

The Ivy C++ and java library guide

CENA NT02-819

Yannick Jestin

`jestin@cena.fr`

This document is a programmer's guide that describes how to use the Ivy Java library to connect applications to an Ivy bus. This guide describes version 1.2 of the library. This document itself


```
$
```

4.3. Testing

We are going to test our application with **fr.dgac.ivy.Probe**. In a terminal window, launch **ivyTranslator**.

```
$ java ivyTranslator
```

Then in another terminal window, launch **java fr.dgac.ivy.Probe '(*)'**. You are then ready to start. Type "Hello Paul", and you should get "Bonjour Paul". Type "Bye", and your application should quit:

```
$ java fr.dgac.ivy.Probe '(*)'
you want to subscribe to (*)
broadcasting on 127.255.255.255:2010
IvyTranslator connected
IvyTranslator subscribes to ^Bye$
IvyTranslator subscribes to ^Hello(.*)
IvyTranslator sent 'Hello le monde'"Helleit:
```

```
$4.966 Tf7.80.9-73281.914 T54.3.laocats[( $ )]TJ/F30 11.955 f7.80.9-34.471914 T5414.3. andrsin
```

This constructor readies the structures for the software bus connexion. It is possible to have different busses at the same time in an application, be it on the same bus or on different ivy busses. The *name* is the name of the application on the bus, and will be transmitted to other application, and possibly be used by them. The *message*

options are passed to main. Use the `bindMsg()` method to bind a callback to a pattern, and the `unbindMsg` method to delete the binding.

```
public int bindMsg(String regex, IvyMessageListener callback);  
public void unBindMsg(int id);
```

The *regex*

6.2. Direct messages

```
count = bus->
```


7.1. Win32 API8. programmer's style guide