

# Integration von Asterisk und .net

Presentation for Astricon 2008

by

Frank Gorgas-Waller

CEO

Explido Software USA Inc.

[www.asteritest.com](http://www.asteritest.com)

# History of Asterisk.NET

- ☎ Asterisk.NET was originally created through porting from asterisk-java
- ☎ At that time hosted on GotDotNet.com (Microsoft's Code Sample site)
- ☎ As of July 2007 Microsoft abandoned GotDotNet.com to promote their new Vista flavored framework “.NET 3.0”.
- ☎ Code is hosted now hosted on [www.asteritest.com](http://www.asteritest.com)

# Why would you want to use it

- ☎ “Easy” Integration with Windows Applications
- ☎ Human Resources  
(Developer too scared to touch Linux)
- ☎ You like the adventure
- ☎ You can get it to run on mono

# Why you might not want to use it

- ☎ Mostly runs on Windows Servers  
(even so technically it could run on a Windows workstation, check your EULA...)
- ☎ You are likely to end up with source code that is not necessarily cross platform portable.
- ☎ Some of the design that resulted from the conversion of asterisk-java is neither performance optimized nor good style.

# Why else you might not want to use it

- ☎ Very little active development since there are other options available now
- ☎ Needs changes each time even minimal changes are done to the AMI interface
- ☎ The current maintainer is heavily overworked with ViciDial and is looking for someone to continue this project
- ☎ Paid support through me is very expensive

## Two paths to follow

### Asterisk Manager Interface (AMI)

In general allows it to handle connections from outside of a call. Including establishing of new connections, independent transfers,...

### Asterisk Gateway Interface (AGI)

Allows you to handle a call from the inside. Best option for Interactive Voice Response (IVR) and similar applications.

# Summoning the deemons

(starting as a service)

- ☎ For both types of Operation it is advisable to start your Asterisk.NET application as a daemon/service.
- ☎ If you have to integrate AMI into your application make sure to use AstManProxy on older versions of Asterisk.
- ☎ Or better create a service on a server to which the clients can connect.

# Using AMI with Asterisk.NET

## First step: we need a connection

```
try
{
    astManager = new AsteriskManager(server, port, username, password);
}
// could not create manager object, probably bad values
catch (Exception ex)
{
    MessageBox.Show(
        "Unable to create Manager object.\n\n" + ex.ToString(),
        "Connection Error",
        MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation
    );
    return;
}
```

## Second step: lets login

```
// try and login with the manager object
try
{
    astManager.Login();
}
// connection failed
catch (Exception ex)
{
    Disconnect();
    MessageBox.Show(
        "Unable to connect to Asterisk server.\n\n" + ex.ToString(),
        "Connection Error",
        MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation
    );
    return;
}
```

## Third step: Register for Events

```
// function to register which events were listening for
private void RegisterEvents()
{
    astManager.Events +=
        new ManagerConnectionEventHandler(AllManagerEvents);

    astManager.DisconnectEvent +=
        new DisconnectEventHandler(GotDisconnect);

    astManager.OriginateFailure +=
        new OriginateFailureEventHandler(OriginateFailure);
}
```

## Forth step: Make a Call

```
// function to actually originate a call
public void OriginateCall(string number, string extension)
{
    // first check that we are connected
    if (IsConnected())
    {
        Originate oc = new Originate();
        string numberToDial = this.textBoxNumberToDial.Text;

        numberToDial = StripNonNumerics(numberToDial);

        oc.Channel = ORIGINATE_CHANNEL + numberToDial;
        oc.Context = ORIGINATE_CONTEXT;

        oc.CallerId = this.textBoxExtension.Text;
        oc.Exten = this.textBoxExtension.Text;
        oc.Timeout = ORIGINATE_TIMEOUT;
        oc.Priority = ORIGINATE_PRIORITY;

        // oc.Variable = "VAR1=abc|VAR2=def";
        // oc.SetVariable("VAR3", "ghi");

        // Better bring some time this call will wait:
        Call call = astManager.OriginateCall(oc);

        ...
    }
}
```

# Alternative step: Here comes the Action

```
// function to actually originate a call
public void OriginateCall(string number, string extension)
{
    // first check that we are connected
    if (IsConnected())
    {
        OriginateAction oa = new OriginateAction();
        string numberToDial = this.textBoxNumberToDial.Text;

        numberToDial = StripNonNumerics(numberToDial);

        oa.Channel = ORIGINATE_CHANNEL + numberToDial;
        oa.Context = ORIGINATE_CONTEXT;

        oa.CallerId = this.textBoxExtension.Text;
        oa.Exten = this.textBoxExtension.Text;
        oa.Timeout = ORIGINATE_TIMEOUT;
        oa.Priority = ORIGINATE_PRIORITY;

        // oa.Variable = "VAR1=abc|VAR2=def";
        // oa.SetVariable("VAR3", "ghi");

        astManager.OriginateCall(oa);

        ...
    }
}
```

## Alternative step 2: Now catch the Call

```
public void OriginateSuccess(object sender,
Asterisk.NET.Manager.Event.OriginateSuccessEvent e)
{
    MessageBox.Show(
        "Origination succeeded.\n\n" + e.ToString(),
        "Originate Success",
        MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation
    );
    Call call = new createOriginateCall((Event.OriginateEvent) (e));

    ...
}
```

## Next step: Work with a Call

```
Call call = new createOriginateCall((Event.OriginateEvent) (e));  
  
call.StartTime  
call.EndTime  
  
call.channel CallerId  
call.channel CallerIdName  
  
...
```

# Pitfalls

- ☎ Code reflection
- ☎ X Locks between Windows application & Linux application
- ☎ Library is not as well maintained as it should be

# Demonstration

 Events

 Source code

 Debug output

# Thank you for attending

 You can download this presentation at  
[Asteritest.org](http://Asteritest.org)

 If you want to maintain [Asterisk.net](http://Asterisk.net) please  
contact me