XmppTask Class

A specialization of the Task class that acts as a base class for asynchronous XMPP tasks. Classes derived from **XmppTask** can act as stanza senders, stanza listeners, or both. They can also perform additional processing, such as listening for presence stanzas, translating them, and sending translated notifications out to registered listeners.

When an incoming stanza arrives from the network, the parent object (**XmppClient**) sends the stanza to each child object to see whether it can handle this stanza by calling its **HandleStanza** method. If the object can handle the stanza, it returns true, and **XmppClient** discards the stanza; otherwise, **XmppClient** tries the next child in its list. Child objects are queried in order of their priority. Priority is determined by a **buzz::XmppEngine::HandlerLevel** enumeration, where a lower number indicates a higher priority. An **XmppTask** object's priority is declared in its constructor.

See Sending and Receiving Stanzas for more information about using **XmppTask** objects.

If extending **XmppTask**, you *must* implement the following methods:

- **HandleStanza** This is called before **ProcessStart** with a stanza to see whether the object can handle the stanza. If the object determines that it can handle the stanza, it should call **QueueStanza** and pass in the stanza, and the task manager will then call **ProcessStart**.
- **ProcessStart** Called by the task manager asynchronously after **HandleStanza** has returned true. To get the stanza queued in HandleStanza, call **NextStanza**.

Additionally, you *might* override the following methods:

- **Stop**
- **OnDisconnect** Perform any special events in the event that the connection is broken.
- **ProcessResponse** If your subclass should be called more than once (for instance, **Receiver** continues to send stanzas to **SessionClient** throughout the lifetime of the application) it should implement this class and return an appropriate state value.
- Any other virtual methods in **XmppTask** or **Task**.

See **Task** to learn more about the base task class.

Syntax

```
class XmppTask : public Task,
                 public XmppStanzaHandler,
                 public sigslot::has_slots<>
```

Methods

Name	Description
XmppClient* **GetClient**()	Returns a handle to the **XmppClient** managing object.
bool **HandleStanza**(const XmlElement **stanza*)	Called by the parent to see whether this object can handle a new incoming stanza. Return true if you can, false if you cannot. If you return true, XmppClient will not send the stanza to any other handlers.

bool **MatchRequestIq**(const XmlElement **stanza*, const std::string &*type*, const QName &*qn*)	Verifies the name, type, and presence of an element.
bool **MatchResponseIq**(const XmlElement **stanza*, const Jid &*to*, const std::string &*task_id*)	Determines whether an incoming stanza is a reply for a previously sent stanza by matching the to, from, and ID fields.
bool **MatchResponseIqFromArbitrary**(const XmlElement **stanza*, const Jid &*to*, const std::string &*task_id*)	Determines whether an incoming stanza matches a previously sent stanza by ID only.
XmlElement* **NextStanza**()	Retrieves the next stanza added to the queue by **QueueStanza**.
void **OnDisconnect**()	Called by XmppClient when the connection is disconnected. Causes an error to bubble up.
int **ProcessReponse**()	You should not need to call this. Called by the task manager on subsequent calls to the object after the initial step. Should return a Task::State value indicating the state of the task after attempting to process data.
void **QueueStanza**(const XmlElement **stanza*)	Sends the stanza to an internal queue to be handled asynchronously. Call this from **HandleStanza** if the object determines that it can handle this stanza.
XmppReturnStatus **SendStanza**(const XmlElement **stanza*)	Called by an external class to send a stanza out over the network.
XmppReturnStatus **SendStanzaError**(const XmlElement **element_original*, XmppStanzaError *code*, const std::string &*text*)	Called by an external class to send out an error code in response to an erroneous stanza.
void **set_task_id**(std::string *id*)	Called internally to set the ID to that of the next stanza queued by XmppClient.
void **Stop**()	Internal method to loop through all queried stanzas.
void **StopImpl**()	Internal method to loop through all queued stanzas.
std::string **task_id**()	Called internally to retrieve the ID to that of the next stanza queued by **XmppClient**.
XmppTask(Task **parent*, XmppEngine::HandlerLevel *level*=XmppEngine::HL_NONE)	Constructor with the following parameters: *parent* The parent object, typically **XmppClient**. *level* The priority level of this object, where objects with a lower enumeration value have a higher priority and are given an earlier opportunity to handle the stanza.
~XmppTask	Destructor.

Friends

- XmppClient

Attributes: public

Declaration file: libjingle-0.3.0/talk/xmpp/xmpptask.h

Last updated March 23, 2012.