Subject: Re: Google Translator
Posted by mrjt on Wed, 18 Mar 2009 09:37:54 GMT
View Forum Message <> Reply to Message

Depends on how you want it to work. First you should be running the Http request in a Thread so that you can still process Gui messages.

Normally to get a progress bar on an Http request you would just pass in a Gate (the Progress class has an operator for this) that the HttpClient uses to update the progress indicator. Remember to create the Gate outside of the Http thread or it won't work properly (I think).

In your case however, I don't think the amount of data you're recieving is actually large enough to make a progress bar work properly. It will do nothing until connection is established and then update and close almost immediately. I'd suggest faking it with a timer and a looping Progress indicator. Something like:

(Add to header)
```
 Progress progress;
 String result;
 volatile Atomic translating;

 enum {
  TIMEID_PROGRESS = Ctrl::TIMEID_COUNT,
  TIMEID_COUNT
 };

 void HttpThread(String url, String post, Gate2<int, int> _progress);
 void UpdateProgress();
 bool CheckCancel(int, int);
```

Source:
```
void GoogleTranslator::TranslateText(){
 String from = inputwindow.fromlanguagectrl.GetKey(inputwindow.fromlanguagectrl.GetIndex());
 String to = inputwindow.tolanguagectrl.GetKey(inputwindow.tolanguagectrl.GetIndex());
 //... need to add proxy
 //http_client.Proxy("");
 String url = "www.google.com/translate_a/t?client=t&sl=" + from + "&tl=" + to;
 String post = String("text=") << inputwindow.textfrom.GetData();

 progress.Reset();
 progress.SetText("Contacting Google Translator");
 progress.Title("Translating");
 progress.Set(0, 50);

 AtomicWrite(translating, 1);
 SetTimeCallback(-200, THISBACK(UpdateProgress), TIMEID_PROGRESS);
 Thread().Run(THISBACK3(HttpThread, url, post, THISBACK(CheckCancel)));
```

```
 while(AtomicRead(translating)) {
  ProcessEvents();
  GuiSleep(300);
 }
 KillTimeCallback(TIMEID_PROGRESS);
 progress.Close();
 textto.Set(result, CHARSET_UTF8);
}

void GoogleTranslator::UpdateProgress()
{
 int p = progress.GetPos() + 1;
 progress.Set((p > progress.GetTotal()) ? 0 : p, progress.GetTotal());
}

bool GoogleTranslator::CheckCancel(int, int)
{
 return progress.Canceled();
}

void GoogleTranslator::HttpThread(String url, String post, Gate2<int, int> _progress)
{
 HttpClient http_client;

 http_client.URL(url);
 //http_client.Agent("Mozilla/5.0");
 http_client.TimeoutMsecs(5000);
 http_client.Post();
 http_client.PostData(post);

 result = http_client.ExecuteRedirect(HttpClient::DEFAULT_MAX_REDIRECT,
     HttpClient::DEFAULT_RETRIES, _progress);

 result = Nvl(result
  ,String('', error:')<<Nvl(http_client.GetError(), '')
  <<', status: '<<http_client.GetStatusCode()<<', '<<http_client.GetStatusLine()
  <<', header: '<<http_client.GetHeaders());
 AtomicWrite(translating, 0);
}
```