

DC plugin SDK (C++)

Generated by Doxygen 1.8.4

Fri Jul 26 2013 15:24:46

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Namespace Documentation	7
4.1	dcapi Namespace Reference	7
5	Class Documentation	9
5.1	dcapi::Hooks::Chat Struct Reference	9
5.1.1	Detailed Description	9
5.1.2	Member Function Documentation	9
5.1.2.1	onIncomingChat	9
5.1.2.2	onIncomingPM	9
5.1.2.3	onOutgoingChat	10
5.1.2.4	onOutgoingPM	10
5.2	dcapi::Config Class Reference	10
5.2.1	Detailed Description	12
5.2.2	Member Function Documentation	12
5.2.2.1	freeCoreConfig	12
5.2.2.2	getBoolConfig	12
5.2.2.3	getConfig	12
5.2.2.4	getConfig	12
5.2.2.5	getConfig	12
5.2.2.6	getCoreConfig	12
5.2.2.7	getInstallPath	12
5.2.2.8	getInt64Config	12
5.2.2.9	getIntConfig	13
5.2.2.10	getPath	13

5.2.2.11	handle	13
5.2.2.12	init	13
5.2.2.13	init	13
5.2.2.14	setConfig	13
5.2.2.15	setConfig	14
5.2.2.16	setConfig	14
5.2.2.17	setConfig	14
5.2.2.18	setConfig	14
5.2.2.19	setConfig	14
5.2.2.20	setConfig	14
5.2.3	Member Data Documentation	15
5.2.3.1	config	15
5.2.3.2	guid	15
5.3	dcapi::Connections Class Reference	15
5.3.1	Detailed Description	16
5.3.2	Member Function Documentation	16
5.3.2.1	handle	16
5.3.2.2	init	16
5.3.2.3	init	17
5.3.3	Member Data Documentation	17
5.3.3.1	connections	17
5.4	dcapi::Core Class Reference	17
5.4.1	Detailed Description	18
5.4.2	Member Function Documentation	18
5.4.2.1	handle	18
5.4.2.2	init	19
5.4.3	Member Data Documentation	19
5.4.3.1	appName	19
5.4.3.2	core	19
5.5	DCUI Struct Reference	19
5.5.1	Detailed Description	20
5.5.2	Member Function Documentation	20
5.5.2.1	void	20
5.5.2.2	void	20
5.5.2.3	void	20
5.5.2.4	void	20
5.5.3	Member Data Documentation	21
5.5.3.1	apiVersion	21
5.5.3.2	command	21
5.5.3.3	icon	21

5.5.3.4	message	21
5.5.3.5	name	21
5.6	dcapi::Hooks Class Reference	21
5.6.1	Detailed Description	22
5.6.2	Member Typedef Documentation	22
5.6.2.1	Callback	22
5.6.3	Member Function Documentation	22
5.6.3.1	addEvent	22
5.6.3.2	clear	24
5.6.3.3	empty	25
5.6.3.4	eventCallback	25
5.6.3.5	handle	26
5.6.3.6	init	27
5.6.3.7	init	27
5.6.3.8	remove	27
5.6.4	Member Data Documentation	27
5.6.4.1	events	27
5.6.4.2	hooks	27
5.7	dcapi::Hooks::Hubs Struct Reference	27
5.7.1	Detailed Description	28
5.7.2	Member Function Documentation	28
5.7.2.1	onOffline	28
5.7.2.2	onOnline	28
5.8	dcapi::Hubs Class Reference	28
5.8.1	Detailed Description	29
5.8.2	Member Function Documentation	29
5.8.2.1	handle	29
5.8.2.2	init	29
5.8.2.3	init	30
5.8.3	Member Data Documentation	30
5.8.3.1	hubs	30
5.9	dcapi::Logger Class Reference	30
5.9.1	Detailed Description	31
5.9.2	Member Function Documentation	31
5.9.2.1	handle	31
5.9.2.2	init	31
5.9.2.3	init	32
5.9.2.4	log	32
5.9.3	Member Data Documentation	32
5.9.3.1	logger	32

5.10	dcapi::Hooks::Network Struct Reference	32
5.10.1	Detailed Description	32
5.10.2	Member Function Documentation	33
5.10.2.1	onClientDataIn	33
5.10.2.2	onClientDataOut	33
5.10.2.3	onHubDataIn	33
5.10.2.4	onHubDataOut	33
5.10.2.5	onUDPDataIn	34
5.10.2.6	onUDPDataOut	34
5.11	Plugin Class Reference	34
5.11.1	Detailed Description	35
5.11.2	Constructor & Destructor Documentation	35
5.11.2.1	Plugin	35
5.11.2.2	~Plugin	35
5.11.3	Member Function Documentation	35
5.11.3.1	main	35
5.11.3.2	onLoad	36
5.12	dcapi::Hooks::Queue Struct Reference	36
5.12.1	Detailed Description	37
5.12.2	Member Function Documentation	37
5.12.2.1	onAdded	37
5.12.2.2	onFinished	37
5.12.2.3	onMoved	37
5.12.2.4	onRemoved	38
5.13	dcapi::Queue Class Reference	38
5.13.1	Detailed Description	39
5.13.2	Member Function Documentation	39
5.13.2.1	handle	39
5.13.2.2	init	39
5.13.2.3	init	40
5.13.3	Member Data Documentation	40
5.13.3.1	queue	40
5.14	tagCommandData Struct Reference	40
5.14.1	Detailed Description	40
5.14.2	Member Data Documentation	40
5.14.2.1	command	40
5.14.2.2	params	41
5.15	tagConfigBool Struct Reference	41
5.15.1	Detailed Description	41
5.15.2	Member Data Documentation	41

5.15.2.1	type	41
5.15.2.2	value	41
5.16	tagConfigInt Struct Reference	41
5.16.1	Detailed Description	42
5.16.2	Member Data Documentation	42
5.16.2.1	type	42
5.16.2.2	value	42
5.17	tagConfigInt64 Struct Reference	42
5.17.1	Detailed Description	43
5.17.2	Member Data Documentation	43
5.17.2.1	type	43
5.17.2.2	value	43
5.18	tagConfigStr Struct Reference	43
5.18.1	Detailed Description	44
5.18.2	Member Data Documentation	44
5.18.2.1	type	44
5.18.2.2	value	44
5.19	tagConfigValue Struct Reference	44
5.19.1	Detailed Description	45
5.19.2	Member Data Documentation	45
5.19.2.1	type	45
5.20	tagConnectionData Struct Reference	45
5.20.1	Detailed Description	46
5.20.2	Member Data Documentation	46
5.20.2.1	ip	46
5.20.2.2	isManaged	46
5.20.2.3	isOp	46
5.20.2.4	isSecure	46
5.20.2.5	object	46
5.20.2.6	port	46
5.20.2.7	protocol	46
5.21	tagDCCConfig Struct Reference	46
5.21.1	Detailed Description	47
5.21.2	Member Function Documentation	47
5.21.2.1	ConfigStrPtr	47
5.21.2.2	ConfigStrPtr	47
5.21.2.3	ConfigStrPtr	47
5.21.2.4	ConfigValuePtr	47
5.21.2.5	ConfigValuePtr	48
5.21.2.6	void	48

5.21.2.7	void	48
5.21.3	Member Data Documentation	48
5.21.3.1	apiVersion	48
5.21.3.2	setting	48
5.21.3.3	type	48
5.21.3.4	val	48
5.22	tagDCConnection Struct Reference	48
5.22.1	Detailed Description	49
5.22.2	Member Function Documentation	49
5.22.2.1	UserDataPtr	49
5.22.2.2	void	49
5.22.2.3	void	49
5.22.2.4	void	49
5.22.3	Member Data Documentation	49
5.22.3.1	apiVersion	49
5.22.3.2	cmd	49
5.22.3.3	data	49
5.22.3.4	graceless	49
5.22.3.5	n	49
5.22.3.6	port	49
5.23	tagDCCore Struct Reference	50
5.23.1	Detailed Description	50
5.23.2	Member Function Documentation	50
5.23.2.1	Bool	50
5.23.2.2	Bool	50
5.23.2.3	DCInterfacePtr	50
5.23.2.4	host_name	50
5.23.2.5	intfHandle	51
5.23.3	Member Data Documentation	51
5.23.3.1	apiVersion	51
5.23.3.2	plInterface	51
5.23.3.3	version	51
5.24	tagDCHooks Struct Reference	51
5.24.1	Detailed Description	52
5.24.2	Member Function Documentation	52
5.24.2.1	Bool	52
5.24.2.2	Bool	52
5.24.2.3	hookHandle	52
5.24.2.4	size_t	52
5.24.2.5	subsHandle	52

5.24.3	Member Data Documentation	52
5.24.3.1	apiVersion	52
5.24.3.2	defProc	52
5.24.3.3	hookProc	52
5.24.3.4	pCommon	52
5.24.3.5	pData	53
5.24.3.6	pObject	53
5.25	tagDCHub Struct Reference	53
5.25.1	Detailed Description	54
5.25.2	Member Function Documentation	54
5.25.2.1	Bool	54
5.25.2.2	HubDataPtr	54
5.25.2.3	HubDataPtr	54
5.25.2.4	HubDataPtr	54
5.25.2.5	UserDataPtr	54
5.25.2.6	UserDataPtr	54
5.25.2.7	void	54
5.25.2.8	void	54
5.25.2.9	void	54
5.25.2.10	void	54
5.25.2.11	void	54
5.25.2.12	void	54
5.25.2.13	void	54
5.25.3	Member Data Documentation	54
5.25.3.1	apiVersion	54
5.25.3.2	cmd	54
5.25.3.3	hubUrl	54
5.25.3.4	msg	54
5.25.3.5	nick	54
5.25.3.6	password	55
5.25.3.7	thirdPerson	55
5.25.3.8	type	55
5.26	tagDCInterface Struct Reference	55
5.26.1	Detailed Description	55
5.26.2	Member Data Documentation	55
5.26.2.1	apiVersion	55
5.27	tagDCLog Struct Reference	55
5.27.1	Detailed Description	56
5.27.2	Member Function Documentation	56
5.27.2.1	void	56

5.27.3	Member Data Documentation	56
5.27.3.1	apiVersion	56
5.28	tagDCQueue Struct Reference	56
5.28.1	Detailed Description	57
5.28.2	Member Function Documentation	57
5.28.2.1	Bool	57
5.28.2.2	QueueDataPtr	57
5.28.2.3	QueueDataPtr	57
5.28.2.4	QueueDataPtr	57
5.28.2.5	QueueDataPtr	57
5.28.2.6	void	58
5.28.2.7	void	58
5.28.2.8	void	58
5.28.3	Member Data Documentation	58
5.28.3.1	apiVersion	58
5.28.3.2	priority	58
5.28.3.3	silent	58
5.28.3.4	size	58
5.28.3.5	target	58
5.29	tagDCTagger Struct Reference	58
5.29.1	Detailed Description	59
5.29.2	Member Function Documentation	59
5.29.2.1	get_text	59
5.29.2.2	void	59
5.29.2.3	void	59
5.29.3	Member Data Documentation	60
5.29.3.1	apiVersion	60
5.29.3.2	attributes	60
5.29.3.3	end	60
5.29.3.4	id	60
5.29.3.5	replacement	60
5.29.3.6	start	60
5.30	tagDCUtils Struct Reference	60
5.30.1	Detailed Description	61
5.30.2	Member Function Documentation	62
5.30.2.1	size_t	62
5.30.2.2	size_t	62
5.30.2.3	size_t	62
5.30.2.4	size_t	62
5.30.2.5	size_t	62

5.30.2.6	size_t	62
5.30.3	Member Data Documentation	62
5.30.3.1	apiVersion	62
5.30.3.2	n	62
5.30.3.3	n	62
5.30.3.4	n	62
5.30.3.5	src	62
5.30.3.6	src	62
5.30.3.7	src	62
5.31	dcapi::Tagger Class Reference	62
5.31.1	Detailed Description	63
5.31.2	Member Function Documentation	63
5.31.2.1	handle	63
5.31.2.2	init	63
5.31.2.3	init	64
5.31.3	Member Data Documentation	64
5.31.3.1	tagger	64
5.32	tagHubData Struct Reference	64
5.32.1	Detailed Description	65
5.32.2	Member Data Documentation	65
5.32.2.1	ip	65
5.32.2.2	isManaged	65
5.32.2.3	isOp	65
5.32.2.4	isSecure	65
5.32.2.5	object	65
5.32.2.6	port	66
5.32.2.7	protocol	66
5.32.2.8	url	66
5.33	tagMetaData Struct Reference	66
5.33.1	Detailed Description	66
5.33.2	Member Data Documentation	67
5.33.2.1	apiVersion	67
5.33.2.2	author	67
5.33.2.3	dependencies	67
5.33.2.4	description	67
5.33.2.5	guid	67
5.33.2.6	name	67
5.33.2.7	numDependencies	67
5.33.2.8	version	67
5.33.2.9	web	67

5.34	tagQueueData Struct Reference	68
5.34.1	Detailed Description	68
5.34.2	Member Data Documentation	68
5.34.2.1	hash	68
5.34.2.2	isFileList	68
5.34.2.3	isManaged	68
5.34.2.4	location	69
5.34.2.5	object	69
5.34.2.6	size	69
5.34.2.7	target	69
5.35	tagStringData Struct Reference	69
5.35.1	Detailed Description	69
5.35.2	Member Data Documentation	69
5.35.2.1	in	69
5.35.2.2	out	70
5.36	tagTagData Struct Reference	70
5.36.1	Detailed Description	70
5.36.2	Member Data Documentation	70
5.36.2.1	isManaged	70
5.36.2.2	object	70
5.37	tagUDPData Struct Reference	70
5.37.1	Detailed Description	71
5.37.2	Member Data Documentation	71
5.37.2.1	ip	71
5.37.2.2	port	71
5.38	tagUserData Struct Reference	71
5.38.1	Detailed Description	72
5.38.2	Member Data Documentation	72
5.38.2.1	cid	72
5.38.2.2	hubHint	72
5.38.2.3	isManaged	72
5.38.2.4	isOp	72
5.38.2.5	nick	72
5.38.2.6	object	73
5.38.2.7	protocol	73
5.38.2.8	sid	73
5.39	dcapi::Hooks::Timer Struct Reference	73
5.39.1	Detailed Description	73
5.39.2	Member Function Documentation	73
5.39.2.1	onMinute	73

5.39.2.2	onSecond	73
5.40	dcapi::Hooks::UI Struct Reference	74
5.40.1	Detailed Description	74
5.40.2	Member Function Documentation	74
5.40.2.1	onChatCommand	74
5.40.2.2	onChatCommandPM	74
5.40.2.3	onChatDisplay	75
5.40.2.4	onChatTags	75
5.40.2.5	onCreated	75
5.41	dcapi::UI Class Reference	76
5.41.1	Detailed Description	76
5.41.2	Member Typedef Documentation	76
5.41.2.1	Command	76
5.41.3	Member Function Documentation	77
5.41.3.1	addCommand	77
5.41.3.2	clearCommands	77
5.41.3.3	commandCallback	77
5.41.3.4	handle	77
5.41.3.5	init	77
5.41.3.6	init	78
5.41.3.7	removeCommand	78
5.41.4	Member Data Documentation	78
5.41.4.1	commands	78
5.41.4.2	guid	78
5.41.4.3	ui	78
5.42	dcapi::Hooks::Users Struct Reference	78
5.42.1	Detailed Description	79
5.42.2	Member Function Documentation	79
5.42.2.1	onOffline	79
5.42.2.2	onOnline	79
5.43	dcapi::Util Class Reference	79
5.43.1	Detailed Description	80
5.43.2	Member Function Documentation	80
5.43.2.1	fromUtf8	80
5.43.2.2	handle	80
5.43.2.3	init	81
5.43.2.4	init	81
5.43.2.5	toUtf8	81
5.43.3	Member Data Documentation	81
5.43.3.1	utils	81

6	File Documentation	83
6.1	Config.c File Reference	83
6.1.1	Function Documentation	84
6.1.1.1	free_core_cfg	84
6.1.1.2	get_cfg	84
6.1.1.3	get_cfg_int	84
6.1.1.4	get_cfg_int64	84
6.1.1.5	get_cfg_path	84
6.1.1.6	get_core_cfg	84
6.1.1.7	get_install_path	84
6.1.1.8	init_cfg	84
6.1.1.9	set_cfg	85
6.1.1.10	set_cfg_int	85
6.1.1.11	set_cfg_int64	85
6.1.1.12	uninit_cfg	85
6.1.2	Variable Documentation	85
6.1.2.1	config	85
6.1.2.2	guid	85
6.2	Config.cpp File Reference	85
6.3	Config.h File Reference	86
6.3.1	Function Documentation	87
6.3.1.1	free_core_cfg	87
6.3.1.2	get_cfg	87
6.3.1.3	get_cfg_int	87
6.3.1.4	get_cfg_int64	87
6.3.1.5	get_cfg_path	88
6.3.1.6	get_core_cfg	88
6.3.1.7	get_install_path	88
6.3.1.8	init_cfg	88
6.3.1.9	set_cfg	88
6.3.1.10	set_cfg_int	88
6.3.1.11	set_cfg_int64	88
6.3.1.12	uninit_cfg	88
6.4	Config.hpp File Reference	88
6.5	Connections.cpp File Reference	89
6.6	Connections.hpp File Reference	90
6.7	Core.cpp File Reference	90
6.8	Core.hpp File Reference	91
6.9	Hooks.cpp File Reference	92
6.10	Hooks.hpp File Reference	92

6.11 Hubs.cpp File Reference	93
6.12 Hubs.hpp File Reference	94
6.13 Logger.cpp File Reference	94
6.14 Logger.hpp File Reference	95
6.15 main.cpp File Reference	95
6.15.1 Function Documentation	96
6.15.1.1 pluginInit	96
6.16 Plugin.cpp File Reference	96
6.17 Plugin.h File Reference	97
6.18 PluginDefs.h File Reference	97
6.18.1 Macro Definition Documentation	101
6.18.1.1 Bool	101
6.18.1.2 DCAPI	101
6.18.1.3 DCAPI_CORE_VER	101
6.18.1.4 DCEXP	101
6.18.1.5 DCIMP	101
6.18.1.6 DCINTF_CONFIG	101
6.18.1.7 DCINTF_CONFIG_VER	101
6.18.1.8 DCINTF_DCPP_CONNECTIONS	102
6.18.1.9 DCINTF_DCPP_CONNECTIONS_VER	102
6.18.1.10 DCINTF_DCPP_HUBS	102
6.18.1.11 DCINTF_DCPP_HUBS_VER	102
6.18.1.12 DCINTF_DCPP_QUEUE	102
6.18.1.13 DCINTF_DCPP_QUEUE_VER	102
6.18.1.14 DCINTF_DCPP_TAGGER	102
6.18.1.15 DCINTF_DCPP_TAGGER_VER	102
6.18.1.16 DCINTF_DCPP_UI	102
6.18.1.17 DCINTF_DCPP_UI_VER	103
6.18.1.18 DCINTF_DCPP_UTILS	103
6.18.1.19 DCINTF_DCPP_UTILS_VER	103
6.18.1.20 DCINTF_HOOKS	103
6.18.1.21 DCINTF_HOOKS_VER	103
6.18.1.22 DCINTF_LOGGING	103
6.18.1.23 DCINTF_LOGGING_VER	103
6.18.1.24 False	103
6.18.1.25 HOOK_CHAT_IN	103
6.18.1.26 HOOK_CHAT_OUT	104
6.18.1.27 HOOK_CHAT_PM_IN	104
6.18.1.28 HOOK_CHAT_PM_OUT	104
6.18.1.29 HOOK_HUB_OFFLINE	104

6.18.1.30	HOOK_HUB_ONLINE	104
6.18.1.31	HOOK_NETWORK_CONN_IN	104
6.18.1.32	HOOK_NETWORK_CONN_OUT	104
6.18.1.33	HOOK_NETWORK_HUB_IN	104
6.18.1.34	HOOK_NETWORK_HUB_OUT	104
6.18.1.35	HOOK_NETWORK_UDP_IN	105
6.18.1.36	HOOK_NETWORK_UDP_OUT	105
6.18.1.37	HOOK_QUEUE_ADDED	105
6.18.1.38	HOOK_QUEUE_FINISHED	105
6.18.1.39	HOOK_QUEUE_MOVED	105
6.18.1.40	HOOK_QUEUE_REMOVED	105
6.18.1.41	HOOK_TIMER_MINUTE	105
6.18.1.42	HOOK_TIMER_SECOND	105
6.18.1.43	HOOK_UI_CHAT_COMMAND	105
6.18.1.44	HOOK_UI_CHAT_COMMAND_PM	106
6.18.1.45	HOOK_UI_CHAT_DISPLAY	106
6.18.1.46	HOOK_UI_CHAT_TAGS	106
6.18.1.47	HOOK_UI_CREATED	106
6.18.1.48	HOOK_USER_OFFLINE	106
6.18.1.49	HOOK_USER_ONLINE	106
6.18.1.50	True	106
6.18.2	Typedef Documentation	106
6.18.2.1	bBreak	106
6.18.2.2	CommandData	107
6.18.2.3	CommandDataPtr	107
6.18.2.4	ConfigBool	107
6.18.2.5	ConfigBoolPtr	107
6.18.2.6	ConfigInt	107
6.18.2.7	ConfigInt64	107
6.18.2.8	ConfigInt64Ptr	107
6.18.2.9	ConfigIntPtr	107
6.18.2.10	ConfigStr	107
6.18.2.11	ConfigStrPtr	107
6.18.2.12	ConfigType	107
6.18.2.13	ConfigValue	107
6.18.2.14	ConfigValuePtr	107
6.18.2.15	ConnectionData	107
6.18.2.16	ConnectionDataPtr	107
6.18.2.17	core	107
6.18.2.18	dcBool	107

6.18.2.19 DCConfig	107
6.18.2.20 DCConfigPtr	107
6.18.2.21 DCConnection	107
6.18.2.22 DCConnectionPtr	107
6.18.2.23 DCCore	107
6.18.2.24 DCCorePtr	107
6.18.2.25 DCHooks	107
6.18.2.26 DCHooksPtr	107
6.18.2.27 DCHub	107
6.18.2.28 DCHubPtr	107
6.18.2.29 DCInterface	108
6.18.2.30 DCInterfacePtr	108
6.18.2.31 DCLog	108
6.18.2.32 DCLogPtr	108
6.18.2.33 dcptr_t	108
6.18.2.34 DCQueue	108
6.18.2.35 DCQueuePtr	108
6.18.2.36 DCTagger	108
6.18.2.37 DCTaggerPtr	108
6.18.2.38 dctime_t	108
6.18.2.39 DCUI	108
6.18.2.40 DCUIPtr	108
6.18.2.41 DCUtils	108
6.18.2.42 DCUtilsPtr	108
6.18.2.43 hookHandle	108
6.18.2.44 HubData	108
6.18.2.45 HubDataPtr	108
6.18.2.46 intfHandle	108
6.18.2.47 MetaData	108
6.18.2.48 MetaDataPtr	108
6.18.2.49 MsgType	108
6.18.2.50 PathType	108
6.18.2.51 pCommon	109
6.18.2.52 pData	109
6.18.2.53 PluginState	109
6.18.2.54 ProtocolType	109
6.18.2.55 QueueData	109
6.18.2.56 QueueDataPtr	109
6.18.2.57 QueuePrio	109
6.18.2.58 StringData	109

6.18.2.59 StringDataPtr	109
6.18.2.60 subsHandle	109
6.18.2.61 TagData	109
6.18.2.62 TagDataPtr	109
6.18.2.63 UDPData	109
6.18.2.64 UDPDataPtr	109
6.18.2.65 UserData	109
6.18.2.66 UserDataPtr	109
6.18.3 Enumeration Type Documentation	109
6.18.3.1 tagConfigType	109
6.18.3.2 tagDCBool	110
6.18.3.3 tagMsgType	110
6.18.3.4 tagPathType	110
6.18.3.5 tagPluginState	110
6.18.3.6 tagProtocolType	110
6.18.3.7 tagQueuePrio	111
6.18.4 Function Documentation	111
6.18.4.1 Bool	111
6.18.4.2 void	111
6.19 Queue.cpp File Reference	111
6.20 Queue.hpp File Reference	112
6.21 resource.h File Reference	112
6.22 stdafx.cpp File Reference	112
6.23 stdafx.h File Reference	113
6.23.1 Macro Definition Documentation	113
6.23.1.1 PLUGIN_VERSION	113
6.23.2 Typedef Documentation	114
6.23.2.1 tstring	114
6.24 Tagger.cpp File Reference	114
6.25 Tagger.hpp File Reference	115
6.26 UI.cpp File Reference	115
6.27 UI.hpp File Reference	116
6.28 Util.cpp File Reference	116
6.29 Util.hpp File Reference	117
6.30 version.h File Reference	118
6.30.1 Macro Definition Documentation	118
6.30.1.1 PLUGIN_AUTHOR	118
6.30.1.2 PLUGIN_DESC	118
6.30.1.3 PLUGIN_GUID	119
6.30.1.4 PLUGIN_NAME	119

6.30.1.5	PLUGIN_VERSION_MAJOR	119
6.30.1.6	PLUGIN_VERSION_MINOR	119
6.30.1.7	PLUGIN_WEB	119

Index	120
--------------	------------

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

dcapi	7
---------------------------------	---

Chapter 2

Class Index

2.1 Class List

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

dcapi::Hooks::Chat	9
dcapi::Config	10
dcapi::Connections	15
dcapi::Core	17
DCUI	19
dcapi::Hooks	21
dcapi::Hooks::Hubs	27
dcapi::Hubs	28
dcapi::Logger	30
dcapi::Hooks::Network	32
Plugin	34
dcapi::Hooks::Queue	36
dcapi::Queue	38
tagCommandData	40
tagConfigBool	41
tagConfigInt	41
tagConfigInt64	42
tagConfigStr	43
tagConfigValue	44
tagConnectionData	45
tagDCConfig	46
tagDCConnection	48
tagDCCore	50
tagDCHooks	51
tagDCHub	53
tagDCInterface	55
tagDCLog	55
tagDCQueue	56
tagDCTagger	58
tagDCUtils	60
dcapi::Tagger	62
tagHubData	64
tagMetaData	66
tagQueueData	68
tagStringData	69
tagTagData	70
tagUDPData	70
tagUserData	71

dcapi::Hooks::Timer	73
dcapi::Hooks::UI	74
dcapi::UI	76
dcapi::Hooks::Users	78
dcapi::Util	79

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Config.c	83
Config.cpp	85
Config.h	86
Config.hpp	88
Connections.cpp	89
Connections.hpp	90
Core.cpp	90
Core.hpp	91
Hooks.cpp	92
Hooks.hpp	92
Hubs.cpp	93
Hubs.hpp	94
Logger.cpp	94
Logger.hpp	95
main.cpp	95
Plugin.cpp	96
Plugin.h	97
PluginDefs.h	97
Queue.cpp	111
Queue.hpp	112
resource.h	112
stdafx.cpp	112
stdafx.h	113
Tagger.cpp	114
Tagger.hpp	115
UI.cpp	115
UI.hpp	116
Util.cpp	116
Util.hpp	117
version.h	118

Chapter 4

Namespace Documentation

4.1 dcapi Namespace Reference

Classes

- class [Config](#)
- class [Connections](#)
- class [Core](#)
- class [Hooks](#)
- class [Hubs](#)
- class [Logger](#)
- class [Queue](#)
- class [Tagger](#)
- class [UI](#)
- class [Util](#)

Chapter 5

Class Documentation

5.1 dcapi::Hooks::Chat Struct Reference

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

Static Public Member Functions

- static [void onIncomingChat](#) (function< bool([HubDataPtr](#), char *, bool &)> f)
- static [void onIncomingPM](#) (function< bool([UserDataPtr](#), char *, bool &)> f)
- static [void onOutgoingChat](#) (function< bool([HubDataPtr](#), char *, bool &)> f)
- static [void onOutgoingPM](#) (function< bool([UserDataPtr](#), char *, bool &)> f)

5.1.1 Detailed Description

Definition at line 54 of file Hooks.hpp.

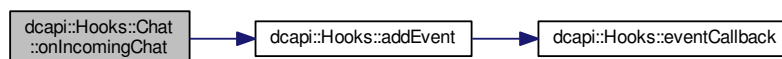
5.1.2 Member Function Documentation

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

Definition at line 38 of file Hooks.cpp.

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

Here is the call graph for this function:



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

Definition at line 46 of file Hooks.cpp.

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

Here is the call graph for this function:

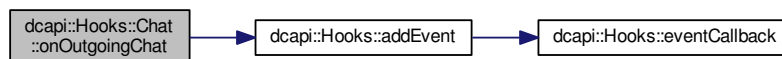


5.1.2.3 void dcapi::Hooks::Chat::onOutgoingChat (function< bool(HubDataPtr, char *, bool &)> f) [static]

Definition at line 42 of file Hooks.cpp.

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

Here is the call graph for this function:

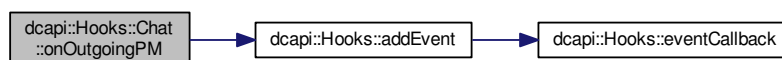


5.1.2.4 void dcapi::Hooks::Chat::onOutgoingPM (function< bool(UserDataPtr, char *, bool &)> f) [static]

Definition at line 50 of file Hooks.cpp.

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

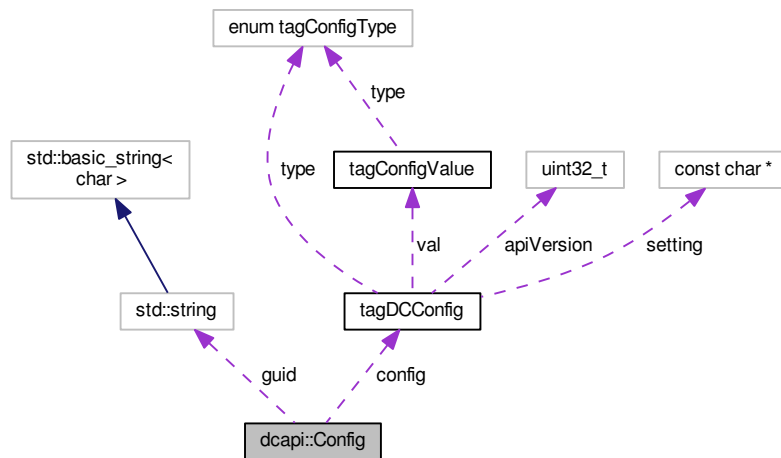
Here is the call graph for this function:



5.2 dcapi::Config Class Reference

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

Collaboration diagram for dcapi::Config:



Public Member Functions

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

Static Public Member Functions

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

Static Private Member Functions

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

Static Private Attributes

- static [DCConfigPtr config](#)
- static string [guid](#)

5.2.1 Detailed Description

Definition at line 32 of file Config.hpp.

5.2.2 Member Function Documentation

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

Definition at line 50 of file Config.cpp.

References [config](#).

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

Definition at line 45 of file Config.cpp.

References [CFG_TYPE_BOOL](#).

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

Definition at line 44 of file Config.cpp.

References [CFG_TYPE_STRING](#).

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

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

Definition at line 70 of file Config.hpp.

References [config](#), and [guid](#).

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

Definition at line 49 of file Config.cpp.

References [CFG_TYPE_UNKNOWN](#), and [config](#).

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

Definition at line 59 of file Config.cpp.

References [config](#), and [guid](#).

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

Definition at line 47 of file Config.cpp.

References [CFG_TYPE_INT64](#).

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

Definition at line 46 of file Config.cpp.

References CFG_TYPE_INT.

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

Definition at line 52 of file Config.cpp.

References config.

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

Definition at line 36 of file Config.cpp.

References config.

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

Definition at line 30 of file Config.cpp.

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

Here is the call graph for this function:

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

Definition at line 35 of file Config.cpp.

References config, and guid.

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

Definition at line 38 of file Config.cpp.

References CFG_TYPE_STRING.

Referenced by setConfig().

Here is the caller graph for this function:



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

Definition at line 39 of file Config.cpp.

References `setConfig()`.

Here is the call graph for this function:



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

Definition at line 40 of file Config.cpp.

References `CFG_TYPE_BOOL`, `False`, and `True`.

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

Definition at line 41 of file Config.cpp.

References `CFG_TYPE_INT`.

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

Definition at line 42 of file Config.cpp.

References `CFG_TYPE_INT64`.

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

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

Definition at line 65 of file Config.hpp.

References config, and guid.

5.2.3 Member Data Documentation

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

Definition at line 60 of file Config.hpp.

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

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

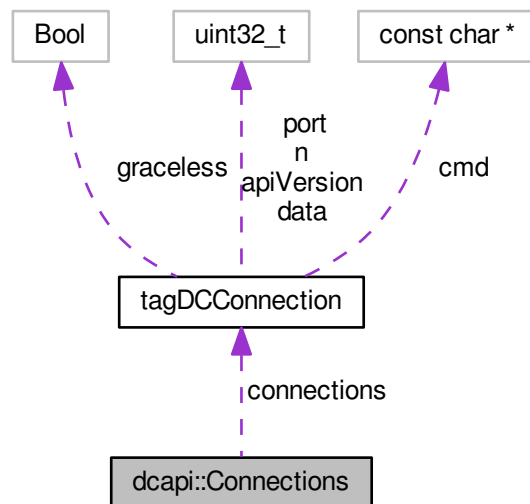
Definition at line 62 of file Config.hpp.

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

5.3 dcapi::Connections Class Reference

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

Collaboration diagram for dcapi::Connections:



Static Public Member Functions

- static [DCConnectionPtr handle](#) ()
- static bool [init](#) ()
- static void [init](#) ([DCConnectionPtr](#) coreConnections)

Static Private Attributes

- static [DCConnectionPtr connections](#)

5.3.1 Detailed Description

Definition at line 28 of file Connections.hpp.

5.3.2 Member Function Documentation

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

Definition at line 35 of file Connections.cpp.

References [connections](#).

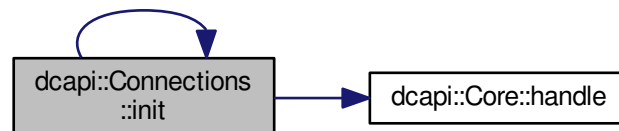
5.3.2.2 `bool dcapi::Connections::init () [static]`

Definition at line 29 of file Connections.cpp.

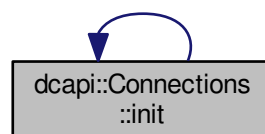
References [connections](#), `DCINTF_DCPP_CONNECTIONS`, `DCINTF_DCPP_CONNECTIONS_VER`, [dcapi::Core::handle\(\)](#), and [init\(\)](#).

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

Here is the call graph for this function:



Here is the caller graph for this function:



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

Definition at line 34 of file Connections.cpp.

References connections.

5.3.3 Member Data Documentation

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

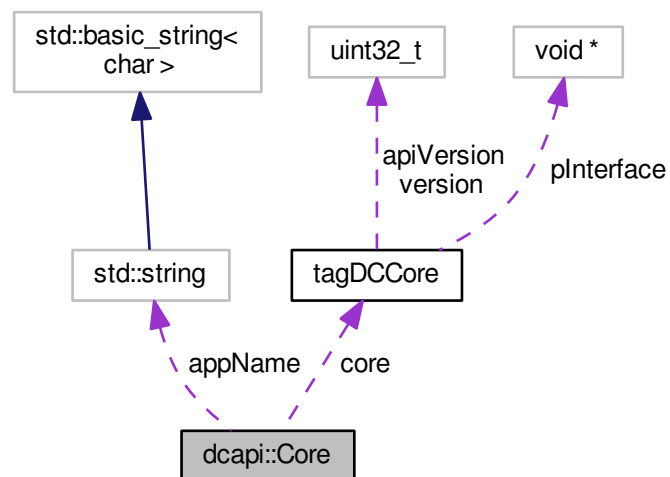
Definition at line 36 of file Connections.hpp.

Referenced by handle(), and init().

5.4 dcapi::Core Class Reference

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

Collaboration diagram for dcapi::Core:



Static Public Member Functions

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

Static Public Attributes

- static string `appName`

Static Private Attributes

- static `DCCorePtr core`

5.4.1 Detailed Description

Definition at line 32 of file Core.hpp.

5.4.2 Member Function Documentation

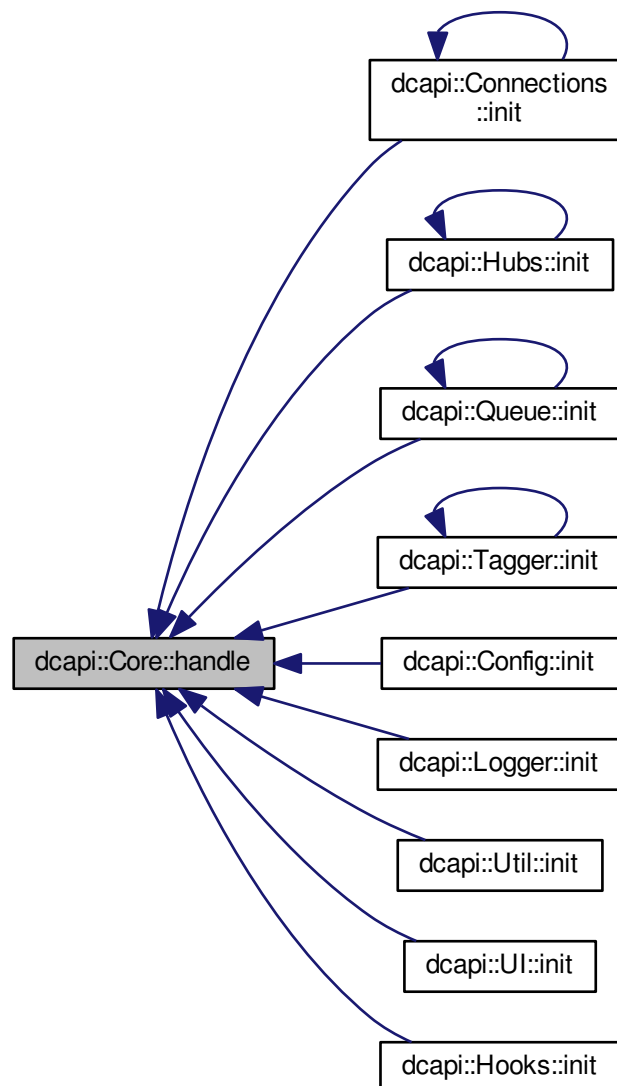
5.4.2.1 `DCCorePtr dcapi::Core::handle () [static]`

Definition at line 33 of file Core.cpp.

References core.

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

Here is the caller graph for this function:



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

Definition at line 28 of file Core.cpp.

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

Here is the call graph for this function:



5.4.3 Member Data Documentation

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

Definition at line 38 of file Core.hpp.

Referenced by `init()`.

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

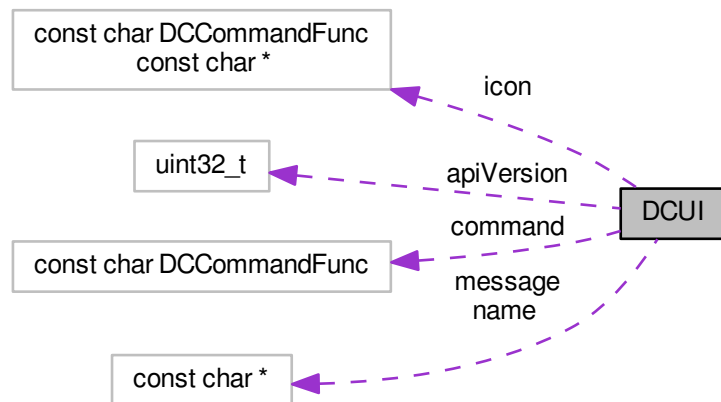
Definition at line 41 of file Core.hpp.

Referenced by `handle()`.

5.5 DCUI Struct Reference

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

Collaboration diagram for DCUI:



Public Member Functions

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

Public Attributes

- `uint32_t apiVersion`
- `const char DCCommandFunc command`
- `const char DCCommandFunc const char * icon`
- `const char * message`
- `const char * name`

5.5.1 Detailed Description

Definition at line 448 of file PluginDefs.h.

5.5.2 Member Function Documentation

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

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

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

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

5.5.3 Member Data Documentation

5.5.3.1 uint32_t DCUI::apiVersion

Definition at line 450 of file PluginDefs.h.

5.5.3.2 const char DCCommandFunc DCUI::command

Definition at line 454 of file PluginDefs.h.

5.5.3.3 const char DCCommandFunc const char* DCUI::icon

Definition at line 454 of file PluginDefs.h.

5.5.3.4 const char* DCUI::message

Definition at line 459 of file PluginDefs.h.

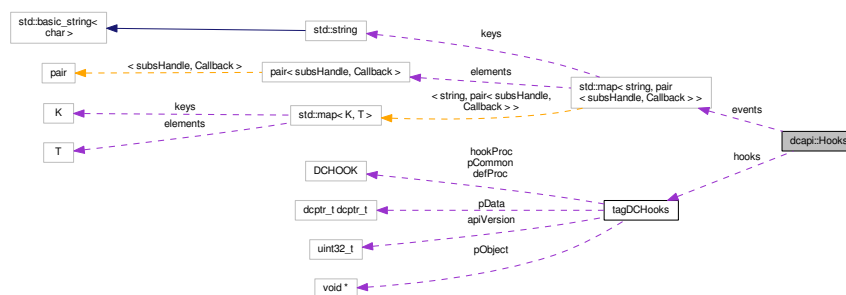
5.5.3.5 const char * DCUI::name

Definition at line 454 of file PluginDefs.h.

5.6 dcapi::Hooks Class Reference

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

Collaboration diagram for dcapi::Hooks:



Classes

- struct [Chat](#)
- struct [Hubs](#)
- struct [Network](#)
- struct [Queue](#)
- struct [Timer](#)
- struct [UI](#)
- struct [Users](#)

Static Public Member Functions

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

Private Types

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

Static Private Member Functions

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

Static Private Attributes

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

5.6.1 Detailed Description

Definition at line 38 of file Hooks.hpp.

5.6.2 Member Typedef Documentation

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

Definition at line 105 of file Hooks.hpp.

5.6.3 Member Function Documentation

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

Definition at line 163 of file Hooks.cpp.

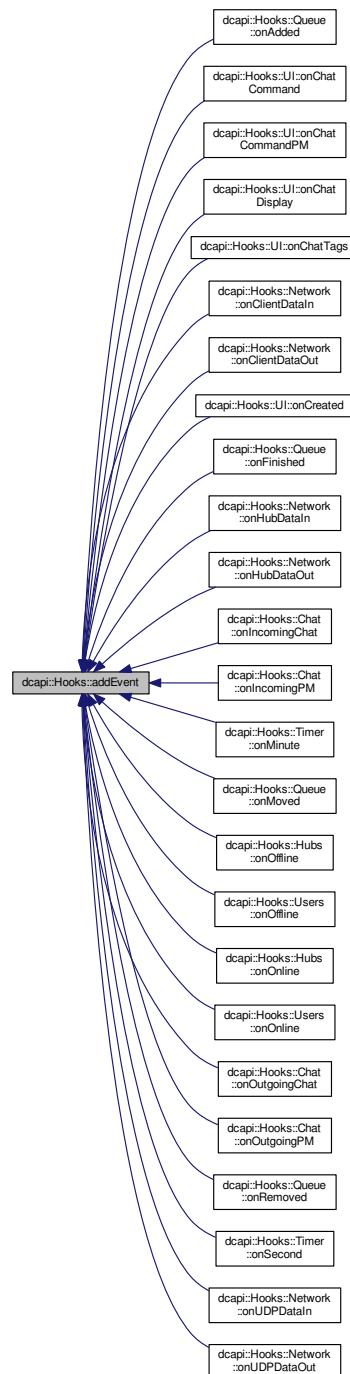
References `eventCallback()`, `events`, and `hooks`.

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

Here is the call graph for this function:



Here is the caller graph for this function:



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

Definition at line 149 of file Hooks.cpp.

References events, and hooks.

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

Definition at line 145 of file Hooks.cpp.

References events.

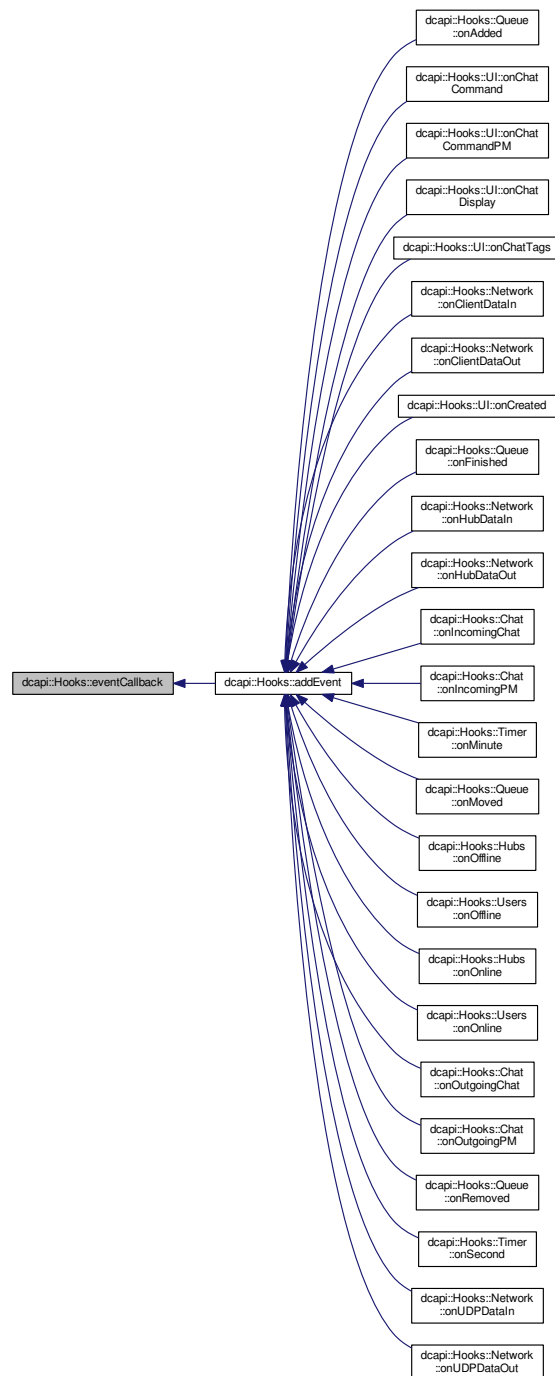
5.6.3.4 `Bool DCAPI dcapi::Hooks::eventCallback (dcptr_t pObject, dcptr_t pData, dcptr_t pCommon, Bool * bBreak) [static], [private]`

Definition at line 169 of file Hooks.cpp.

References dcTrue, events, False, and True.

Referenced by addEvent().

Here is the caller graph for this function:



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

Definition at line 36 of file Hooks.cpp.

References hooks.

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

Definition at line 30 of file Hooks.cpp.

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

Here is the call graph for this function:



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

Definition at line 35 of file Hooks.cpp.

References hooks.

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

Definition at line 155 of file Hooks.cpp.

References events, and hooks.

5.6.4 Member Data Documentation

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

Definition at line 111 of file Hooks.hpp.

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

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

Definition at line 109 of file Hooks.hpp.

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

5.7 dcapi::Hooks::Hubs Struct Reference

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

Static Public Member Functions

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

5.7.1 Detailed Description

Definition at line 66 of file Hooks.hpp.

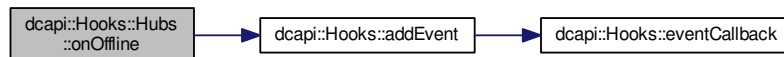
5.7.2 Member Function Documentation

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

Definition at line 68 of file Hooks.cpp.

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

Here is the call graph for this function:

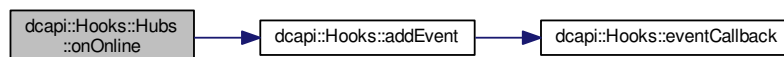


5.7.2.2 `void dcapi::Hooks::Hubs::onOnline (function< bool(HubDataPtr, bool &)> f) [static]`

Definition at line 64 of file Hooks.cpp.

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

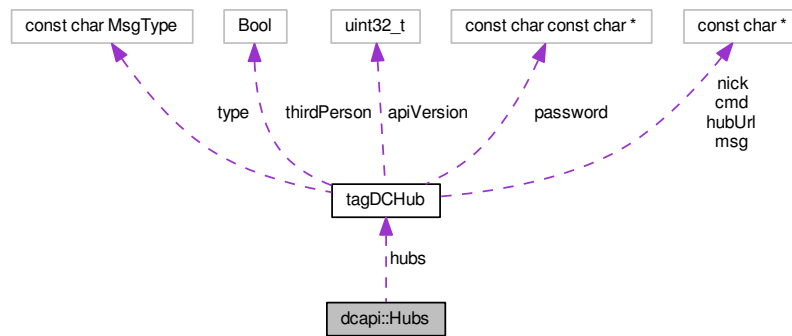
Here is the call graph for this function:



5.8 dcapi::Hubs Class Reference

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


Collaboration diagram for dcapi::Hubs:



Static Public Member Functions

- static [DCHubPtr handle](#) ()
- static bool [init](#) ()
- static void [init](#) ([DCHubPtr](#) coreHubs)

Static Private Attributes

- static [DCHubPtr hubs](#)

5.8.1 Detailed Description

Definition at line 28 of file Hubs.hpp.

5.8.2 Member Function Documentation

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

Definition at line 35 of file Hubs.cpp.

References [hubs](#).

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

Definition at line 29 of file Hubs.cpp.

References [DCINTF_DCAPP_HUBS](#), [DCINTF_DCAPP_HUBS_VER](#), [dcapi::Core::handle\(\)](#), [hubs](#), and [init\(\)](#).

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

Here is the call graph for this function:



Here is the caller graph for this function:



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

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

References `hubs`.

5.8.3 Member Data Documentation

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

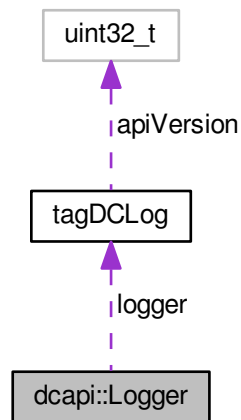
Definition at line 36 of file `Hubs.hpp`.

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

5.9 dcapi::Logger Class Reference

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

Collaboration diagram for dcapi::Logger:



Static Public Member Functions

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

Static Private Attributes

- static `DCLogPtr logger`

5.9.1 Detailed Description

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

5.9.2 Member Function Documentation

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

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

References `logger`.

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

Definition at line 29 of file `Logger.cpp`.

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

Here is the call graph for this function:



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

Definition at line 34 of file Logger.cpp.

References logger.

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

Definition at line 37 of file Logger.cpp.

References logger.

5.9.3 Member Data Documentation

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

Definition at line 42 of file Logger.hpp.

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

5.10 dcapi::Hooks::Network Struct Reference

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

Static Public Member Functions

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

5.10.1 Detailed Description

Definition at line 76 of file Hooks.hpp.

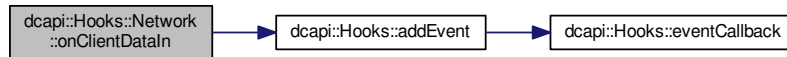
5.10.2 Member Function Documentation

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

Definition at line 90 of file Hooks.cpp.

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

Here is the call graph for this function:

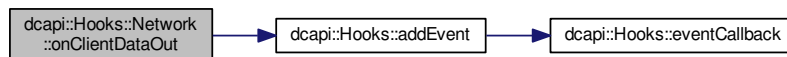


5.10.2.2 `void dcapi::Hooks::Network::onClientDataOut (function< bool(ConnectionDataPtr, char *, bool &)> f)` [static]

Definition at line 94 of file Hooks.cpp.

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

Here is the call graph for this function:

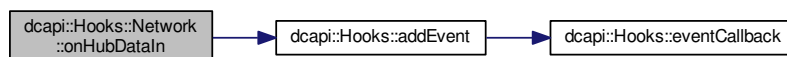


5.10.2.3 `void dcapi::Hooks::Network::onHubDataIn (function< bool(HubDataPtr, char *, bool &)> f)` [static]

Definition at line 82 of file Hooks.cpp.

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

Here is the call graph for this function:



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

Definition at line 86 of file Hooks.cpp.

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

Here is the call graph for this function:

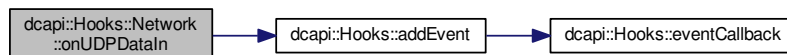


5.10.2.5 void `dcapi::Hooks::Network::onUDPDataIn` (function< bool(`UDPDataPtr`, char *, bool &)> *f*) [static]

Definition at line 98 of file `Hooks.cpp`.

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

Here is the call graph for this function:

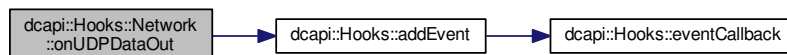


5.10.2.6 void `dcapi::Hooks::Network::onUDPDataOut` (function< bool(`UDPDataPtr`, char *, bool &)> *f*) [static]

Definition at line 102 of file `Hooks.cpp`.

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

Here is the call graph for this function:



5.11 Plugin Class Reference

```
#include <Plugin.h>
```

Static Public Member Functions

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

Private Member Functions

- bool [onLoad](#) ([DCCorePtr core](#), bool install, bool runtime)
- [Plugin](#) ()
- [~Plugin](#) ()

5.11.1 Detailed Description

Definition at line 24 of file `Plugin.h`.

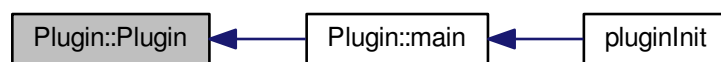
5.11.2 Constructor & Destructor Documentation

5.11.2.1 `Plugin::Plugin ()` `[private]`

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

Referenced by `main()`.

Here is the caller graph for this function:



5.11.2.2 `Plugin::~~Plugin ()` `[private]`

Definition at line 40 of file `Plugin.cpp`.

5.11.3 Member Function Documentation

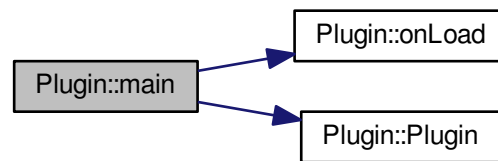
5.11.3.1 `Bool DCAPI Plugin::main (PluginState state, DCCorePtr core, dcptr_t)` `[static]`

Definition at line 44 of file `Plugin.cpp`.

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

Referenced by `pluginInit()`.

Here is the call graph for this function:



Here is the caller graph for this function:



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

Definition at line 71 of file `Plugin.cpp`.

References `PLUGIN_GUID`.

Referenced by `main()`.

Here is the caller graph for this function:



5.12 `dcapi::Hooks::Queue` Struct Reference

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

Static Public Member Functions

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

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

5.12.1 Detailed Description

Definition at line 85 of file Hooks.hpp.

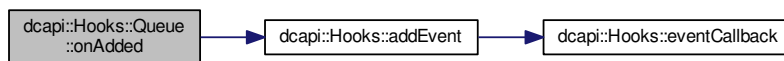
5.12.2 Member Function Documentation

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

Definition at line 107 of file Hooks.cpp.

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

Here is the call graph for this function:

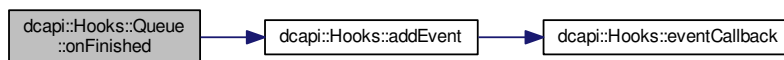


5.12.2.2 void dcapi::Hooks::Queue::onFinished (function< bool([QueueDataPtr](#), bool &)> f) [static]

Definition at line 119 of file Hooks.cpp.

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

Here is the call graph for this function:

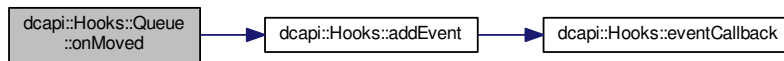


5.12.2.3 void dcapi::Hooks::Queue::onMoved (function< bool([QueueDataPtr](#), bool &)> f) [static]

Definition at line 111 of file Hooks.cpp.

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

Here is the call graph for this function:

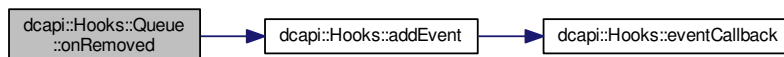


5.12.2.4 void dcapi::Hooks::Queue::onRemoved (function< bool(QueueDataPtr, bool &)> f) [static]

Definition at line 115 of file Hooks.cpp.

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

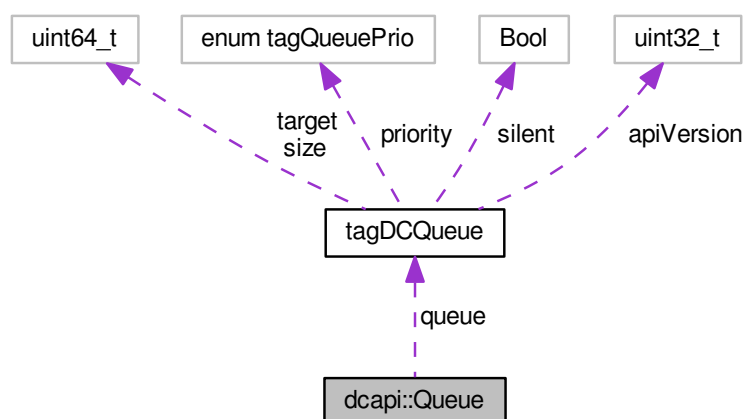
Here is the call graph for this function:



5.13 dcapi::Queue Class Reference

```
#include <Queue.hpp>
```

Collaboration diagram for `dcapi::Queue`:



Static Public Member Functions

- static [DCQueuePtr handle](#) ()

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

Static Private Attributes

- static `DCQueuePtr` `queue`

5.13.1 Detailed Description

Definition at line 28 of file Queue.hpp.

5.13.2 Member Function Documentation

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

Definition at line 35 of file Queue.cpp.

References `queue`.

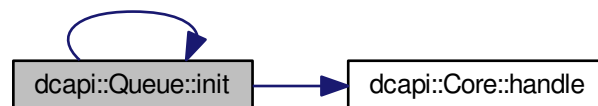
5.13.2.2 `bool` `dcapi::Queue::init` () [static]

Definition at line 29 of file Queue.cpp.

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

Referenced by `init()`.

Here is the call graph for this function:



Here is the caller graph for this function:



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

Definition at line 34 of file Queue.cpp.

References queue.

5.13.3 Member Data Documentation

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

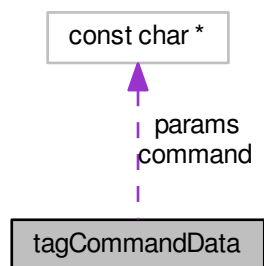
Definition at line 36 of file Queue.hpp.

Referenced by handle(), and init().

5.14 tagCommandData Struct Reference

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

Collaboration diagram for tagCommandData:



Public Attributes

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

5.14.1 Detailed Description

Definition at line 212 of file PluginDefs.h.

5.14.2 Member Data Documentation

5.14.2.1 const char* tagCommandData::command

Definition at line 213 of file PluginDefs.h.

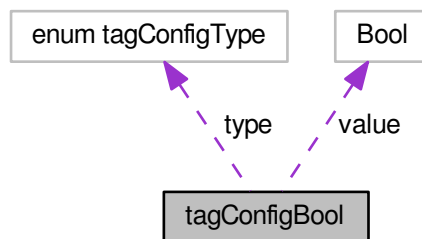
5.14.2.2 `const char* tagCommandData::params`

Definition at line 214 of file PluginDefs.h.

5.15 tagConfigBool Struct Reference

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

Collaboration diagram for tagConfigBool:



Public Attributes

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

5.15.1 Detailed Description

Definition at line 194 of file PluginDefs.h.

5.15.2 Member Data Documentation

5.15.2.1 `ConfigType tagConfigBool::type`

Definition at line 195 of file PluginDefs.h.

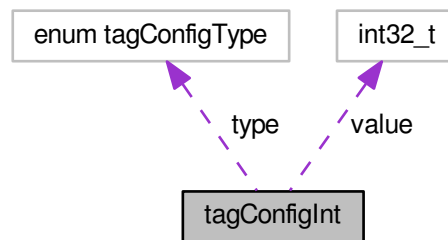
5.15.2.2 `Bool tagConfigBool::value`

Definition at line 196 of file PluginDefs.h.

5.16 tagConfigInt Struct Reference

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

Collaboration diagram for tagConfigInt:



Public Attributes

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

5.16.1 Detailed Description

Definition at line 188 of file `PluginDefs.h`.

5.16.2 Member Data Documentation

5.16.2.1 ConfigType tagConfigInt::type

Definition at line 189 of file `PluginDefs.h`.

Referenced by `set_cfg_int()`.

5.16.2.2 int32_t tagConfigInt::value

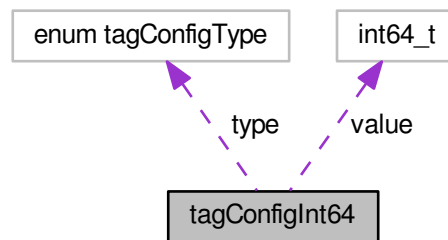
Definition at line 190 of file `PluginDefs.h`.

Referenced by `get_cfg_int()`, and `set_cfg_int()`.

5.17 tagConfigInt64 Struct Reference

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

Collaboration diagram for tagConfigInt64:



Public Attributes

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

5.17.1 Detailed Description

Definition at line 200 of file `PluginDefs.h`.

5.17.2 Member Data Documentation

5.17.2.1 `ConfigType` `tagConfigInt64::type`

Definition at line 201 of file `PluginDefs.h`.

Referenced by `set_cfg_int64()`.

5.17.2.2 `int64_t` `tagConfigInt64::value`

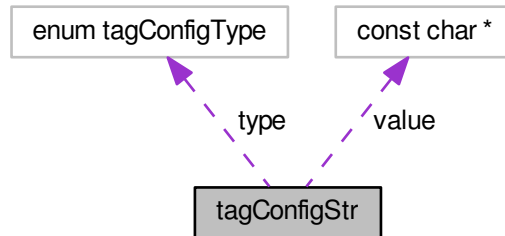
Definition at line 202 of file `PluginDefs.h`.

Referenced by `get_cfg_int64()`, and `set_cfg_int64()`.

5.18 tagConfigStr Struct Reference

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

Collaboration diagram for tagConfigStr:



Public Attributes

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

5.18.1 Detailed Description

Definition at line 182 of file `PluginDefs.h`.

5.18.2 Member Data Documentation

5.18.2.1 `ConfigType` `tagConfigStr::type`

Definition at line 183 of file `PluginDefs.h`.

Referenced by `set_cfg()`.

5.18.2.2 `const char*` `tagConfigStr::value`

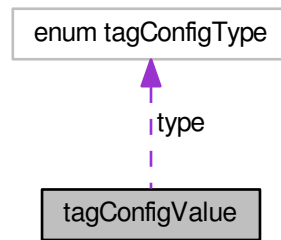
Definition at line 184 of file `PluginDefs.h`.

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

5.19 tagConfigValue Struct Reference

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


Collaboration diagram for tagConfigValue:



Public Attributes

- [ConfigType type](#)

5.19.1 Detailed Description

Definition at line 177 of file `PluginDefs.h`.

5.19.2 Member Data Documentation

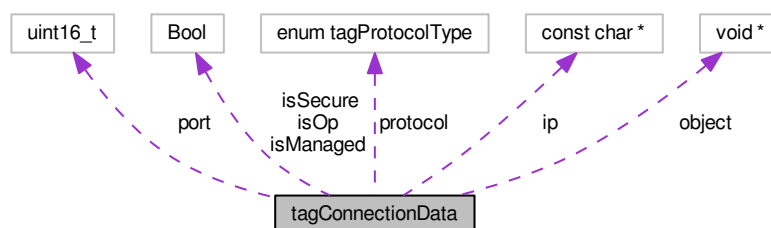
5.19.2.1 ConfigType tagConfigValue::type

Definition at line 178 of file `PluginDefs.h`.

5.20 tagConnectionData Struct Reference

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

Collaboration diagram for tagConnectionData:



Public Attributes

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

5.20.1 Detailed Description

Definition at line 242 of file PluginDefs.h.

5.20.2 Member Data Documentation

5.20.2.1 `const char* tagConnectionData::ip`

Definition at line 243 of file PluginDefs.h.

5.20.2.2 `Bool tagConnectionData::isManaged`

Definition at line 249 of file PluginDefs.h.

5.20.2.3 `Bool tagConnectionData::isOp`

Definition at line 247 of file PluginDefs.h.

5.20.2.4 `Bool tagConnectionData::isSecure`

Definition at line 248 of file PluginDefs.h.

5.20.2.5 `dcptr_t tagConnectionData::object`

Definition at line 244 of file PluginDefs.h.

5.20.2.6 `uint16_t tagConnectionData::port`

Definition at line 245 of file PluginDefs.h.

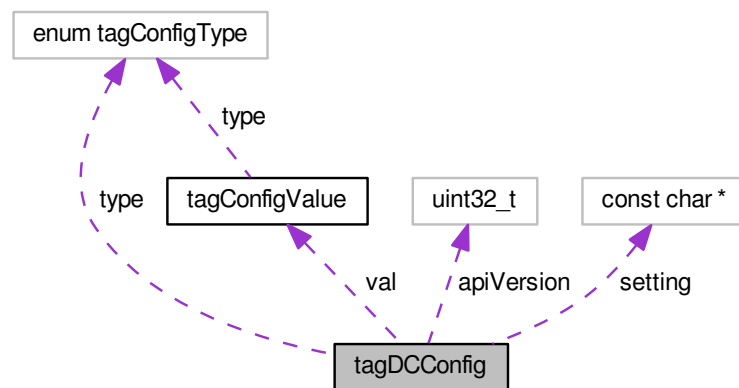
5.20.2.7 `ProtocolType tagConnectionData::protocol`

Definition at line 246 of file PluginDefs.h.

5.21 tagDCConfig Struct Reference

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

Collaboration diagram for tagDCConfig:



Public Member Functions

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

Public Attributes

- [uint32_t](#) [apiVersion](#)
- [const char](#) * [setting](#)
- [const char](#) [ConfigType](#) [type](#)
- [const char](#) [ConfigValuePtr](#) [val](#)

5.21.1 Detailed Description

Definition at line 337 of file [PluginDefs.h](#).

5.21.2 Member Function Documentation

5.21.2.1 [tagDCConfig::ConfigStrPtr](#) ([DCAPI](#) * [get_path](#))

5.21.2.2 [tagDCConfig::ConfigStrPtr](#) ([DCAPI](#) * [get_install_path](#)) [const](#)

5.21.2.3 [tagDCConfig::ConfigStrPtr](#) ([DCAPI](#) * [get_language](#))

5.21.2.4 [tagDCConfig::ConfigValuePtr](#) ([DCAPI](#) * [get_cfg](#)) [const](#)

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

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

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

5.21.3 Member Data Documentation

5.21.3.1 `uint32_t tagDCConfig::apiVersion`

Definition at line 339 of file PluginDefs.h.

5.21.3.2 `const char * tagDCConfig::setting`

Definition at line 344 of file PluginDefs.h.

5.21.3.3 `const char ConfigType tagDCConfig::type`

Definition at line 345 of file PluginDefs.h.

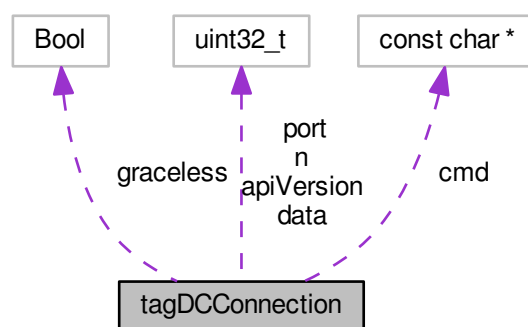
5.21.3.4 `const char ConfigValuePtr tagDCConfig::val`

Definition at line 344 of file PluginDefs.h.

5.22 tagDCConnection Struct Reference

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

Collaboration diagram for tagDCConnection:



Public Member Functions

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

- `void (DCAPI *terminate_conn)(ConnectionDataPtr hConn`

Public Attributes

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

5.22.1 Detailed Description

Definition at line 365 of file PluginDefs.h.

5.22.2 Member Function Documentation

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

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

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

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

5.22.3 Member Data Documentation

5.22.3.1 `uint32_t tagDCConnection::apiVersion`

Definition at line 367 of file PluginDefs.h.

5.22.3.2 `const char* tagDCConnection::cmd`

Definition at line 370 of file PluginDefs.h.

5.22.3.3 `uint32_t dcptr_t tagDCConnection::data`

Definition at line 369 of file PluginDefs.h.

5.22.3.4 `Bool tagDCConnection::graceless`

Definition at line 371 of file PluginDefs.h.

5.22.3.5 `uint32_t dcptr_t size_t tagDCConnection::n`

Definition at line 369 of file PluginDefs.h.

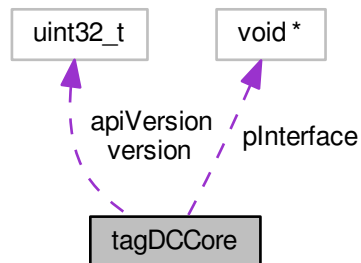
5.22.3.6 `uint32_t tagDCConnection::port`

Definition at line 369 of file PluginDefs.h.

5.23 tagDCCore Struct Reference

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

Collaboration diagram for tagDCCore:



Public Member Functions

- [Bool](#) ([DCAPI](#) *release_interface)([intfHandle](#) hInterface)
- [Bool](#) ([DCAPI](#) *has_plugin)(const char *guid)
- [DCInterfacePtr](#) ([DCAPI](#) *query_interface)(const char *guid
- const char *[DCAPI](#) * host_name ()
- [intfHandle](#) ([DCAPI](#) *register_interface)(const char *guid

Public Attributes

- uint32_t [apiVersion](#)
- [dcptr_t](#) [pInterface](#)
- uint32_t [version](#)

5.23.1 Detailed Description

Definition at line 295 of file PluginDefs.h.

5.23.2 Member Function Documentation

5.23.2.1 tagDCCore::Bool ([DCAPI](#) * release_interface)

5.23.2.2 tagDCCore::Bool ([DCAPI](#) * has_plugin) const

5.23.2.3 tagDCCore::DCInterfacePtr ([DCAPI](#) * query_interface) const

5.23.2.4 const char* [DCAPI](#)* tagDCCore::host_name ()

Referenced by [dcapi::Core::init\(\)](#).

Here is the caller graph for this function:



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

5.23.3 Member Data Documentation

5.23.3.1 `uint32_t tagDCCore::apiVersion`

Definition at line 297 of file `PluginDefs.h`.

5.23.3.2 `dcptr_t tagDCCore::plInterface`

Definition at line 300 of file `PluginDefs.h`.

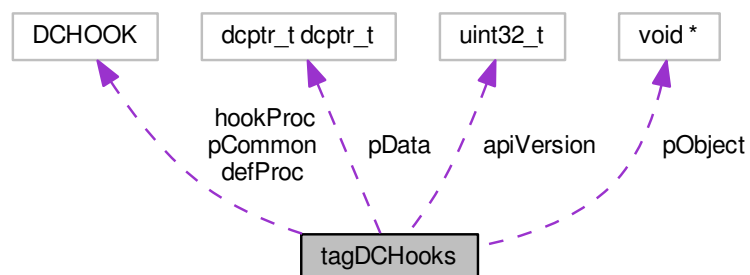
5.23.3.3 `uint32_t tagDCCore::version`

Definition at line 301 of file `PluginDefs.h`.

5.24 tagDCHooks Struct Reference

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

Collaboration diagram for `tagDCHooks`:



Public Member Functions

- [Bool](#) ([DCAPI](#) *destroy_hook)([hookHandle](#) hHook)
- [Bool](#) ([DCAPI](#) *run_hook)([hookHandle](#) hHook)
- [hookHandle](#) ([DCAPI](#) *create_hook)(const char *guid)
- [size_t](#) ([DCAPI](#) *release_hook)([subsHandle](#) hHook)
- [subsHandle](#) ([DCAPI](#) *bind_hook)(const char *guid)

Public Attributes

- [uint32_t](#) [apiVersion](#)
- [DCHOOK](#) [defProc](#)
- [DCHOOK](#) [hookProc](#)
- [DCHOOK](#) [void](#) * [pCommon](#)
- [dcptr_t](#) [dcptr_t](#) [pData](#)
- [dcptr_t](#) [pObject](#)

5.24.1 Detailed Description

Definition at line 320 of file PluginDefs.h.

5.24.2 Member Function Documentation

5.24.2.1 [tagDCHooks::Bool](#) ([DCAPI](#) * *destroy_hook*)

5.24.2.2 [tagDCHooks::Bool](#) ([DCAPI](#) * *run_hook*)

5.24.2.3 [tagDCHooks::hookHandle](#) ([DCAPI](#) * *create_hook*) const

5.24.2.4 [tagDCHooks::size_t](#) ([DCAPI](#) * *release_hook*)

5.24.2.5 [tagDCHooks::subsHandle](#) ([DCAPI](#) * *bind_hook*) const

5.24.3 Member Data Documentation

5.24.3.1 [uint32_t](#) [tagDCHooks::apiVersion](#)

Definition at line 322 of file PluginDefs.h.

5.24.3.2 [DCHOOK](#) [tagDCHooks::defProc](#)

Definition at line 325 of file PluginDefs.h.

5.24.3.3 [DCHOOK](#) [tagDCHooks::hookProc](#)

Definition at line 329 of file PluginDefs.h.

5.24.3.4 [DCHOOK](#) [void](#)* [tagDCHooks::pCommon](#)

Definition at line 329 of file PluginDefs.h.

5.24.3.5 dcptr_t dcptr_t tagDCHooks::pData

Definition at line 330 of file PluginDefs.h.

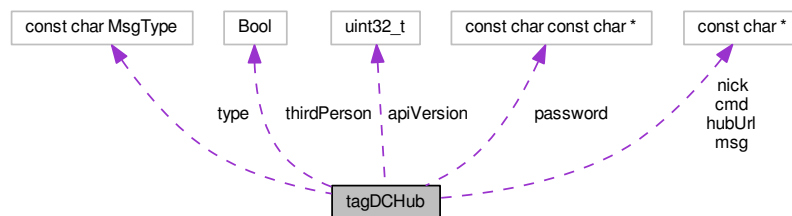
5.24.3.6 dcptr_t tagDCHooks::pObject

Definition at line 330 of file PluginDefs.h.

5.25 tagDCHub Struct Reference

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

Collaboration diagram for tagDCHub:



Public Member Functions

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

Public Attributes

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

5.25.1 Detailed Description

Definition at line 377 of file PluginDefs.h.

5.25.2 Member Function Documentation

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

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

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

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

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

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

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

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

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

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

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

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

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

5.25.3 Member Data Documentation

5.25.3.1 `uint32_t tagDCHub::apiVersion`

Definition at line 379 of file PluginDefs.h.

5.25.3.2 `const char * tagDCHub::cmd`

Definition at line 385 of file PluginDefs.h.

5.25.3.3 `const char* tagDCHub::hubUrl`

Definition at line 392 of file PluginDefs.h.

5.25.3.4 `const char * tagDCHub::msg`

Definition at line 388 of file PluginDefs.h.

5.25.3.5 `const char* tagDCHub::nick`

Definition at line 381 of file PluginDefs.h.

5.25.3.6 const char const char* tagDCHub::password

Definition at line 381 of file PluginDefs.h.

5.25.3.7 const char Bool tagDCHub::thirdPerson

Definition at line 388 of file PluginDefs.h.

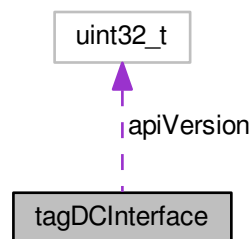
5.25.3.8 const char MsgType tagDCHub::type

Definition at line 389 of file PluginDefs.h.

5.26 tagDCInterface Struct Reference

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

Collaboration diagram for tagDCInterface:



Public Attributes

- uint32_t [apiVersion](#)

5.26.1 Detailed Description

Definition at line 289 of file PluginDefs.h.

5.26.2 Member Data Documentation

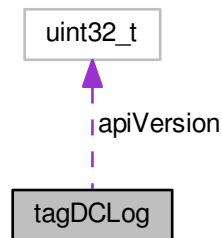
5.26.2.1 uint32_t tagDCInterface::apiVersion

Definition at line 291 of file PluginDefs.h.

5.27 tagDCLog Struct Reference

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

Collaboration diagram for tagDCLog:



Public Member Functions

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

Public Attributes

- `uint32_t apiVersion`

5.27.1 Detailed Description

Definition at line 355 of file `PluginDefs.h`.

5.27.2 Member Function Documentation

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

5.27.3 Member Data Documentation

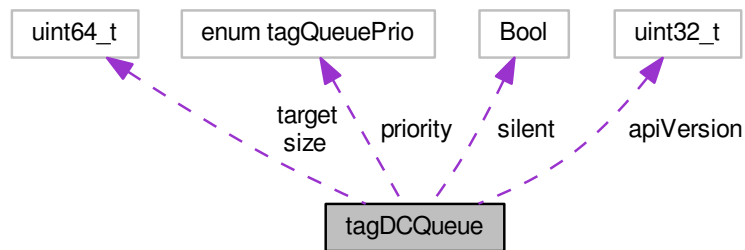
5.27.3.1 `uint32_t tagDCLog::apiVersion`

Definition at line 357 of file `PluginDefs.h`.

5.28 tagDCQueue Struct Reference

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

Collaboration diagram for tagDCQueue:



Public Member Functions

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

Public Attributes

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

5.28.1 Detailed Description

Definition at line 401 of file PluginDefs.h.

5.28.2 Member Function Documentation

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

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

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

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

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

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

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

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

5.28.3 Member Data Documentation

5.28.3.1 `uint32_t tagDCQueue::apiVersion`

Definition at line 403 of file PluginDefs.h.

5.28.3.2 `QueuePrio tagDCQueue::priority`

Definition at line 410 of file PluginDefs.h.

5.28.3.3 `Bool tagDCQueue::silent`

Definition at line 405 of file PluginDefs.h.

5.28.3.4 `uint64_t tagDCQueue::size`

Definition at line 406 of file PluginDefs.h.

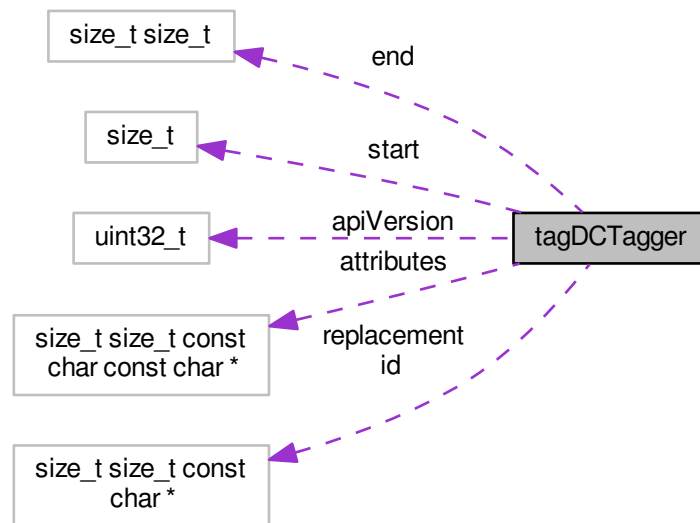
5.28.3.5 `uint64_t const char* tagDCQueue::target`

Definition at line 406 of file PluginDefs.h.

5.29 tagDCTagger Struct Reference

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

Collaboration diagram for tagDCTagger:



Public Member Functions

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

Public Attributes

- `uint32_t apiVersion`
- `size_t size_t const char const char * attributes`
- `size_t size_t end`
- `size_t size_t const char * id`
- `size_t size_t const char * replacement`
- `size_t start`

5.29.1 Detailed Description

Definition at line 435 of file `PluginDefs.h`.

5.29.2 Member Function Documentation

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

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

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

5.29.3 Member Data Documentation

5.29.3.1 `uint32_t tagDCTagger::apiVersion`

Definition at line 437 of file PluginDefs.h.

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

Definition at line 441 of file PluginDefs.h.

5.29.3.3 `size_t size_t tagDCTagger::end`

Definition at line 441 of file PluginDefs.h.

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

Definition at line 441 of file PluginDefs.h.

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

Definition at line 442 of file PluginDefs.h.

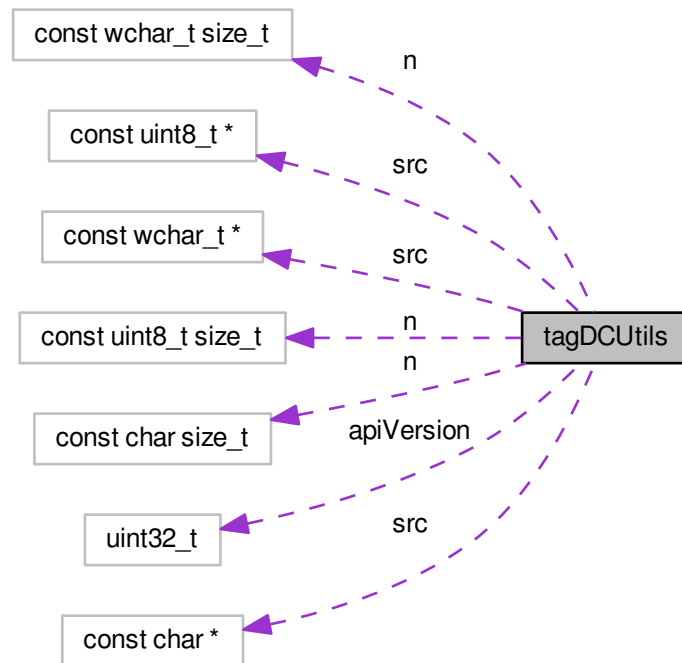
5.29.3.6 `size_t tagDCTagger::start`

Definition at line 441 of file PluginDefs.h.

5.30 tagDCUtils Struct Reference

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


Collaboration diagram for tagDCUtils:



Public Member Functions

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

Public Attributes

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

5.30.1 Detailed Description

Definition at line 418 of file PluginDefs.h.

5.30.2 Member Function Documentation

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

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

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

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

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

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

5.30.3 Member Data Documentation

5.30.3.1 `uint32_t tagDCUtils::apiVersion`

Definition at line 420 of file PluginDefs.h.

5.30.3.2 `const char size_t tagDCUtils::n`

Definition at line 424 of file PluginDefs.h.

5.30.3.3 `const wchar_t size_t tagDCUtils::n`

Definition at line 428 of file PluginDefs.h.

5.30.3.4 `const uint8_t size_t tagDCUtils::n`

Definition at line 430 of file PluginDefs.h.

5.30.3.5 `const char * tagDCUtils::src`

Definition at line 424 of file PluginDefs.h.

5.30.3.6 `const wchar_t* tagDCUtils::src`

Definition at line 428 of file PluginDefs.h.

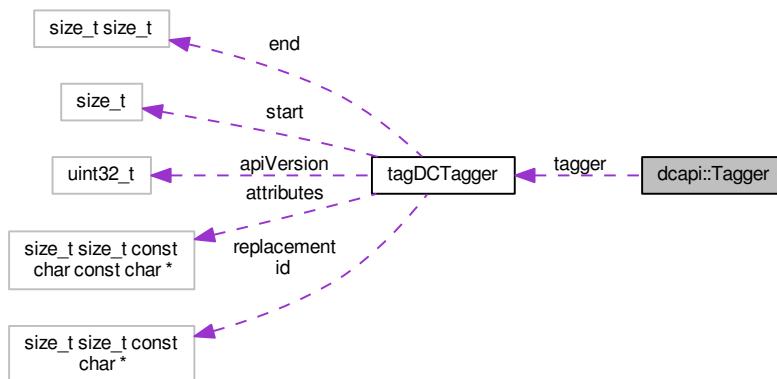
5.30.3.7 `const uint8_t* tagDCUtils::src`

Definition at line 430 of file PluginDefs.h.

5.31 dcapi::Tagger Class Reference

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

Collaboration diagram for dcapi::Tagger:



Static Public Member Functions

- static [DCTaggerPtr handle](#) ()
- static bool [init](#) ()
- static void [init](#) ([DCTaggerPtr](#) coreTagger)

Static Private Attributes

- static [DCTaggerPtr tagger](#)

5.31.1 Detailed Description

Definition at line 28 of file `Tagger.hpp`.

5.31.2 Member Function Documentation

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

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

References `tagger`.

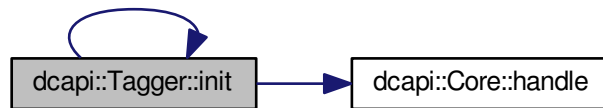
5.31.2.2 bool dcapi::Tagger::init () [static]

Definition at line 29 of file `Tagger.cpp`.

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

Referenced by `init()`.

Here is the call graph for this function:



Here is the caller graph for this function:



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

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

References `tagger`.

5.31.3 Member Data Documentation

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

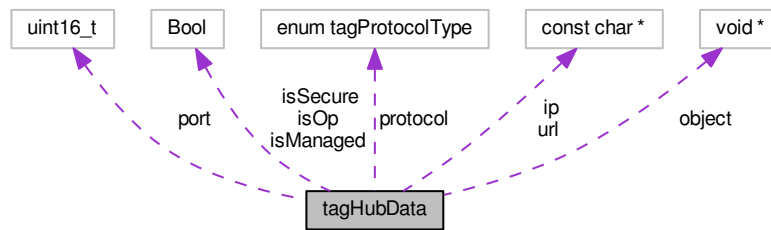
Definition at line 36 of file `Tagger.hpp`.

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

5.32 tagHubData Struct Reference

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

Collaboration diagram for tagHubData:



Public Attributes

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

5.32.1 Detailed Description

Definition at line 230 of file `PluginDefs.h`.

5.32.2 Member Data Documentation

5.32.2.1 `const char* tagHubData::ip`

Definition at line 232 of file `PluginDefs.h`.

5.32.2.2 `Bool tagHubData::isManaged`

Definition at line 238 of file `PluginDefs.h`.

5.32.2.3 `Bool tagHubData::isOp`

Definition at line 236 of file `PluginDefs.h`.

5.32.2.4 `Bool tagHubData::isSecure`

Definition at line 237 of file `PluginDefs.h`.

5.32.2.5 `dcptr_t tagHubData::object`

Definition at line 233 of file `PluginDefs.h`.

5.32.2.6 uint16_t tagHubData::port

Definition at line 234 of file PluginDefs.h.

5.32.2.7 ProtocolType tagHubData::protocol

Definition at line 235 of file PluginDefs.h.

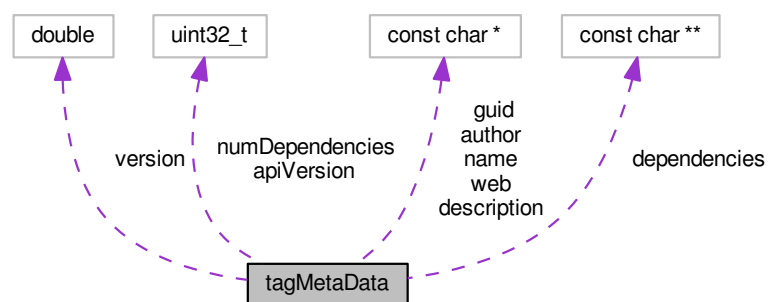
5.32.2.8 const char* tagHubData::url

Definition at line 231 of file PluginDefs.h.

5.33 tagMetaData Struct Reference

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

Collaboration diagram for tagMetaData:



Public Attributes

- uint32_t [apiVersion](#)
- const char * [author](#)
- const char ** [dependencies](#)
- const char * [description](#)
- const char * [guid](#)
- const char * [name](#)
- uint32_t [numDependencies](#)
- double [version](#)
- const char * [web](#)

5.33.1 Detailed Description

Definition at line 276 of file PluginDefs.h.

5.33.2 Member Data Documentation

5.33.2.1 uint32_t tagMetaData::apiVersion

Definition at line 284 of file PluginDefs.h.

Referenced by pluginInit().

5.33.2.2 const char* tagMetaData::author

Definition at line 278 of file PluginDefs.h.

Referenced by pluginInit().

5.33.2.3 const char** tagMetaData::dependencies

Definition at line 282 of file PluginDefs.h.

5.33.2.4 const char* tagMetaData::description

Definition at line 279 of file PluginDefs.h.

Referenced by pluginInit().

5.33.2.5 const char* tagMetaData::guid

Definition at line 281 of file PluginDefs.h.

Referenced by pluginInit().

5.33.2.6 const char* tagMetaData::name

Definition at line 277 of file PluginDefs.h.

Referenced by pluginInit().

5.33.2.7 uint32_t tagMetaData::numDependencies

Definition at line 283 of file PluginDefs.h.

5.33.2.8 double tagMetaData::version

Definition at line 285 of file PluginDefs.h.

Referenced by pluginInit().

5.33.2.9 const char* tagMetaData::web

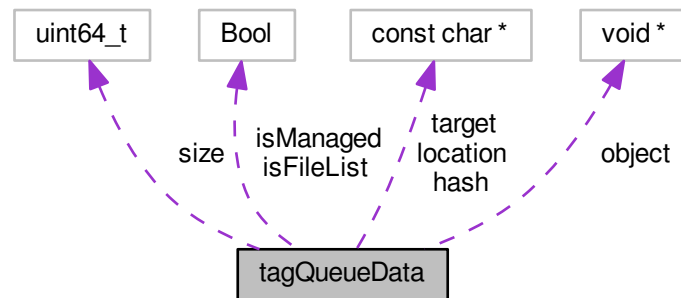
Definition at line 280 of file PluginDefs.h.

Referenced by pluginInit().

5.34 tagQueueData Struct Reference

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

Collaboration diagram for tagQueueData:



Public Attributes

- `const char *` [hash](#)
- `Bool` [isFileList](#)
- `Bool` [isManaged](#)
- `const char *` [location](#)
- `dcptr_t` [object](#)
- `uint64_t` [size](#)
- `const char *` [target](#)

5.34.1 Detailed Description

Definition at line 259 of file `PluginDefs.h`.

5.34.2 Member Data Documentation

5.34.2.1 `const char* tagQueueData::hash`

Definition at line 262 of file `PluginDefs.h`.

5.34.2.2 `Bool tagQueueData::isFileList`

Definition at line 265 of file `PluginDefs.h`.

5.34.2.3 `Bool tagQueueData::isManaged`

Definition at line 266 of file `PluginDefs.h`.

5.34.2.4 `const char* tagQueueData::location`

Definition at line 261 of file PluginDefs.h.

5.34.2.5 `dcptr_t tagQueueData::object`

Definition at line 263 of file PluginDefs.h.

5.34.2.6 `uint64_t tagQueueData::size`

Definition at line 264 of file PluginDefs.h.

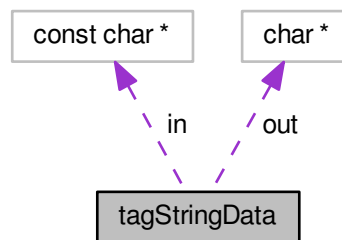
5.34.2.7 `const char* tagQueueData::target`

Definition at line 260 of file PluginDefs.h.

5.35 tagStringData Struct Reference

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

Collaboration diagram for tagStringData:



Public Attributes

- `const char *` [in](#)
- `char *` [out](#)

5.35.1 Detailed Description

Definition at line 206 of file PluginDefs.h.

5.35.2 Member Data Documentation

5.35.2.1 `const char* tagStringData::in`

Definition at line 207 of file PluginDefs.h.

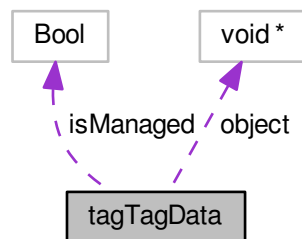
5.35.2.2 `char* tagStringData::out`

Definition at line 208 of file PluginDefs.h.

5.36 tagTagData Struct Reference

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

Collaboration diagram for tagTagData:



Public Attributes

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

5.36.1 Detailed Description

Definition at line 270 of file PluginDefs.h.

5.36.2 Member Data Documentation

5.36.2.1 `Bool tagTagData::isManaged`

Definition at line 272 of file PluginDefs.h.

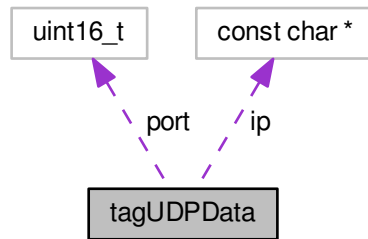
5.36.2.2 `dcptr_t tagTagData::object`

Definition at line 271 of file PluginDefs.h.

5.37 tagUDPData Struct Reference

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

Collaboration diagram for tagUDPData:



Public Attributes

- `const char *` [ip](#)
- `uint16_t` [port](#)

5.37.1 Detailed Description

Definition at line 253 of file `PluginDefs.h`.

5.37.2 Member Data Documentation

5.37.2.1 `const char* tagUDPData::ip`

Definition at line 254 of file `PluginDefs.h`.

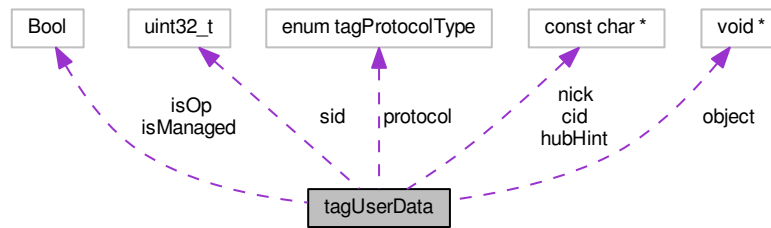
5.37.2.2 `uint16_t tagUDPData::port`

Definition at line 255 of file `PluginDefs.h`.

5.38 tagUserData Struct Reference

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

Collaboration diagram for tagUserData:



Public Attributes

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

5.38.1 Detailed Description

Definition at line 218 of file PluginDefs.h.

5.38.2 Member Data Documentation

5.38.2.1 `const char* tagUserData::cid`

Definition at line 221 of file PluginDefs.h.

5.38.2.2 `const char* tagUserData::hubHint`

Definition at line 220 of file PluginDefs.h.

5.38.2.3 `Bool tagUserData::isManaged`

Definition at line 226 of file PluginDefs.h.

5.38.2.4 `Bool tagUserData::isOp`

Definition at line 225 of file PluginDefs.h.

5.38.2.5 `const char* tagUserData::nick`

Definition at line 219 of file PluginDefs.h.

5.38.2.6 dcptr_t tagUserData::object

Definition at line 222 of file PluginDefs.h.

5.38.2.7 ProtocolType tagUserData::protocol

Definition at line 224 of file PluginDefs.h.

5.38.2.8 uint32_t tagUserData::sid

Definition at line 223 of file PluginDefs.h.

5.39 dcapi::Hooks::Timer Struct Reference

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

Static Public Member Functions

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

5.39.1 Detailed Description

Definition at line 61 of file Hooks.hpp.

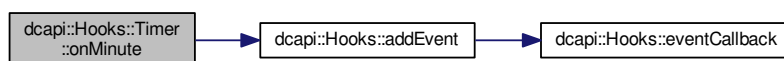
5.39.2 Member Function Documentation

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

Definition at line 59 of file Hooks.cpp.

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

Here is the call graph for this function:



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

Definition at line 55 of file Hooks.cpp.

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

Here is the call graph for this function:



5.40 dcapi::Hooks::UI Struct Reference

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

Static Public Member Functions

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

5.40.1 Detailed Description

Definition at line 92 of file Hooks.hpp.

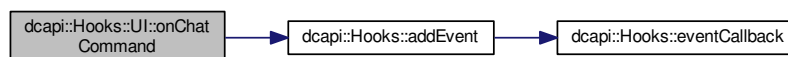
5.40.2 Member Function Documentation

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

Definition at line 136 of file Hooks.cpp.

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

Here is the call graph for this function:



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

Definition at line 140 of file Hooks.cpp.

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

Here is the call graph for this function:

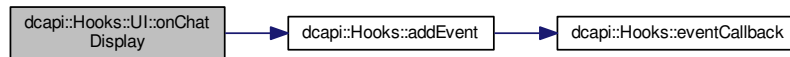


5.40.2.3 void dcapi::Hooks::UI::onChatDisplay (function< bool(UserDataPtr, StringDataPtr, bool &)> f) [static]

Definition at line 132 of file Hooks.cpp.

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

Here is the call graph for this function:

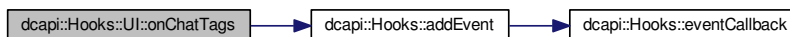


5.40.2.4 void dcapi::Hooks::UI::onChatTags (function< bool(UserDataPtr, TagDataPtr, bool &)> f) [static]

Definition at line 128 of file Hooks.cpp.

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

Here is the call graph for this function:

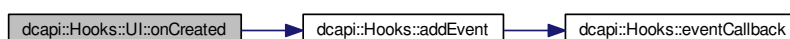


5.40.2.5 void dcapi::Hooks::UI::onCreated (function< bool(dcptr_t, bool &)> f) [static]

Definition at line 124 of file Hooks.cpp.

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

Here is the call graph for this function:



5.41.3 Member Function Documentation

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

Definition at line 39 of file UI.cpp.

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

Here is the call graph for this function:



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

Definition at line 49 of file UI.cpp.

References `commands`, `guid`, and `ui`.

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

Definition at line 56 of file UI.cpp.

References `commands`.

Referenced by `addCommand()`.

Here is the caller graph for this function:



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

Definition at line 37 of file UI.cpp.

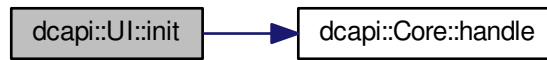
References `ui`.

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

Definition at line 31 of file UI.cpp.

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

Here is the call graph for this function:



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

Definition at line 36 of file UI.cpp.

References guid, and ui.

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

Definition at line 44 of file UI.cpp.

References commands, guid, and ui.

5.41.4 Member Data Documentation

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

Definition at line 57 of file UI.hpp.

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

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

Definition at line 54 of file UI.hpp.

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

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

Definition at line 52 of file UI.hpp.

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

5.42 dcapi::Hooks::Users Struct Reference

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

Static Public Member Functions

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

5.42.1 Detailed Description

Definition at line 71 of file Hooks.hpp.

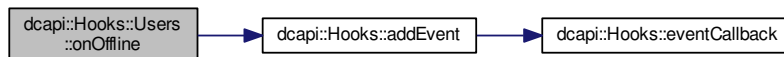
5.42.2 Member Function Documentation

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

Definition at line 77 of file Hooks.cpp.

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

Here is the call graph for this function:

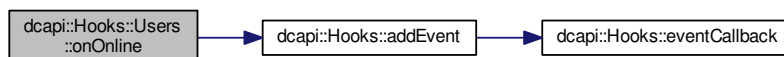


5.42.2.2 void dcapi::Hooks::Users::onOnline (function< bool(UserDataPtr, bool &);> f) [static]

Definition at line 73 of file Hooks.cpp.

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

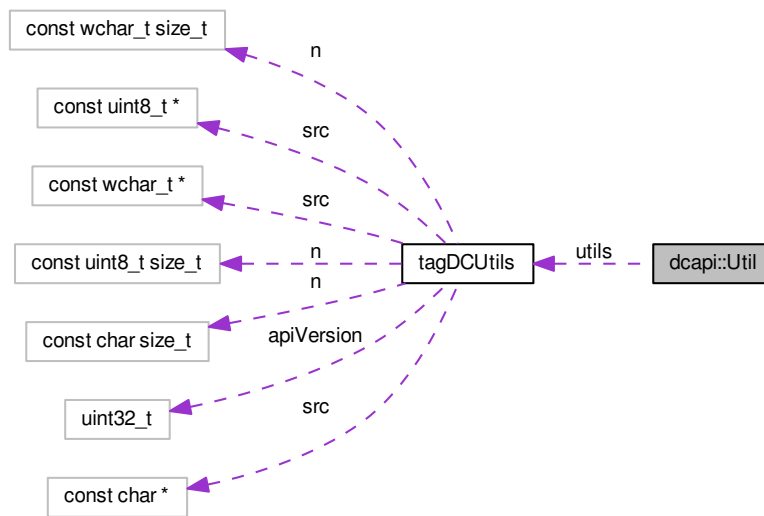
Here is the call graph for this function:



5.43 dcapi::Util Class Reference

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

Collaboration diagram for `dcapi::Util`:



Static Public Member Functions

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

Static Private Attributes

- static [DCUtilsPtr utils](#)

5.43.1 Detailed Description

Definition at line 35 of file `Util.hpp`.

5.43.2 Member Function Documentation

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

Definition at line 68 of file `Util.cpp`.

References `utils`.

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

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

References `utils`.

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

Definition at line 29 of file Util.cpp.

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

Here is the call graph for this function:



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

Definition at line 34 of file Util.cpp.

References `utils`.

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

Definition at line 81 of file Util.cpp.

References `utils`.

5.43.3 Member Data Documentation

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

Definition at line 55 of file Util.hpp.

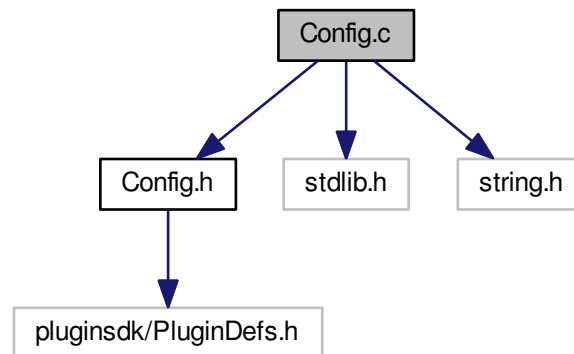
Referenced by `fromUtf8()`, `handle()`, `init()`, and `toUtf8()`.

Chapter 6

File Documentation

6.1 Config.c File Reference

```
#include "Config.h"  
#include <stdlib.h>  
#include <string.h>  
Include dependency graph for Config.c:
```



Functions

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

Variables

- [DCConfigPtr config](#)
- `char * guid`

6.1.1 Function Documentation

6.1.1.1 `void DCAPI free_core_cfg (ConfigValuePtr val)`

Definition at line 100 of file Config.c.

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

Definition at line 43 of file Config.c.

References CFG_TYPE_STRING, [guid](#), and [tagConfigStr::value](#).

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

Definition at line 54 of file Config.c.

References CFG_TYPE_INT, [guid](#), and [tagConfigInt::value](#).

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

Definition at line 63 of file Config.c.

References CFG_TYPE_INT64, [guid](#), and [tagConfigInt64::value](#).

6.1.1.5 `char* DCAPI get_cfg_path (PathType path)`

Definition at line 102 of file Config.c.

References [tagConfigStr::value](#).

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

Definition at line 99 of file Config.c.

References CFG_TYPE_UNKNOWN.

6.1.1.7 `char* DCAPI get_install_path ()`

Definition at line 113 of file Config.c.

References [guid](#), and [tagConfigStr::value](#).

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

Definition at line 29 of file Config.c.

References DCINTF_CONFIG, DCINTF_CONFIG_VER, `False`, [guid](#), and `True`.

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

Definition at line 72 of file Config.c.

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

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

Definition at line 81 of file Config.c.

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

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

Definition at line 90 of file Config.c.

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

6.1.1.12 void DCAPI uninit_cfg ()

Definition at line 39 of file Config.c.

References guid.

6.1.2 Variable Documentation

6.1.2.1 DConfigPtr config

Definition at line 26 of file Config.c.

6.1.2.2 char* guid

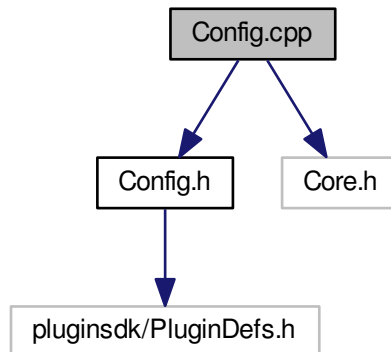
Definition at line 27 of file Config.c.

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

6.2 Config.cpp File Reference

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

Include dependency graph for Config.cpp:



Namespaces

- [dcapi](#)

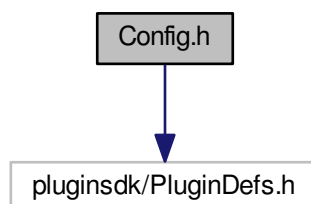
Constant Groups

- [dcapi](#)

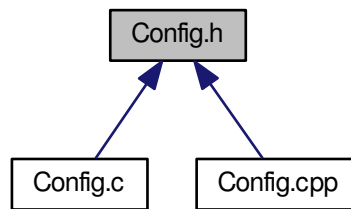
6.3 Config.h File Reference

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

Include dependency graph for Config.h:



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



Functions

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

6.3.1 Function Documentation

6.3.1.1 void DCAPI free_core_cfg (ConfigValuePtr val)

Definition at line 100 of file Config.c.

6.3.1.2 char* DCAPI get_cfg (const char * name)

Definition at line 43 of file Config.c.

References CFG_TYPE_STRING, guid, and tagConfigStr::value.

6.3.1.3 int32_t DCAPI get_cfg_int (const char * name)

Definition at line 54 of file Config.c.

References CFG_TYPE_INT, guid, and tagConfigInt::value.

6.3.1.4 int64_t DCAPI get_cfg_int64 (const char * name)

Definition at line 63 of file Config.c.

References CFG_TYPE_INT64, guid, and tagConfigInt64::value.

6.3.1.5 `char* DCAPI get_cfg_path (PathType path)`

Definition at line 102 of file Config.c.

References tagConfigStr::value.

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

Definition at line 99 of file Config.c.

References CFG_TYPE_UNKNOWN.

6.3.1.7 `char* DCAPI get_install_path ()`

Definition at line 113 of file Config.c.

References guid, and tagConfigStr::value.

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

Definition at line 29 of file Config.c.

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

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

Definition at line 72 of file Config.c.

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

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

Definition at line 81 of file Config.c.

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

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

Definition at line 90 of file Config.c.

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

6.3.1.12 `void DCAPI uninit_cfg ()`

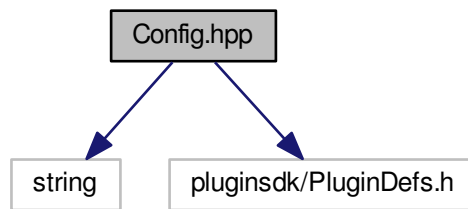
Definition at line 39 of file Config.c.

References guid.

6.4 Config.hpp File Reference

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

Include dependency graph for Config.hpp:



Classes

- class [dcapi::Config](#)

Namespaces

- [dcapi](#)

Constant Groups

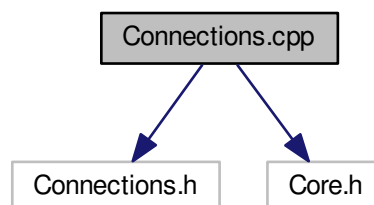
- [dcapi](#)

6.5 Connections.cpp File Reference

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

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

Include dependency graph for Connections.cpp:



Namespaces

- [dcapi](#)

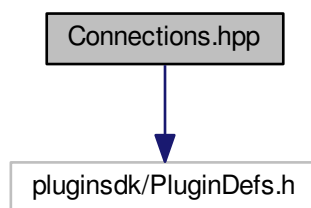
Constant Groups

- [dcapi](#)

6.6 Connections.hpp File Reference

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

Include dependency graph for Connections.hpp:



Classes

- class [dcapi::Connections](#)

Namespaces

- [dcapi](#)

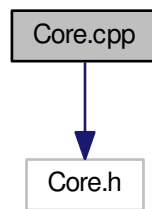
Constant Groups

- [dcapi](#)

6.7 Core.cpp File Reference

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

Include dependency graph for Core.cpp:



Namespaces

- [dcapi](#)

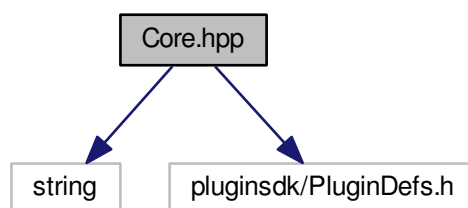
Constant Groups

- [dcapi](#)

6.8 Core.hpp File Reference

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

Include dependency graph for Core.hpp:



Classes

- class [dcapi::Core](#)

Namespaces

- [dcapi](#)

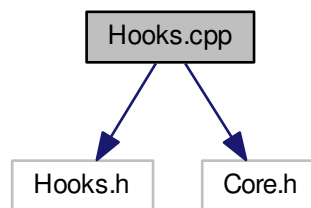
Constant Groups

- [dcapi](#)

6.9 Hooks.cpp File Reference

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

Include dependency graph for Hooks.cpp:



Namespaces

- [dcapi](#)

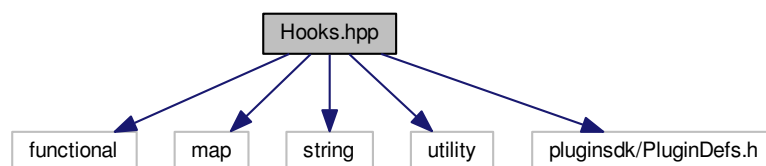
Constant Groups

- [dcapi](#)

6.10 Hooks.hpp File Reference

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

Include dependency graph for Hooks.hpp:



Classes

- struct [dcapi::Hooks::Chat](#)
- class [dcapi::Hooks](#)
- struct [dcapi::Hooks::Hubs](#)
- struct [dcapi::Hooks::Network](#)
- struct [dcapi::Hooks::Queue](#)
- struct [dcapi::Hooks::Timer](#)
- struct [dcapi::Hooks::UI](#)
- struct [dcapi::Hooks::Users](#)

Namespaces

- [dcapi](#)

Constant Groups

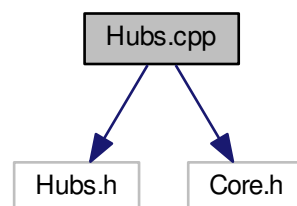
- [dcapi](#)

6.11 Hubs.cpp File Reference

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

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

Include dependency graph for Hubs.cpp:



Namespaces

- [dcapi](#)

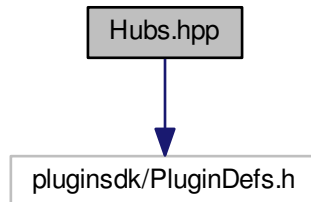
Constant Groups

- [dcapi](#)

6.12 Hubs.hpp File Reference

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

Include dependency graph for Hubs.hpp:



Classes

- class [dcapi::Hubs](#)

Namespaces

- [dcapi](#)

Constant Groups

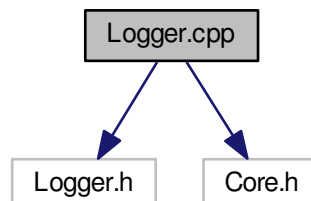
- [dcapi](#)

6.13 Logger.cpp File Reference

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

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

Include dependency graph for Logger.cpp:



Namespaces

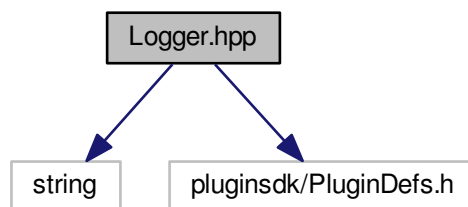
- [dcapi](#)

Constant Groups

- [dcapi](#)

6.14 Logger.hpp File Reference

```
#include <string>
#include <pluginsdk/PluginDefs.h>
Include dependency graph for Logger.hpp:
```



Classes

- class [dcapi::Logger](#)

Namespaces

- [dcapi](#)

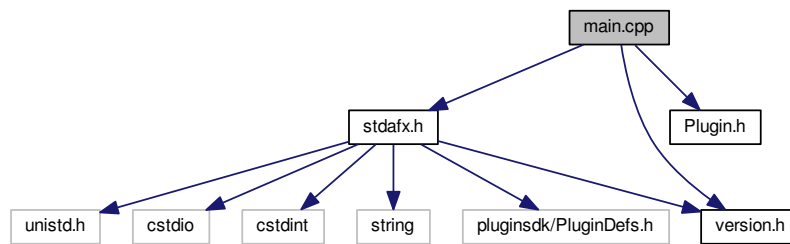
Constant Groups

- [dcapi](#)

6.15 main.cpp File Reference

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

Include dependency graph for main.cpp:



Functions

- [DCEXP DCMAIN DCAPI pluginInit](#) ([MetaDataPtr](#) info)

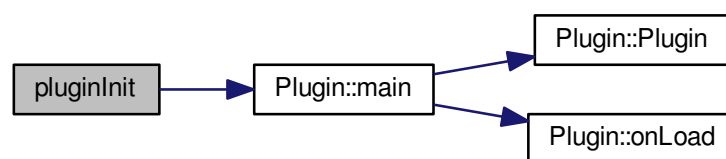
6.15.1 Function Documentation

6.15.1.1 DCEXP DCMAIN DCAPI pluginInit ([MetaDataPtr](#) info)

Definition at line 26 of file main.cpp.

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

Here is the call graph for this function:



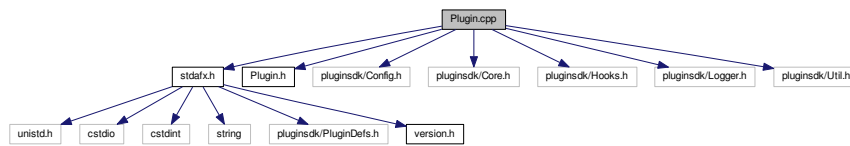
6.16 Plugin.cpp File Reference

```

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

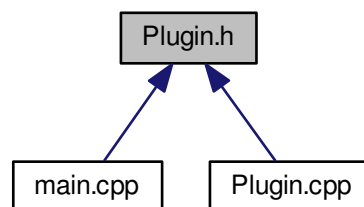
```

Include dependency graph for Plugin.cpp:



6.17 Plugin.h File Reference

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



Classes

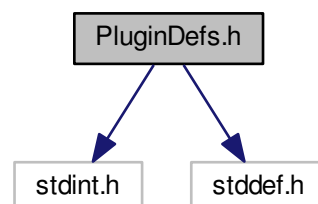
- class [Plugin](#)

6.18 PluginDefs.h File Reference

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

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

Include dependency graph for PluginDefs.h:



Classes

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

Macros

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

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

Typedefs

- typedef [dcptr_t](#) [dcptr_t](#) [Bool](#) * [bBreak](#)
- typedef struct [tagCommandData](#) [CommandData](#)
- typedef struct [tagCommandData](#) * [CommandDataPtr](#)
- typedef struct [tagConfigBool](#) [ConfigBool](#)
- typedef struct [tagConfigBool](#) * [ConfigBoolPtr](#)
- typedef struct [tagConfigInt](#) [ConfigInt](#)
- typedef struct [tagConfigInt64](#) [ConfigInt64](#)
- typedef struct [tagConfigInt64](#) * [ConfigInt64Ptr](#)
- typedef struct [tagConfigInt](#) * [ConfigIntPtr](#)
- typedef struct [tagConfigStr](#) [ConfigStr](#)

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

Enumerations

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

Functions

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

6.18.1 Macro Definition Documentation

6.18.1.1 #define Bool dcBool

Definition at line 172 of file `PluginDefs.h`.

6.18.1.2 #define DCAPI

Definition at line 49 of file `PluginDefs.h`.

6.18.1.3 #define DCAPI_CORE_VER 8

Definition at line 35 of file `PluginDefs.h`.

Referenced by `pluginInit()`.

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

Definition at line 51 of file `PluginDefs.h`.

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

Definition at line 52 of file `PluginDefs.h`.

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

Definition at line 60 of file `PluginDefs.h`.

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

6.18.1.7 #define DCINTF_CONFIG_VER 1

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

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

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

Definition at line 67 of file PluginDefs.h.

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

6.18.1.9 `#define DCINTF_DCPP_CONNECTIONS_VER 1`

Definition at line 68 of file PluginDefs.h.

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

6.18.1.10 `#define DCINTF_DCPP_HUBS "dcpp.network.DCHub" /* Hubs */`

Definition at line 70 of file PluginDefs.h.

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

6.18.1.11 `#define DCINTF_DCPP_HUBS_VER 1`

Definition at line 71 of file PluginDefs.h.

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

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

Definition at line 73 of file PluginDefs.h.

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

6.18.1.13 `#define DCINTF_DCPP_QUEUE_VER 1`

Definition at line 74 of file PluginDefs.h.

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

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

Definition at line 79 of file PluginDefs.h.

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

6.18.1.15 `#define DCINTF_DCPP_TAGGER_VER 1`

Definition at line 80 of file PluginDefs.h.

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

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

Definition at line 82 of file PluginDefs.h.

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

6.18.1.17 #define DCINTF_DCPP_UI_VER 1

Definition at line 83 of file PluginDefs.h.

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

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

Definition at line 76 of file PluginDefs.h.

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

6.18.1.19 #define DCINTF_DCPP_UTILS_VER 1

Definition at line 77 of file PluginDefs.h.

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

6.18.1.20 #define DCINTF_HOOKS "generic.plugins.DCHooks"

Definition at line 56 of file PluginDefs.h.

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

6.18.1.21 #define DCINTF_HOOKS_VER DCAPI_CORE_VER

Definition at line 57 of file PluginDefs.h.

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

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

Definition at line 63 of file PluginDefs.h.

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

6.18.1.23 #define DCINTF_LOGGING_VER 1

Definition at line 64 of file PluginDefs.h.

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

6.18.1.24 #define False dcFalse

Definition at line 174 of file PluginDefs.h.

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

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

Definition at line 86 of file PluginDefs.h.

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

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

Definition at line 87 of file PluginDefs.h.

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

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

Definition at line 88 of file PluginDefs.h.

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

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

Definition at line 89 of file PluginDefs.h.

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

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

Definition at line 95 of file PluginDefs.h.

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

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

Definition at line 94 of file PluginDefs.h.

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

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

Definition at line 101 of file PluginDefs.h.

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

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

Definition at line 102 of file PluginDefs.h.

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

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

Definition at line 99 of file PluginDefs.h.

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

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

Definition at line 100 of file PluginDefs.h.

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

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

Definition at line 103 of file PluginDefs.h.

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

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

Definition at line 104 of file PluginDefs.h.

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

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

Definition at line 106 of file PluginDefs.h.

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

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

Definition at line 109 of file PluginDefs.h.

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

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

Definition at line 107 of file PluginDefs.h.

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

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

Definition at line 108 of file PluginDefs.h.

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

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

Definition at line 92 of file PluginDefs.h.

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

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

Definition at line 91 of file PluginDefs.h.

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

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

Definition at line 114 of file PluginDefs.h.

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

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

Definition at line 115 of file `PluginDefs.h`.

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

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

Definition at line 113 of file `PluginDefs.h`.

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

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

Definition at line 112 of file `PluginDefs.h`.

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

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

Definition at line 111 of file `PluginDefs.h`.

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

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

Definition at line 97 of file `PluginDefs.h`.

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

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

Definition at line 96 of file `PluginDefs.h`.

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

6.18.1.50 `#define True dcTrue`

Definition at line 173 of file `PluginDefs.h`.

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

6.18.2 Typedef Documentation

6.18.2.1 `typedef dcptr_t dcptr_t Bool* bBreak`

Definition at line 317 of file `PluginDefs.h`.

- 6.18.2.2 `typedef struct tagCommandData CommandData`
 - 6.18.2.3 `typedef struct tagCommandData * CommandDataPtr`
 - 6.18.2.4 `typedef struct tagConfigBool ConfigBool`
 - 6.18.2.5 `typedef struct tagConfigBool * ConfigBoolPtr`
 - 6.18.2.6 `typedef struct tagConfigInt ConfigInt`
 - 6.18.2.7 `typedef struct tagConfigInt64 ConfigInt64`
 - 6.18.2.8 `typedef struct tagConfigInt64 * ConfigInt64Ptr`
 - 6.18.2.9 `typedef struct tagConfigInt * ConfigIntPtr`
 - 6.18.2.10 `typedef struct tagConfigStr ConfigStr`
 - 6.18.2.11 `typedef struct tagConfigStr * ConfigStrPtr`
 - 6.18.2.12 `typedef enum tagConfigType ConfigType`
 - 6.18.2.13 `typedef struct tagConfigValue ConfigValue`
 - 6.18.2.14 `typedef struct tagConfigValue * ConfigValuePtr`
 - 6.18.2.15 `typedef struct tagConnectionData ConnectionData`
 - 6.18.2.16 `typedef struct tagConnectionData * ConnectionDataPtr`
 - 6.18.2.17 `typedef DCCorePtr core`
- Definition at line 312 of file PluginDefs.h.
- 6.18.2.18 `typedef enum tagDCBool dcBool`
 - 6.18.2.19 `typedef struct tagDCConfig DCConfig`
 - 6.18.2.20 `typedef struct tagDCConfig * DCConfigPtr`
 - 6.18.2.21 `typedef struct tagDCConnection DCConnection`
 - 6.18.2.22 `typedef struct tagDCConnection * DCConnectionPtr`
 - 6.18.2.23 `typedef struct tagDCCore DCCore`
 - 6.18.2.24 `typedef struct tagDCCore * DCCorePtr`
 - 6.18.2.25 `typedef struct tagDCHooks DCHooks`
 - 6.18.2.26 `typedef struct tagDCHooks * DCHooksPtr`
 - 6.18.2.27 `typedef struct tagDCHub DCHub`
 - 6.18.2.28 `typedef struct tagDCHub * DCHubPtr`

6.18.2.29 `typedef struct tagDCInterface DCInterface`

6.18.2.30 `typedef struct tagDCInterface * DCInterfacePtr`

6.18.2.31 `typedef struct tagDCLog DCLog`

6.18.2.32 `typedef struct tagDCLog * DCLogPtr`

6.18.2.33 `typedef void * dcptr_t`

Definition at line 167 of file PluginDefs.h.

6.18.2.34 `typedef struct tagDCQueue DCQueue`

6.18.2.35 `typedef struct tagDCQueue * DCQueuePtr`

6.18.2.36 `typedef struct tagDCTagger DCTagger`

6.18.2.37 `typedef struct tagDCTagger * DCTaggerPtr`

6.18.2.38 `typedef uint64_t dctime_t`

Definition at line 169 of file PluginDefs.h.

6.18.2.39 `typedef struct DCUI DCUI`

6.18.2.40 `typedef struct DCUI * DCUIPtr`

6.18.2.41 `typedef struct tagDCUtils DCUtils`

6.18.2.42 `typedef struct tagDCUtils * DCUtilsPtr`

6.18.2.43 `typedef void* hookHandle`

Definition at line 167 of file PluginDefs.h.

6.18.2.44 `typedef struct tagHubData HubData`

6.18.2.45 `typedef struct tagHubData * HubDataPtr`

6.18.2.46 `typedef void * intfHandle`

Definition at line 167 of file PluginDefs.h.

6.18.2.47 `typedef struct tagMetaData MetaData`

6.18.2.48 `typedef struct tagMetaData * MetaDataPtr`

6.18.2.49 `typedef enum tagMsgType MsgType`

6.18.2.50 `typedef enum tagPathType PathType`

6.18.2.51 typedef dcptr_t dcptr_t pCommon

Definition at line 317 of file PluginDefs.h.

6.18.2.52 typedef dcptr_t pData

Definition at line 312 of file PluginDefs.h.

6.18.2.53 typedef enum tagPluginState PluginState

6.18.2.54 typedef enum tagProtocolType ProtocolType

6.18.2.55 typedef struct tagQueueData QueueData

6.18.2.56 typedef struct tagQueueData * QueueDataPtr

6.18.2.57 typedef enum tagQueuePrio QueuePrio

6.18.2.58 typedef struct tagStringData StringData

6.18.2.59 typedef struct tagStringData * StringDataPtr

6.18.2.60 typedef void * subsHandle

Definition at line 167 of file PluginDefs.h.

6.18.2.61 typedef struct tagTagData TagData

6.18.2.62 typedef struct tagTagData * TagDataPtr

6.18.2.63 typedef struct tagUDPData UDPData

6.18.2.64 typedef struct tagUDPData * UDPDataPtr

6.18.2.65 typedef struct tagUserData UserData

6.18.2.66 typedef struct tagUserData * UserDataPtr

6.18.3 Enumeration Type Documentation

6.18.3.1 enum tagConfigType

Enumerator

CFG_TYPE_UNKNOWN

CFG_TYPE_REMOVE

CFG_TYPE_STRING

CFG_TYPE_INT

CFG_TYPE_BOOL

CFG_TYPE_INT64

Definition at line 128 of file PluginDefs.h.

6.18.3.2 enum tagDCBool

Enumerator

dcFalse

dcTrue

Definition at line 168 of file PluginDefs.h.

6.18.3.3 enum tagMsgType

Enumerator

MSG_CLIENT

MSG_STATUS

MSG_SYSTEM

Definition at line 151 of file PluginDefs.h.

6.18.3.4 enum tagPathType

Enumerator

PATH_GLOBAL_CONFIG

PATH_USER_CONFIG

PATH_USER_LOCAL

PATH_RESOURCES

PATH_LOCALE

Definition at line 143 of file PluginDefs.h.

6.18.3.5 enum tagPluginState

Enumerator

ON_INSTALL

ON_LOAD

ON_LOAD_RUNTIME

ON_CONFIGURE

ON_UNLOAD

ON_UNINSTALL

Definition at line 118 of file PluginDefs.h.

6.18.3.6 enum tagProtocolType

Enumerator

PROTOCOL_ADC

PROTOCOL_NMDC

PROTOCOL_DHT

Definition at line 137 of file PluginDefs.h.

6.18.3.7 enum tagQueuePrio

Enumerator

PRIO_DEFAULT**PRIO_LOWEST****PRIO_LOW****PRIO_NORMAL****PRIO_HIGH****PRIO_HIGHEST**

Definition at line 157 of file PluginDefs.h.

6.18.4 Function Documentation

6.18.4.1 typedef Bool (DCAPI * DCMAIN)

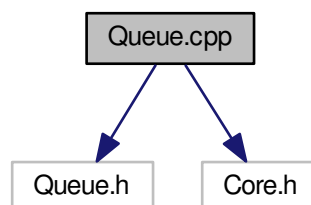
6.18.4.2 typedef void (DCAPI * DCCommandFunc) const

6.19 Queue.cpp File Reference

#include "Queue.h"

#include "Core.h"

Include dependency graph for Queue.cpp:



Namespaces

- [dcapi](#)

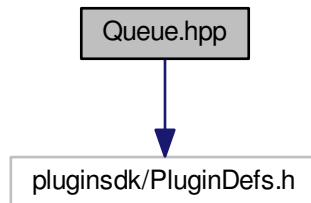
Constant Groups

- [dcapi](#)

6.20 Queue.hpp File Reference

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

Include dependency graph for Queue.hpp:



Classes

- class `dcapi::Queue`

Namespaces

- `dcapi`

Constant Groups

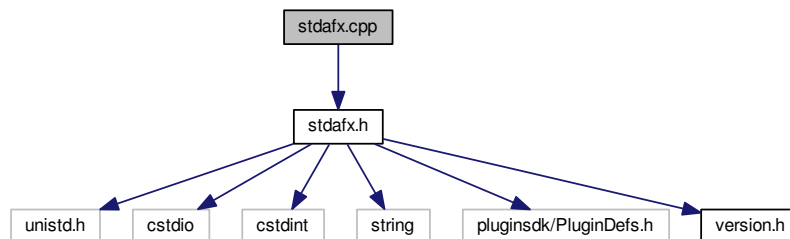
- `dcapi`

6.21 resource.h File Reference

6.22 stdafx.cpp File Reference

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

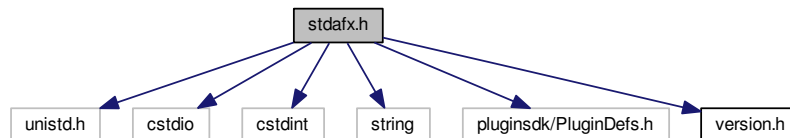
Include dependency graph for stdafx.cpp:



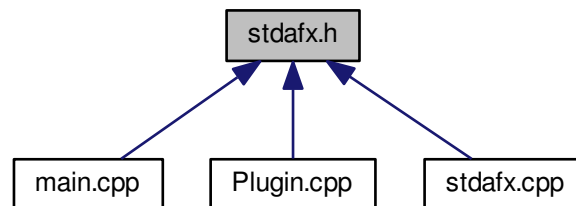
6.23 stdafx.h File Reference

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

Include dependency graph for stdafx.h:



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



Macros

- `#define` [PLUGIN_VERSION](#)

Typedefs

- `typedef std::string` [tstring](#)

6.23.1 Macro Definition Documentation

6.23.1.1 `#define` [PLUGIN_VERSION](#)

Value:

```
(PLUGIN\_VERSION\_MAJOR + (PLUGIN\_VERSION\_MINOR >= 10 ?
    PLUGIN\_VERSION\_MINOR >= 100 ? \
    PLUGIN\_VERSION\_MINOR / 1000. : PLUGIN\_VERSION\_MINOR / 100. :
    PLUGIN\_VERSION\_MINOR / 10.))
```

Definition at line 41 of file stdafx.h.

Referenced by pluginInit().

6.23.2 Typedef Documentation

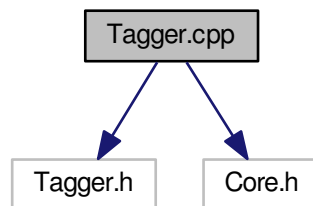
6.23.2.1 typedef std::string tstring

Definition at line 47 of file stdafx.h.

6.24 Tagger.cpp File Reference

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

Include dependency graph for Tagger.cpp:



Namespaces

- [dcapi](#)

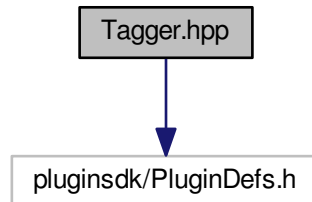
Constant Groups

- [dcapi](#)

6.25 Tagger.hpp File Reference

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

Include dependency graph for Tagger.hpp:



Classes

- class [dcapi::Tagger](#)

Namespaces

- [dcapi](#)

Constant Groups

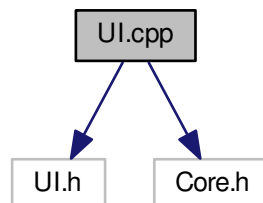
- [dcapi](#)

6.26 UI.cpp File Reference

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

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

Include dependency graph for UI.cpp:



Namespaces

- [dcapi](#)

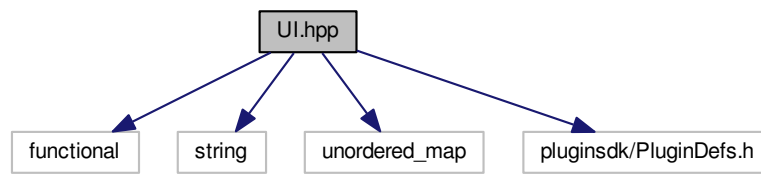
Constant Groups

- [dcapi](#)

6.27 UI.hpp File Reference

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

Include dependency graph for UI.hpp:



Classes

- class [dcapi::UI](#)

Namespaces

- [dcapi](#)

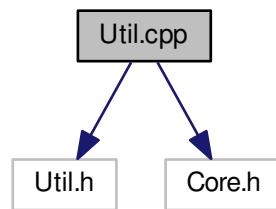
Constant Groups

- [dcapi](#)

6.28 Util.cpp File Reference

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


Include dependency graph for Util.cpp:



Namespaces

- [dcapi](#)

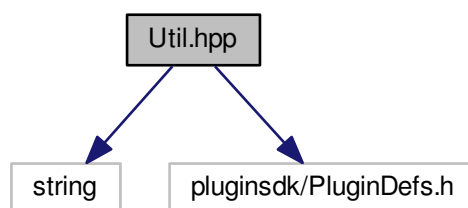
Constant Groups

- [dcapi](#)

6.29 Util.hpp File Reference

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

Include dependency graph for Util.hpp:



Classes

- class [dcapi::Util](#)

Namespaces

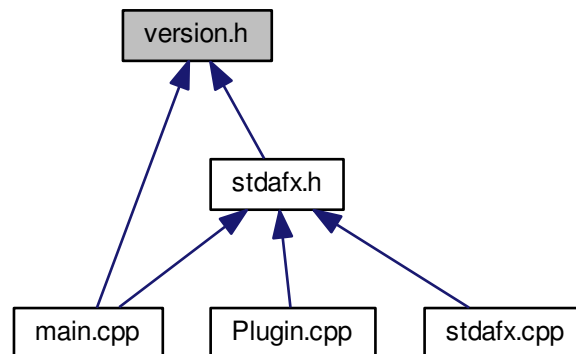
- [dcapi](#)

Constant Groups

- [dcapi](#)

6.30 version.h File Reference

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



Macros

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

6.30.1 Macro Definition Documentation

6.30.1.1 `#define PLUGIN_AUTHOR "..."`

Definition at line 33 of file version.h.

Referenced by pluginInit().

6.30.1.2 `#define PLUGIN_DESC "..."`

Definition at line 36 of file version.h.

Referenced by pluginInit().

6.30.1.3 #define PLUGIN_GUID "..."

Definition at line 27 of file version.h.

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

6.30.1.4 #define PLUGIN_NAME "..."

Definition at line 30 of file version.h.

Referenced by pluginInit().

6.30.1.5 #define PLUGIN_VERSION_MAJOR 1

Definition at line 39 of file version.h.

6.30.1.6 #define PLUGIN_VERSION_MINOR 0

Definition at line 40 of file version.h.

6.30.1.7 #define PLUGIN_WEB "N/A"

Definition at line 43 of file version.h.

Referenced by pluginInit().

Index

- ~Plugin
 - Plugin, [35](#)
- addCommand
 - dcapi::UI, [77](#)
- addEvent
 - dcapi::Hooks, [22](#)
- apiVersion
 - DCUI, [21](#)
 - tagDCCConfig, [48](#)
 - tagDCCConnection, [49](#)
 - tagDCCore, [51](#)
 - tagDCHooks, [52](#)
 - tagDCHub, [54](#)
 - tagDCInterface, [55](#)
 - tagDCLog, [56](#)
 - tagDCQueue, [58](#)
 - tagDCTagger, [60](#)
 - tagDCUtils, [62](#)
 - tagMetaData, [67](#)
- appName
 - dcapi::Core, [19](#)
- attributes
 - tagDCTagger, [60](#)
- author
 - tagMetaData, [67](#)
- bBreak
 - PluginDefs.h, [106](#)
- Bool
 - PluginDefs.h, [101](#), [111](#)
 - tagDCCore, [50](#)
 - tagDCHooks, [52](#)
 - tagDCHub, [54](#)
 - tagDCQueue, [57](#)
- CFG_TYPE_BOOL
 - PluginDefs.h, [109](#)
- CFG_TYPE_INT
 - PluginDefs.h, [109](#)
- CFG_TYPE_INT64
 - PluginDefs.h, [109](#)
- CFG_TYPE_REMOVE
 - PluginDefs.h, [109](#)
- CFG_TYPE_STRING
 - PluginDefs.h, [109](#)
- CFG_TYPE_UNKNOWN
 - PluginDefs.h, [109](#)
- Callback
 - dcapi::Hooks, [22](#)
- cid
 - tagUserData, [72](#)
- clear
 - dcapi::Hooks, [24](#)
- clearCommands
 - dcapi::UI, [77](#)
- cmd
 - tagDCCConnection, [49](#)
 - tagDCHub, [54](#)
- Command
 - dcapi::UI, [76](#)
- command
 - DCUI, [21](#)
 - tagCommandData, [40](#)
- commandCallback
 - dcapi::UI, [77](#)
- CommandData
 - PluginDefs.h, [106](#)
- CommandDataPtr
 - PluginDefs.h, [107](#)
- commands
 - dcapi::UI, [78](#)
- config
 - Config.c, [85](#)
 - dcapi::Config, [15](#)
- Config.c, [83](#)
 - config, [85](#)
 - free_core_cfg, [84](#)
 - get_cfg, [84](#)
 - get_cfg_int, [84](#)
 - get_cfg_int64, [84](#)
 - get_cfg_path, [84](#)
 - get_core_cfg, [84](#)
 - get_install_path, [84](#)
 - guid, [85](#)
 - init_cfg, [84](#)
 - set_cfg, [84](#)
 - set_cfg_int, [85](#)
 - set_cfg_int64, [85](#)
 - uninit_cfg, [85](#)
- Config.cpp, [85](#)
- Config.h, [86](#)
 - free_core_cfg, [87](#)
 - get_cfg, [87](#)
 - get_cfg_int, [87](#)
 - get_cfg_int64, [87](#)
 - get_cfg_path, [87](#)
 - get_core_cfg, [88](#)
 - get_install_path, [88](#)

- init_cfg, [88](#)
- set_cfg, [88](#)
- set_cfg_int, [88](#)
- set_cfg_int64, [88](#)
- uninit_cfg, [88](#)
- Config.hpp, [88](#)
- ConfigBool
 - PluginDefs.h, [107](#)
- ConfigBoolPtr
 - PluginDefs.h, [107](#)
- ConfigInt
 - PluginDefs.h, [107](#)
- ConfigInt64
 - PluginDefs.h, [107](#)
- ConfigInt64Ptr
 - PluginDefs.h, [107](#)
- ConfigIntPtr
 - PluginDefs.h, [107](#)
- ConfigStr
 - PluginDefs.h, [107](#)
- ConfigStrPtr
 - PluginDefs.h, [107](#)
 - tagDCConfig, [47](#)
- ConfigType
 - PluginDefs.h, [107](#)
- ConfigValue
 - PluginDefs.h, [107](#)
- ConfigValuePtr
 - PluginDefs.h, [107](#)
 - tagDCConfig, [47](#)
- ConnectionData
 - PluginDefs.h, [107](#)
- ConnectionDataPtr
 - PluginDefs.h, [107](#)
- connections
 - dcapi::Connections, [17](#)
- Connections.cpp, [89](#)
- Connections.hpp, [90](#)
- core
 - dcapi::Core, [19](#)
 - PluginDefs.h, [107](#)
- Core.cpp, [90](#)
- Core.hpp, [91](#)
- DCAPI
 - PluginDefs.h, [101](#)
- DCAPI_CORE_VER
 - PluginDefs.h, [101](#)
- DCConfig
 - PluginDefs.h, [107](#)
- DCConfigPtr
 - PluginDefs.h, [107](#)
- DCConnection
 - PluginDefs.h, [107](#)
- DCConnectionPtr
 - PluginDefs.h, [107](#)
- DCCore
 - PluginDefs.h, [107](#)
- DCCorePtr
 - PluginDefs.h, [107](#)
- PluginDefs.h, [107](#)
- DCEXP
 - PluginDefs.h, [101](#)
- DCHooks
 - PluginDefs.h, [107](#)
- DCHooksPtr
 - PluginDefs.h, [107](#)
- DCHub
 - PluginDefs.h, [107](#)
- DCHubPtr
 - PluginDefs.h, [107](#)
- DCIMP
 - PluginDefs.h, [101](#)
- DCINTF_CONFIG
 - PluginDefs.h, [101](#)
- DCINTF_CONFIG_VER
 - PluginDefs.h, [101](#)
- DCINTF_DCPP_HUBS
 - PluginDefs.h, [102](#)
- DCINTF_DCPP_QUEUE
 - PluginDefs.h, [102](#)
- DCINTF_DCPP_TAGGER
 - PluginDefs.h, [102](#)
- DCINTF_DCPP_UI
 - PluginDefs.h, [102](#)
- DCINTF_DCPP_UI_VER
 - PluginDefs.h, [102](#)
- DCINTF_DCPP_UTILS
 - PluginDefs.h, [103](#)
- DCINTF_HOOKS
 - PluginDefs.h, [103](#)
- DCINTF_HOOKS_VER
 - PluginDefs.h, [103](#)
- DCINTF_LOGGING
 - PluginDefs.h, [103](#)
- DCINTF_LOGGING_VER
 - PluginDefs.h, [103](#)
- DCInterface
 - PluginDefs.h, [107](#)
- DCInterfacePtr
 - PluginDefs.h, [108](#)
 - tagDCCore, [50](#)
- DCLog
 - PluginDefs.h, [108](#)
- DCLogPtr
 - PluginDefs.h, [108](#)
- DCQueue
 - PluginDefs.h, [108](#)
- DCQueuePtr
 - PluginDefs.h, [108](#)
- DCTagger
 - PluginDefs.h, [108](#)
- DCTaggerPtr
 - PluginDefs.h, [108](#)
- DCUI, [19](#)
 - apiVersion, [21](#)
 - command, [21](#)
 - icon, [21](#)

- message, 21
- name, 21
- PluginDefs.h, 108
- void, 20
- DCUIPtr
 - PluginDefs.h, 108
- DCUtils
 - PluginDefs.h, 108
- DCUtilsPtr
 - PluginDefs.h, 108
- data
 - tagDCCConnection, 49
- dcFalse
 - PluginDefs.h, 110
- dcTrue
 - PluginDefs.h, 110
- dcBool
 - PluginDefs.h, 107
- dcapi, 7
- dcapi::Config, 10
 - config, 15
 - freeCoreConfig, 12
 - getBoolConfig, 12
 - getConfig, 12
 - getCoreConfig, 12
 - getInstallPath, 12
 - getInt64Config, 12
 - getIntConfig, 12
 - getPath, 13
 - guid, 15
 - handle, 13
 - init, 13
 - setConfig, 13, 14
- dcapi::Connections, 15
 - connections, 17
 - handle, 16
 - init, 16
- dcapi::Core, 17
 - appName, 19
 - core, 19
 - handle, 18
 - init, 19
- dcapi::Hooks, 21
 - addEvent, 22
 - Callback, 22
 - clear, 24
 - empty, 24
 - eventCallback, 25
 - events, 27
 - handle, 26
 - hooks, 27
 - init, 26, 27
 - remove, 27
- dcapi::Hooks::Chat, 9
 - onIncomingChat, 9
 - onIncomingPM, 9
 - onOutgoingChat, 10
 - onOutgoingPM, 10
- dcapi::Hooks::Hubs, 27
 - onOffline, 28
 - onOnline, 28
- dcapi::Hooks::Network, 32
 - onClientDataIn, 33
 - onClientDataOut, 33
 - onHubDataIn, 33
 - onHubDataOut, 33
 - onUDPDataIn, 34
 - onUDPDataOut, 34
- dcapi::Hooks::Queue, 36
 - onAdded, 37
 - onFinished, 37
 - onMoved, 37
 - onRemoved, 38
- dcapi::Hooks::Timer, 73
 - onMinute, 73
 - onSecond, 73
- dcapi::Hooks::UI, 74
 - onChatCommand, 74
 - onChatCommandPM, 74
 - onChatDisplay, 75
 - onChatTags, 75
 - onCreated, 75
- dcapi::Hooks::Users, 78
 - onOffline, 79
 - onOnline, 79
- dcapi::Hubs, 28
 - handle, 29
 - hubs, 30
 - init, 29, 30
- dcapi::Logger, 30
 - handle, 31
 - init, 31, 32
 - log, 32
 - logger, 32
- dcapi::Queue, 38
 - handle, 39
 - init, 39
 - queue, 40
- dcapi::Tagger, 62
 - handle, 63
 - init, 63, 64
 - tagger, 64
- dcapi::UI, 76
 - addCommand, 77
 - clearCommands, 77
 - Command, 76
 - commandCallback, 77
 - commands, 78
 - guid, 78
 - handle, 77
 - init, 77, 78
 - removeCommand, 78
 - ui, 78
- dcapi::Util, 79
 - fromUtf8, 80
 - handle, 80

- init, [80](#), [81](#)
- toUtf8, [81](#)
- utils, [81](#)
- dcptr_t
 - PluginDefs.h, [108](#)
- dctime_t
 - PluginDefs.h, [108](#)
- defProc
 - tagDCHooks, [52](#)
- dependencies
 - tagMetaData, [67](#)
- description
 - tagMetaData, [67](#)
- empty
 - dcapi::Hooks, [24](#)
- end
 - tagDCTagger, [60](#)
- eventCallback
 - dcapi::Hooks, [25](#)
- events
 - dcapi::Hooks, [27](#)
- False
 - PluginDefs.h, [103](#)
- free_core_cfg
 - Config.c, [84](#)
 - Config.h, [87](#)
- freeCoreConfig
 - dcapi::Config, [12](#)
- fromUtf8
 - dcapi::Util, [80](#)
- get_cfg
 - Config.c, [84](#)
 - Config.h, [87](#)
- get_cfg_int
 - Config.c, [84](#)
 - Config.h, [87](#)
- get_cfg_int64
 - Config.c, [84](#)
 - Config.h, [87](#)
- get_cfg_path
 - Config.c, [84](#)
 - Config.h, [87](#)
- get_core_cfg
 - Config.c, [84](#)
 - Config.h, [88](#)
- get_install_path
 - Config.c, [84](#)
 - Config.h, [88](#)
- get_text
 - tagDCTagger, [59](#)
- getBoolConfig
 - dcapi::Config, [12](#)
- getConfig
 - dcapi::Config, [12](#)
- getCoreConfig
 - dcapi::Config, [12](#)
- getInstallPath
 - dcapi::Config, [12](#)
- getInt64Config
 - dcapi::Config, [12](#)
- getIntConfig
 - dcapi::Config, [12](#)
- getPath
 - dcapi::Config, [13](#)
- graceless
 - tagDCConnection, [49](#)
- guid
 - Config.c, [85](#)
 - dcapi::Config, [15](#)
 - dcapi::UI, [78](#)
 - tagMetaData, [67](#)
- HOOK_CHAT_IN
 - PluginDefs.h, [103](#)
- HOOK_CHAT_OUT
 - PluginDefs.h, [103](#)
- HOOK_CHAT_PM_IN
 - PluginDefs.h, [104](#)
- HOOK_CHAT_PM_OUT
 - PluginDefs.h, [104](#)
- HOOK_HUB_OFFLINE
 - PluginDefs.h, [104](#)
- HOOK_HUB_ONLINE
 - PluginDefs.h, [104](#)
- HOOK_QUEUE_ADDED
 - PluginDefs.h, [105](#)
- HOOK_QUEUE_FINISHED
 - PluginDefs.h, [105](#)
- HOOK_QUEUE_MOVED
 - PluginDefs.h, [105](#)
- HOOK_QUEUE_REMOVED
 - PluginDefs.h, [105](#)
- HOOK_TIMER_MINUTE
 - PluginDefs.h, [105](#)
- HOOK_TIMER_SECOND
 - PluginDefs.h, [105](#)
- HOOK_UI_CHAT_TAGS
 - PluginDefs.h, [106](#)
- HOOK_UI_CREATED
 - PluginDefs.h, [106](#)
- HOOK_USER_OFFLINE
 - PluginDefs.h, [106](#)
- HOOK_USER_ONLINE
 - PluginDefs.h, [106](#)
- handle
 - dcapi::Config, [13](#)
 - dcapi::Connections, [16](#)
 - dcapi::Core, [18](#)
 - dcapi::Hooks, [26](#)
 - dcapi::Hubs, [29](#)
 - dcapi::Logger, [31](#)
 - dcapi::Queue, [39](#)
 - dcapi::Tagger, [63](#)
 - dcapi::UI, [77](#)
 - dcapi::Util, [80](#)

- hash
 - tagQueueData, 68
- hookHandle
 - PluginDefs.h, 108
 - tagDCHooks, 52
- hookProc
 - tagDCHooks, 52
- hooks
 - dcapi::Hooks, 27
- Hooks.cpp, 92
- Hooks.hpp, 92
- host_name
 - tagDCCore, 50
- HubData
 - PluginDefs.h, 108
- HubDataPtr
 - PluginDefs.h, 108
 - tagDCHub, 54
- hubHint
 - tagUserData, 72
- hubUrl
 - tagDCHub, 54
- hubs
 - dcapi::Hubs, 30
- Hubs.cpp, 93
- Hubs.hpp, 94
- icon
 - DCUI, 21
- id
 - tagDCTagger, 60
- in
 - tagStringData, 69
- init
 - dcapi::Config, 13
 - dcapi::Connections, 16
 - dcapi::Core, 19
 - dcapi::Hooks, 26, 27
 - dcapi::Hubs, 29, 30
 - dcapi::Logger, 31, 32
 - dcapi::Queue, 39
 - dcapi::Tagger, 63, 64
 - dcapi::UI, 77, 78
 - dcapi::Util, 80, 81
- init_cfg
 - Config.c, 84
 - Config.h, 88
- intfHandle
 - PluginDefs.h, 108
 - tagDCCore, 51
- ip
 - tagConnectionData, 46
 - tagHubData, 65
 - tagUDPData, 71
- isFileList
 - tagQueueData, 68
- isManaged
 - tagConnectionData, 46
 - tagHubData, 65
- tagQueueData, 68
- tagTagData, 70
- tagUserData, 72
- isOp
 - tagConnectionData, 46
 - tagHubData, 65
 - tagUserData, 72
- isSecure
 - tagConnectionData, 46
 - tagHubData, 65
- location
 - tagQueueData, 68
- log
 - dcapi::Logger, 32
- logger
 - dcapi::Logger, 32
- Logger.cpp, 94
- Logger.hpp, 95
- MSG_CLIENT
 - PluginDefs.h, 110
- MSG_STATUS
 - PluginDefs.h, 110
- MSG_SYSTEM
 - PluginDefs.h, 110
- main
 - Plugin, 35
- main.cpp, 95
- pluginInit, 96
- message
 - DCUI, 21
- MetaData
 - PluginDefs.h, 108
- MetaDataPtr
 - PluginDefs.h, 108
- msg
 - tagDCHub, 54
- MsgType
 - PluginDefs.h, 108
- n
 - tagDCConnection, 49
 - tagDCUtils, 62
- name
 - DCUI, 21
 - tagMetaData, 67
- nick
 - tagDCHub, 54
 - tagUserData, 72
- numDependencies
 - tagMetaData, 67
- ON_CONFIGURE
 - PluginDefs.h, 110
- ON_INSTALL
 - PluginDefs.h, 110
- ON_LOAD
 - PluginDefs.h, 110

ON_LOAD_RUNTIME
 PluginDefs.h, [110](#)

ON_UNINSTALL
 PluginDefs.h, [110](#)

ON_UNLOAD
 PluginDefs.h, [110](#)

object
 tagConnectionData, [46](#)
 tagHubData, [65](#)
 tagQueueData, [69](#)
 tagTagData, [70](#)
 tagUserData, [72](#)

onAdded
 dcapi::Hooks::Queue, [37](#)

onChatCommand
 dcapi::Hooks::UI, [74](#)

onChatCommandPM
 dcapi::Hooks::UI, [74](#)

onChatDisplay
 dcapi::Hooks::UI, [75](#)

onChatTags
 dcapi::Hooks::UI, [75](#)

onClientDataIn
 dcapi::Hooks::Network, [33](#)

onClientDataOut
 dcapi::Hooks::Network, [33](#)

onCreated
 dcapi::Hooks::UI, [75](#)

onFinished
 dcapi::Hooks::Queue, [37](#)

onHubDataIn
 dcapi::Hooks::Network, [33](#)

onHubDataOut
 dcapi::Hooks::Network, [33](#)

onIncomingChat
 dcapi::Hooks::Chat, [9](#)

onIncomingPM
 dcapi::Hooks::Chat, [9](#)

onLoad
 Plugin, [36](#)

onMinute
 dcapi::Hooks::Timer, [73](#)

onMoved
 dcapi::Hooks::Queue, [37](#)

onOffline
 dcapi::Hooks::Hubs, [28](#)
 dcapi::Hooks::Users, [79](#)

onOnline
 dcapi::Hooks::Hubs, [28](#)
 dcapi::Hooks::Users, [79](#)

onOutgoingChat
 dcapi::Hooks::Chat, [10](#)

onOutgoingPM
 dcapi::Hooks::Chat, [10](#)

onRemoved
 dcapi::Hooks::Queue, [38](#)

onSecond
 dcapi::Hooks::Timer, [73](#)

onUDPDataIn
 dcapi::Hooks::Network, [34](#)

onUDPDataOut
 dcapi::Hooks::Network, [34](#)

out
 tagStringData, [69](#)

PATH_GLOBAL_CONFIG
 PluginDefs.h, [110](#)

PATH_LOCALE
 PluginDefs.h, [110](#)

PATH_RESOURCES
 PluginDefs.h, [110](#)

PATH_USER_CONFIG
 PluginDefs.h, [110](#)

PATH_USER_LOCAL
 PluginDefs.h, [110](#)

PRIO_DEFAULT
 PluginDefs.h, [111](#)

PRIO_HIGH
 PluginDefs.h, [111](#)

PRIO_HIGHEST
 PluginDefs.h, [111](#)

PRIO_LOW
 PluginDefs.h, [111](#)

PRIO_LOWEST
 PluginDefs.h, [111](#)

PRIO_NORMAL
 PluginDefs.h, [111](#)

PROTOCOL_ADC
 PluginDefs.h, [110](#)

PROTOCOL_DHT
 PluginDefs.h, [110](#)

PROTOCOL_NMDC
 PluginDefs.h, [110](#)

pCommon
 PluginDefs.h, [108](#)
 tagDCHooks, [52](#)

pData
 PluginDefs.h, [109](#)
 tagDCHooks, [52](#)

pInterface
 tagDCCore, [51](#)

PLUGIN_AUTHOR
 version.h, [118](#)

PLUGIN_DESC
 version.h, [118](#)

PLUGIN_GUID
 version.h, [118](#)

PLUGIN_NAME
 version.h, [119](#)

PLUGIN_VERSION
 stdafx.h, [113](#)

PLUGIN_WEB
 version.h, [119](#)

pObject
 tagDCHooks, [53](#)

params
 tagCommandData, [40](#)

password
 tagDCHub, 54
 PathType
 PluginDefs.h, 108
 Plugin, 34
 ~Plugin, 35
 main, 35
 onLoad, 36
 Plugin, 35
 Plugin.cpp, 96
 Plugin.h, 97
 PluginDefs.h
 CFG_TYPE_BOOL, 109
 CFG_TYPE_INT, 109
 CFG_TYPE_INT64, 109
 CFG_TYPE_REMOVE, 109
 CFG_TYPE_STRING, 109
 CFG_TYPE_UNKNOWN, 109
 dcFalse, 110
 dcTrue, 110
 MSG_CLIENT, 110
 MSG_STATUS, 110
 MSG_SYSTEM, 110
 ON_CONFIGURE, 110
 ON_INSTALL, 110
 ON_LOAD, 110
 ON_LOAD_RUNTIME, 110
 ON_UNINSTALL, 110
 ON_UNLOAD, 110
 PATH_GLOBAL_CONFIG, 110
 PATH_LOCALE, 110
 PATH_RESOURCES, 110
 PATH_USER_CONFIG, 110
 PATH_USER_LOCAL, 110
 PRIO_DEFAULT, 111
 PRIO_HIGH, 111
 PRIO_HIGHEST, 111
 PRIO_LOW, 111
 PRIO_LOWEST, 111
 PRIO_NORMAL, 111
 PROTOCOL_ADC, 110
 PROTOCOL_DHT, 110
 PROTOCOL_NMDC, 110
 PluginDefs.h, 97
 bBreak, 106
 Bool, 101, 111
 CommandData, 106
 CommandDataPtr, 107
 ConfigBool, 107
 ConfigBoolPtr, 107
 ConfigInt, 107
 ConfigInt64, 107
 ConfigInt64Ptr, 107
 ConfigIntPtr, 107
 ConfigStr, 107
 ConfigStrPtr, 107
 ConfigType, 107
 ConfigValue, 107
 ConfigValuePtr, 107
 ConnectionData, 107
 ConnectionDataPtr, 107
 core, 107
 DCAPI, 101
 DCAPI_CORE_VER, 101
 DCConfig, 107
 DCConfigPtr, 107
 DCConnection, 107
 DCConnectionPtr, 107
 DCCore, 107
 DCCorePtr, 107
 DCEXP, 101
 DCHooks, 107
 DCHooksPtr, 107
 DCHub, 107
 DCHubPtr, 107
 DCIMP, 101
 DCINTF_CONFIG, 101
 DCINTF_CONFIG_VER, 101
 DCINTF_DCPP_HUBS, 102
 DCINTF_DCPP_QUEUE, 102
 DCINTF_DCPP_TAGGER, 102
 DCINTF_DCPP_UI, 102
 DCINTF_DCPP_UTILS, 103
 DCINTF_HOOKS, 103
 DCINTF_HOOKS_VER, 103
 DCINTF_LOGGING, 103
 DCINTF_LOGGING_VER, 103
 DCInterface, 107
 DCInterfacePtr, 108
 DCLog, 108
 DCLogPtr, 108
 DCQueue, 108
 DCQueuePtr, 108
 DCTagger, 108
 DCTaggerPtr, 108
 DCUI, 108
 DCUIPtr, 108
 DCUtils, 108
 DCUtilsPtr, 108
 dcBool, 107
 dcptr_t, 108
 dctime_t, 108
 False, 103
 HOOK_CHAT_IN, 103
 HOOK_CHAT_OUT, 103
 HOOK_CHAT_PM_IN, 104
 HOOK_CHAT_PM_OUT, 104
 HOOK_HUB_OFFLINE, 104
 HOOK_HUB_ONLINE, 104
 HOOK_QUEUE_ADDED, 105
 HOOK_QUEUE_MOVED, 105
 HOOK_QUEUE_REMOVED, 105
 HOOK_TIMER_MINUTE, 105
 HOOK_TIMER_SECOND, 105
 HOOK_UI_CHAT_TAGS, 106
 HOOK_UI_CREATED, 106

- HOOK_USER_OFFLINE, [106](#)
- HOOK_USER_ONLINE, [106](#)
- hookHandle, [108](#)
- HubData, [108](#)
- HubDataPtr, [108](#)
- intfHandle, [108](#)
- MetaData, [108](#)
- MetaDataPtr, [108](#)
- MsgType, [108](#)
- pCommon, [108](#)
- pData, [109](#)
- PathType, [108](#)
- PluginState, [109](#)
- ProtocolType, [109](#)
- QueueData, [109](#)
- QueueDataPtr, [109](#)
- QueuePrio, [109](#)
- StringData, [109](#)
- StringDataPtr, [109](#)
- subsHandle, [109](#)
- tagConfigType, [109](#)
- tagDCBool, [109](#)
- TagData, [109](#)
- TagDataPtr, [109](#)
- tagMsgType, [110](#)
- tagPathType, [110](#)
- tagPluginState, [110](#)
- tagProtocolType, [110](#)
- tagQueuePrio, [110](#)
- True, [106](#)
- UDPData, [109](#)
- UDPDataPtr, [109](#)
- UserData, [109](#)
- UserDataPtr, [109](#)
- void, [111](#)
- pluginInit
 - main.cpp, [96](#)
- PluginState
 - PluginDefs.h, [109](#)
- port
 - tagConnectionData, [46](#)
 - tagDCCConnection, [49](#)
 - tagHubData, [65](#)
 - tagUDPData, [71](#)
- priority
 - tagDCQueue, [58](#)
- protocol
 - tagConnectionData, [46](#)
 - tagHubData, [66](#)
 - tagUserData, [73](#)
- ProtocolType
 - PluginDefs.h, [109](#)
- queue
 - dcapi::Queue, [40](#)
- Queue.cpp, [111](#)
- Queue.hpp, [112](#)
- QueueData
 - PluginDefs.h, [109](#)
- QueueDataPtr
 - PluginDefs.h, [109](#)
- QueuePrio
 - PluginDefs.h, [109](#)
- remove
 - dcapi::Hooks, [27](#)
- removeCommand
 - dcapi::UI, [78](#)
- replacement
 - tagDCTagger, [60](#)
- resource.h, [112](#)
- set_cfg
 - Config.c, [84](#)
 - Config.h, [88](#)
- set_cfg_int
 - Config.c, [85](#)
 - Config.h, [88](#)
- set_cfg_int64
 - Config.c, [85](#)
 - Config.h, [88](#)
- setConfig
 - dcapi::Config, [13](#), [14](#)
- setting
 - tagDCConfig, [48](#)
- sid
 - tagUserData, [73](#)
- silent
 - tagDCQueue, [58](#)
- size
 - tagDCQueue, [58](#)
 - tagQueueData, [69](#)
- size_t
 - tagDCHooks, [52](#)
 - tagDCUtils, [62](#)
- src
 - tagDCUtils, [62](#)
- start
 - tagDCTagger, [60](#)
- stdafx.cpp, [112](#)
- stdafx.h, [113](#)
- PLUGIN_VERSION, [113](#)
- tstring, [114](#)
- StringData
 - PluginDefs.h, [109](#)
- StringDataPtr
 - PluginDefs.h, [109](#)
- subsHandle
 - PluginDefs.h, [109](#)
 - tagDCHooks, [52](#)
- tagCommandData, [40](#)
 - command, [40](#)
 - params, [40](#)
- tagConfigBool, [41](#)
 - type, [41](#)
 - value, [41](#)

- tagConfigInt, 41
 - type, 42
 - value, 42
- tagConfigInt64, 42
 - type, 43
 - value, 43
- tagConfigStr, 43
 - type, 44
 - value, 44
- tagConfigType
 - PluginDefs.h, 109
- tagConfigValue, 44
 - type, 45
- tagConnectionData, 45
 - ip, 46
 - isManaged, 46
 - isOp, 46
 - isSecure, 46
 - object, 46
 - port, 46
 - protocol, 46
- tagDCBool
 - PluginDefs.h, 109
- tagDCConfig, 46
 - apiVersion, 48
 - ConfigStrPtr, 47
 - ConfigValuePtr, 47
 - setting, 48
 - type, 48
 - val, 48
 - void, 48
- tagDCConnection, 48
 - apiVersion, 49
 - cmd, 49
 - data, 49
 - graceless, 49
 - n, 49
 - port, 49
 - UserDataPtr, 49
 - void, 49
- tagDCCore, 50
 - apiVersion, 51
 - Bool, 50
 - DCInterfacePtr, 50
 - host_name, 50
 - intfHandle, 51
 - plInterface, 51
 - version, 51
- tagDCHooks, 51
 - apiVersion, 52
 - Bool, 52
 - defProc, 52
 - hookHandle, 52
 - hookProc, 52
 - pCommon, 52
 - pData, 52
 - pObject, 53
 - size_t, 52
 - subsHandle, 52
- tagDCHub, 53
 - apiVersion, 54
 - Bool, 54
 - cmd, 54
 - HubDataPtr, 54
 - hubUrl, 54
 - msg, 54
 - nick, 54
 - password, 54
 - thirdPerson, 55
 - type, 55
 - UserDataPtr, 54
 - void, 54
- tagDCInterface, 55
 - apiVersion, 55
- tagDCLog, 55
 - apiVersion, 56
 - void, 56
- tagDCQueue, 56
 - apiVersion, 58
 - Bool, 57
 - priority, 58
 - QueueDataPtr, 57
 - silent, 58
 - size, 58
 - target, 58
 - void, 57, 58
- tagDCTagger, 58
 - apiVersion, 60
 - attributes, 60
 - end, 60
 - get_text, 59
 - id, 60
 - replacement, 60
 - start, 60
 - void, 59
- tagDCUtils, 60
 - apiVersion, 62
 - n, 62
 - size_t, 62
 - src, 62
- TagData
 - PluginDefs.h, 109
- TagDataPtr
 - PluginDefs.h, 109
- tagHubData, 64
 - ip, 65
 - isManaged, 65
 - isOp, 65
 - isSecure, 65
 - object, 65
 - port, 65
 - protocol, 66
 - url, 66
- tagMetaData, 66
 - apiVersion, 67
 - author, 67

- dependencies, [67](#)
- description, [67](#)
- guid, [67](#)
- name, [67](#)
- numDependencies, [67](#)
- version, [67](#)
- web, [67](#)
- tagMsgType
 - PluginDefs.h, [110](#)
- tagPathType
 - PluginDefs.h, [110](#)
- tagPluginState
 - PluginDefs.h, [110](#)
- tagProtocolType
 - PluginDefs.h, [110](#)
- tagQueueData, [68](#)
 - hash, [68](#)
 - isFileList, [68](#)
 - isManaged, [68](#)
 - location, [68](#)
 - object, [69](#)
 - size, [69](#)
 - target, [69](#)
- tagQueuePrio
 - PluginDefs.h, [110](#)
- tagStringData, [69](#)
 - in, [69](#)
 - out, [69](#)
- tagTagData, [70](#)
 - isManaged, [70](#)
 - object, [70](#)
- tagUDPData, [70](#)
 - ip, [71](#)
 - port, [71](#)
- tagUserData, [71](#)
 - cid, [72](#)
 - hubHint, [72](#)
 - isManaged, [72](#)
 - isOp, [72](#)
 - nick, [72](#)
 - object, [72](#)
 - protocol, [73](#)
 - sid, [73](#)
- tagger
 - dcapi::Tagger, [64](#)
- Tagger.cpp, [114](#)
- Tagger.hpp, [115](#)
- target
 - tagDCQueue, [58](#)
 - tagQueueData, [69](#)
- thirdPerson
 - tagDCHub, [55](#)
- toUtf8
 - dcapi::Util, [81](#)
- True
 - PluginDefs.h, [106](#)
- tstring
 - stdafx.h, [114](#)
- type
 - tagConfigBool, [41](#)
 - tagConfigInt, [42](#)
 - tagConfigInt64, [43](#)
 - tagConfigStr, [44](#)
 - tagConfigValue, [45](#)
 - tagDCConfig, [48](#)
 - tagDCHub, [55](#)
- UDPData
 - PluginDefs.h, [109](#)
- UDPDataPtr
 - PluginDefs.h, [109](#)
- UI.cpp, [115](#)
- UI.hpp, [116](#)
- ui
 - dcapi::UI, [78](#)
- uninit_cfg
 - Config.c, [85](#)
 - Config.h, [88](#)
- url
 - tagHubData, [66](#)
- UserData
 - PluginDefs.h, [109](#)
- UserDataPtr
 - PluginDefs.h, [109](#)
 - tagDCConnection, [49](#)
 - tagDCHub, [54](#)
- Util.cpp, [116](#)
- Util.hpp, [117](#)
- utils
 - dcapi::Util, [81](#)
- val
 - tagDCConfig, [48](#)
- value
 - tagConfigBool, [41](#)
 - tagConfigInt, [42](#)
 - tagConfigInt64, [43](#)
 - tagConfigStr, [44](#)
- version
 - tagDCCore, [51](#)
 - tagMetaData, [67](#)
- version.h, [118](#)
 - PLUGIN_AUTHOR, [118](#)
 - PLUGIN_DESC, [118](#)
 - PLUGIN_GUID, [118](#)
 - PLUGIN_NAME, [119](#)
 - PLUGIN_WEB, [119](#)
- void
 - DCUI, [20](#)
 - PluginDefs.h, [111](#)
 - tagDCConfig, [48](#)
 - tagDCConnection, [49](#)
 - tagDCHub, [54](#)
 - tagDCLog, [56](#)
 - tagDCQueue, [57](#), [58](#)
 - tagDCTagger, [59](#)

web

tagMetaData, [67](#)