BotPlatform SDK User Guide
Contents

BOTPLATFORM SDK USER GUIDE

CONTENTS ................................................................................................................... 1

1 SUPPORT .................................................................................................................... 3
   1.1 CONTACT US ........................................................................................................ 3
   1.2 FORUM .................................................................................................................. 3
   1.3 OTHER DOCUMENTATION .................................................................................. 3

2 ARCHITECTURE ....................................................................................................... 4

3 DEVELOPMENT FLOW ............................................................................................ 5
   3.1 REGISTER SPID AT botplatform.com .................................................................. 5
   3.2 LOG INTO BotPlatform USING YOUR SPID ......................................................... 6
   3.3 CONFIG THE ROBOT DEFAULT INFORMATION ...................................................... 7
   3.4 ADD ROBOT ACCOUNTS TO BotPlatform ................................................................. 7
   3.5 DOWNLOAD SDK AND DEVELOP YOUR RobotServer ............................................ 8
      3.5.1 Language ....................................................................................................... 8
      3.5.2 Implementation codes .................................................................................... 8
      3.5.3 Debug ............................................................................................................ 8
   3.6 DEPLOY AND START YOUR RobotServer .............................................................. 8

4 FEATURES ............................................................................................................... 9
   4.1 ROBOT PROFILE .................................................................................................... 9
   4.2 DELUXE DISPLAY PICTURE ................................................................................. 10
   4.3 CUSTOMIZED EMOTICON ................................................................................... 10
   4.4 SCENE .................................................................................................................. 11
   4.5 ACTIVITY .............................................................................................................. 11
   4.6 WINK ................................................................................................................... 12
   4.7 INK ....................................................................................................................... 12
   4.8 FILE TRANSFER ................................................................................................... 13
   4.9 VOICECLIP .......................................................................................................... 13

5 EXAMPLE CODES .................................................................................................... 14
1 Support

1.1 Contact us

You can always contact us for technique support by the following ways:

<table>
<thead>
<tr>
<th>E-MAIL</th>
<th><a href="mailto:support@botplatform.com">support@botplatform.com</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td></td>
</tr>
</tbody>
</table>

1.2 Forum

You can also post questions to our forum, and we will reply as soon as possible. URL of our forum: http://botplatform.uservoice.com/

1.3 Other documentation

BotPlatform API has been well documented in several program languages (including java, vc++, c#, vb, etc.) which can help you to find answers.
2 Architecture

As shown in Figure 1, the whole BotPlatform involves Service Providers (SP), BotPlatform Servers (BPS), IM Server and IM User client, whose relations are shown as followings:

* SPs connect to BPs to provide service;
* IM User clients connect to IM Server for IM service;
* The BotPlatform Server is the key that connects the SPs to IM User clients. You can take BotPlatform as an adapter from BotPlatform API to IM protocol API (ex: MSN, Yahoo, GTalk, etc.). So it’s transparent for the SPs to interact with IM User clients.

SPs can have their own service servers which connect to BotPlatform Server by our SDK. When one service server meets some bottle neck, SP can balance your burden by running your service in several parallel service servers; In the meantime, BotPlatform will guarantee the stability of network (both SP-BP and BP-MSN Server).

The bots (SPs) log on as IM client by BotPlatform, and will be kept online at BotPlatform. So SP should never care about this part.
3 Development Flow

3.1 Register SPID at botplatform.com

Go to the SP website (http://botplatform.com) for registration
After registration, BotPlatform will issue a SPID as the login account.

**Figure 4**

### 3.2 Log into BotPlatform using your SPID

Log into BotPlatform using your SPID and password
After you logged in, you will see the BotPlatform administration system.
The system consists of the following modules:

* Management
  - Manage your the basic information and the robot default information.
* SP Resource
  - Manage your robots’ resources such as display picture, emoticon, etc..
* User resource
  - This module is a temporary store of your friends’ resources. You can check out these resources if you want.
* Download
  - This module provide you with various version of SDKs downloading.
* Documentation
  - Useful documents for developping.
3.3 Config the robot default information

![Image of BotPlatform interface with SP information and robot information sections]

Figure 5

3.4 Add robot accounts to BotPlatform

Bot account is an IM account (ex: Windows Live account). To add bot accounts, click the "Management" -> "MSN Account Management" link in the left region of the page. You will see your bot account list in the right region of the page. After adding some accounts, you will see the "sign in" link in every account entry, you can click to sign in; After signing in successfully, the "sign out" link will appear instead of "sign in".

Then add the bot account signed in as your MSN buddy, you will see it in status "Away". If you connect your SP client to BotPlatform, the account will switch to status "Available".

![Image of BotPlatform interface with bot account management and account addition options]

Figure 6
3.5 Download SDK and develop your RobotServer

3.5.1 Language

There are four languages available: Java C# C++ VB

A. Java programs can run in Linux or window with JDK installed (version 1.5+).
B. C# C++ VB SDK is based on COM which can only run in windows. After downloading DLL of COM, you should register in windows (regsvr32 BotPlatformSDK.dll), and then import it into your project for developing.

Besides, the COM SDK fit all language based on COM.

3.5.2 Implementation codes

See Example codes and API Documentation.

3.5.3 Debug

Make sure that your client has connected to internet.

You can use any developing tools to debug your program. Once you start debugging, your client will connect BotPlatform; And you will see your bot account switch its status from "Away" to "Available"; Any action in your codes will be reflected in your conversation with the bot. If you stop debugging, the bot account will be "Away" again.

3.6 Deploy and start your RobotServer

After coding and testing, you can deploy your server program to any kind of physical server. And one physical server can hold several SP server; Or you can deploy them to several physical server (depends on your user amount and network traffic).
4 Features

The following features are available by using BotPlatform SDK:

<table>
<thead>
<tr>
<th>Text Message</th>
<th>Emoticon</th>
<th>Nudge</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>FriendlyName</td>
<td>Signature</td>
<td>Display Picture</td>
<td>Deluxe Picture</td>
</tr>
<tr>
<td>Scene</td>
<td>Buddy Online/Offline Event</td>
<td>Buddy Information Event</td>
<td>Wink</td>
</tr>
<tr>
<td>Ink</td>
<td>File Transfer</td>
<td>Voiceclip</td>
<td>Webcam</td>
</tr>
<tr>
<td>Multi-User Conversation</td>
<td>Push Message</td>
<td>Get Buddy Information</td>
<td></td>
</tr>
</tbody>
</table>

4.1 Robot Profile

After logged into the BotPlatform website, SP can manage the bot information. For example, change the friendlyName, Personal Message and display picture of your bot.

Figure 7
4.2 Deluxe Display Picture

Size should be 98x142

![Deluxe Display Picture](image)

Figure 8

4.3 Customized Emoticon

SP can customize their own emoticon, the effect is sth. like the following figure:

![Customized Emoticon](image)

Figure 9
4.4 Scene

Scene is also available in BotPlatform SDK

Figure 10

4.5 Activity

SP can use Activity to provide some enhanced presentation. It’s a simple URL, you can interact with IM Use without knowledge about Microsoft Activity SDK.

Figure 11
4.6 Wink

BotPlatform also support wink which consist of animation and sound.

![Figure 12](image)

4.7 Ink

The Ink is a feature which provide the handwriting function.

![Figure 13](image)
4.8 File Transfer

![Image of File Transfer](Figure 14)

4.9 Voiceclip

![Image of Voiceclip](Figure 15)
5 Example codes

Check the DEMO part of SDK package for more example.

Java Example:

```java
RobotServerFactory serverFactory = RobotServerFactory.getInstance();
final RobotServer server = serverFactory.createRobotServer("server.botplatform.com", 6602);
server.setReconnectedSupport(true);
server.setRobotHandler(new RobotAdapter()
{
    public void messageReceived(RobotSession session,
            RobotMessage message) {
        session.send("Hello World!");
        System.out.println(message.getString());
    }
});
server.login("you spid", "your sppwd");
Runtime.getRuntime().addShutdownHook(new Thread()
{
    public void run()
    {
        server.logout();
    }
});
```

C# Example:

```csharp
static void server_MessageReceived(IRobotSession session, IRobotMessage message)
{
    session.SendText("Hello World!");
    Console.WriteLine(message.Text);
}
static void Main(string args)
{
    RobotServerFactory serverFactory = new RobotServerFactory();
    serverFactory.Init(2);
    RobotServer server = serverFactory.CreateRobotServer("server.botplatform.com", 6602);
    server.MessageReceived += new IRobotServerEvents_MessageReceivedEventHandler(server_MessageReceived);
    server.Login("your sppid", "your sppwd", 60000);
    string cmd = null;
    while ((cmd = Console.ReadLine()) != null)
```
if (cmd.Equals("exit")) break;
server.Logout();
serverFactory.Destroy();
}

C++ Example:

```cpp
ATL_FUNC_INFO s_info_onMessageReceived = { CC_STDCALL, VT_EMPTY, 2,
{ VT_UNKNOWN, VT_UNKNOWN } }

class RobotServerEventsImpl : public IDispEventSimpleImpl<1, RobotServerEventsImpl,
&DIIDIRobotServerEvents>
{
public:

BEGIN_SINK_MAP(RobotServerEventsImpl)
    SINK_ENTRY_INFO(1, DIIDIRobotServerEvents, 3,
    &RobotServerEventsImpl::onMessageReceived, &s_info_onMessageReceived)
END_SINK_MAP()

HRESULT stdcall onMessageReceived( IRobotSession session, IRobotMessage message )
{
session->SendText("Hello World!"); std::cout << message->Text << std::endl; return S_OK; };

int tmain(int argc, TCHAR argv) {
    CoInitializeEx(NULL, COINIT_MULTITHREADED);
    { CComPtr<IRobotServerFactory> spRobotServerFactory;
        spRobotServerFactory.CoCreateInstance( CLSID_RobotServerFactory, NULL,
        CLSCTX_INPROC ); spRobotServerFactory->Init(2 );

    IRobotServerPtr spRobotServer =
        spRobotServerFactory->CreateRobotServer("server.botplatform.com", 6602 );

    RobotServerEventsImpl eventImpl;

    eventImpl.DispEventAdvise( spRobotServer );

    spRobotServer->Login("your spid", "your sppwd", 60000 );

    std::string cmd;

    while ( true )
    {
        std::cin >> cmd;
    }
```
if (cmd == "exit")
    break;
}

spRobotServer->Logout();

spRobotServerFactory->Destroy();

CoUninitialize();

return 0;

}

VB Example:

Sub MessageReceived(ByVal session As IRobotSession, ByVal message As IRobotMessage)
    session.SendText("Hello World!")
    Console.WriteLine(message.Text)
End Sub

<MTAThread()> Sub Main()
    Dim robotServerFactory As RobotServerFactory = New RobotServerFactory
    robotServerFactory.Init(2)


    AddHandler robotServer.MessageReceived, AddressOf MessageReceived
    robotServer.Login("your spid", "your sppwd", 60000)

    Dim cmd As String = ""

    Do While cmd <> "exit"
cmd = Console.ReadLine()

Loop

robotServer.Logout()

robotServerFactory.Destroy()

End Sub