Subject: Plugin Framework
Posted by tojocky on Tue, 02 Feb 2010 08:18:04 GMT
View Forum Message <> Reply to Message

Hello All!

I found very interesting article from Google Chrome about multiprocess, and how every plugin is in separate process. In this way it is possible to make plugin more safety. it is possible to kill process safety!
Article address: http://dev.chromium.org/developers/design-documents/process- models

What is your opinion?

Regards, Ion Lupascu (tojocky)

Subject: Re: Plugin Framework
Posted by mirek on Tue, 02 Feb 2010 18:18:04 GMT
View Forum Message <> Reply to Message

tojocky wrote on Tue, 02 February 2010 03:18Hello All!

I found very interesting article from Google Chrome about multiprocess, and how every plugin is in separate process. In this way it is possible to make plugin more safety. it is possible to kill process safety!

I am 100% for this approach! Whenever I was thinking about safe plugin architecture, I have always ended with something like this.

The only problem is how to achieve effective and maintainable communication between both process. And how to use that to do GUI...

Subject: Re: Plugin Framework
Posted by tojocky on Tue, 02 Feb 2010 20:38:32 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Thank you for your opinion!
Google Chrome is an open source project! This give us possibility to understand how can google realized this and try to adapt in our project!
As you can see, Google can to do GUI and comunication between process.

In the end, I will try to study the google plugin framework technology and make simple demo in

u++!

Regards, Ion Lupascu (tojocky)

luzr wrote on Tue, 02 February 2010 20:18

I am 100% for this approach! Whenever I was thinking about safe plugin architecture, I have always ended with something like this.

The only problem is how to achieve effective and maintainable communication between both process. And how to use that to do GUI...

Subject: Re: Plugin Framework
Posted by tojocky on Wed, 03 Feb 2010 07:19:27 GMT
View Forum Message <> Reply to Message

Interesting new!

Mozilla tries to make it extensions/plugins multiprocess too to make more safety:

http://www.taranfx.com/firefox-gets-multi-process-plugins

Subject: Re: Plugin Framework
Posted by tojocky on Sun, 07 Feb 2010 20:55:27 GMT
View Forum Message <> Reply to Message

Mirek,
I found the answer of your question:
- IPC(Inter process comunication):
http://dev.chromium.org/developers/design-documents/inter-pr ocess-communication
- multi process architecture: http://dev.chromium.org/developers/design-documents/multi-pr ocess-architecture

With respect, Ion Lupascu (tojocky)
luzr wrote on Tue, 02 February 2010 20:18tojocky wrote on Tue, 02 February 2010 03:18Hello All!

I found very interesting article from Google Chrome about multiprocess, and how every plugin is in separate process. In this way it is possible to make plugin more safety. it is possible to kill process safety!

I am 100% for this approach! Whenever I was thinking about safe plugin architecture, I have

always ended with something like this.

The only problem is how to achieve effective and maintainable communication between both process. And how to use that to do GUI...

Subject: Re: Plugin Framework
Posted by mirek on Fri, 13 Aug 2010 07:22:24 GMT
View Forum Message <> Reply to Message

tojocky wrote on Sun, 07 February 2010 15:55Mirek,
I found the answer of your question:
 - IPC(Inter process comunication):
http://dev.chromium.org/developers/design-documents/inter-pr ocess-communication
 - multi process architecture: http://dev.chromium.org/developers/design-documents/multi-pr ocess-architecture

With respect, Ion Lupascu (tojocky)
luzr wrote on Tue, 02 February 2010 20:18tojocky wrote on Tue, 02 February 2010 03:18Hello All!

I found very interesting article from Google Chrome about multiprocess, and how every plugin is in separate process. In this way it is possible to make plugin more safety. it is possible to kill process safety!

I am 100% for this approach! Whenever I was thinking about safe plugin architecture, I have always ended with something like this.

The only problem is how to achieve effective and maintainable communication between both process. And how to use that to do GUI...

In the one big application I am working on now, we are using intensively using XmlRpc to communicate with backend.

I have to say that the current XmlRpc package has quite good usability even without external marshalling utilities (that are used in e.g. good old Corba). Now I am not saying that we should use XmlRpc - that one is perhaps way too ineffective, but the style with XMLRPC_METHOD is quite good and simple...

http://www.ultimatepp.org/reference$XmlRpcServer$en-us.html
http://www.ultimatepp.org/reference$XmlRpcClient$en-us.html
