#)
- void
 - DCUI, [27](#)
 - PluginDefs.h, [124](#)
 - tagDCConfig, [55](#)
 - tagDCConnection, [57](#)
 - tagDCHub, [63](#)
 - tagDCLog, [65](#)
 - tagDCQueue, [67](#)
 - tagDCTagger, [69](#)
- web
 - tagMetaData, [77](#)