

Telepathy Echo Protocol Example

Maksim Melnikau (max_posedon)

Linux Mobile hobbyist
World of Tanks developer

January 26, 2013

Let's start

- ▶ python
- ▶ gobject
- ▶ telepathy
- ▶ telepathy-python

telepathy-foo

```
DBusGMainLoop(set_as_default=True)  
FooConnectionManager()  
MainLoop().run()
```

Properties

- ▶ EnglishName
- ▶ Parameters
- ▶ VCardField
- ▶ Icon

class Protocol

foo/protocol.py

```
class Protocol:

    _proto = PROTOCOL
    _english_name = PROTOCOL.capitalize()
    _icon = "im-%s" % PROTOCOL
    _vcard_field = "im-%s" % PROTOCOL
    _mandatory_parameters = {'account': 's'}

    def create_connection(self,
        connection_manager, parameters):
        return FooConnection(self,
            connection_manager, parameters)
```

org.freedesktop.Telepathy.ConnectionManager

Methods

- ▶ GetParameters
- ▶ ListProtocols
- ▶ RequestConnection

Signals

- ▶ NewConnections

Properties

- ▶ Protocols
- ▶ Interfaceds

class FooConnectionManager

foo/connection_manager.py

```
class FooConnectionManager:  
  
    def __init__(self):  
        self._implement_protocol(PROTOCOL, FooProtocol)
```

org.freedesktop.Telepathy.Connection

Methods

- ▶ Connect
- ▶ Disconnect

Properties

- ▶ Status
- ▶ Interfaces

Signals

- ▶ StatusChanged

Methods

- ▶ GetContactAttributes

Properties

- ▶ ContactAttributeInterfaces

Methods

- ▶ CreateChannel

Properties

- ▶ Channels
- ▶ RequestableChannelClasses

class Connection

foo/connection.py

```
class FooConnection:

    def Connect(self):
        self.StatusChanged(CONNECTION_STATUS_CONNECTED,
                            CONNECTION_STATUS_REASON_REQUESTED)
    def Disconnect(self):
        self.StatusChanged(CONNECTION_STATUS_DISCONNECTED,
                            CONNECTION_STATUS_REASON_REQUESTED)

    def GetContactListAttributes(self, interfaces, hold):
        ret = Dictionary(signature='ua{sv}')
        ...
        return ret
```

Properties

- ▶ ChannelType
- ▶ Interfaces

class FooChannelManager

foo/channel_manager.py

```
class FooChannelManager:

    def __init__(self, connection, protocol):
        self.implement_channel_classes(CHANNEL_TYPE_TEXT,
                                       self._get_text_channel)

    def _get_text_channel(self, props):
        self.__text_channel_id += 1
        path = "TextChannel%d" % self.__text_channel_id
        return FooTextChannel(self._conn,
                              self, props, object_path=path)
```

org.freedesktop.Telepathy.Channel.Type.Text

Methods

- ▶ AcknowledgePendingMessages

org.freedesktop.Telepathy.Channel.Interface.Messages

Methods

- ▶ SendMessage

Signals

- ▶ MessageSent
- ▶ MessageReceived

Properties

- ▶ SupportedContentType
- ▶ PendingMessages

class FooTextChannel

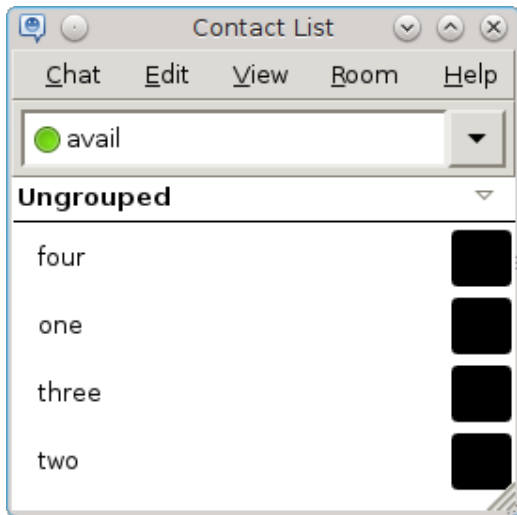
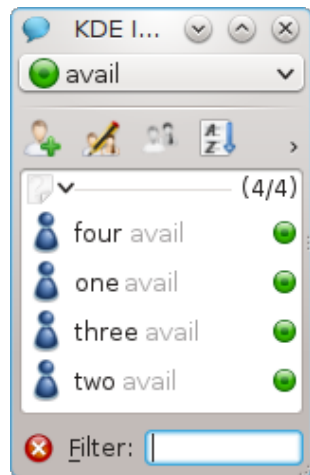
foo/channel/text.py

```
class FooTextChannel:
    def SendMessage(self, message, flags):
        gobject.timeout_add(50, self._send_message,
                            message, flags, token)

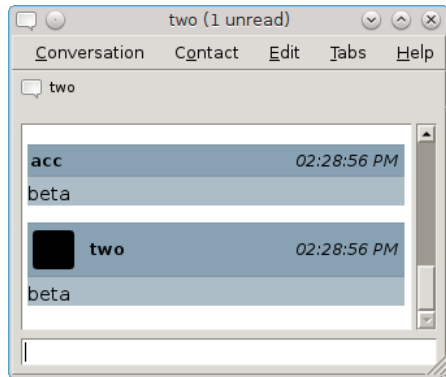
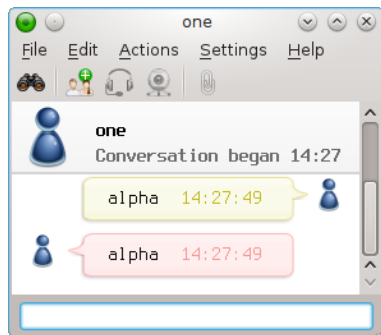
    def _send_message(self, message, flags, token):
        self.MessageSent(message, flags, token)
        gobject.timeout_add(50, self._message_received,
                            str(message[1]['content']))

    def _message_received(self, msg):
        self.MessageReceived(message)
```

Contact List



Chat



More Info

- ▶ email: maxposedon@gmail.com
- ▶ <https://github.com/max-posedon/telepathy-foo>
- ▶ <https://github.com/max-posedon/telepathy-python>
- ▶ <https://github.com/max-posedon/talk-telepathy-echo>