



asoft

Growing Together

CHỨC NĂNG IN TRÊN WEB GUIDELINE

Người tạo: Lê Minh Lâm

Theo dõi sửa đổi

[illegible]

TÀI LIỆU MÔ TẢ RELEASE GUIDELINE

---oOo---

| | | |
|------|---|-------------------------------------|
| I) | Giới thiệu | 4 |
| 1) | Tổng quan | 4 |
| 2) | Coding guideline | Error! Bookmark not defined. |
| 3) | Môi trường phát triển | 4 |
| II) | Coding guideline | 4 |
| III) | Chạy chương trình ứng dụng in | 13 |
| | Bước 1: Start Server (Ví dụ: localhost:3327) | 13 |
| | Bước 2: Thiết lập máy in cho phù hợp với nghiệp vụ - Phải test máy in trước khi chạy chương trình | 13 |
| | Bước 3: Copy thông tin IP của máy In, lưu vào thông tin ShopID của cửa hàng | 13 |
| | Bước 4: Lưu để xác nhận thông tin cài đặt, chương trình sẽ chạy trong process của task Management | 14 |
| IV) | Thực hiện in ấn trên web browser | 14 |
| | Bước 1: Kiểm tra client đã được kết nối hay chưa? | 14 |
| | Bước 2: Lấy dữ liệu để thực hiện xuất file excel – Lý do phải lấy trước vì khi start phần in thì không thể lấy các thông tin môi trường để xử lý | 14 |
| | Bước 3: Lưu file excel kiểu FileStreamResult để chờ client gọi lệnh in | 14 |
| | Bước 4: Thực hiện kết nối với Client và gọi hàm In | 14 |
| | Bước 5: Client sẽ tiến hành download file excel về client và thực hiện hàm in..... | 15 |
| | Bước 6: Hàm sau khi download file thành công sẽ in | 15 |
| | Bước 7: Kết thúc In | 16 |
| V) | Một số chú ý nếu không kết nối được máy in | 16 |
| VI) | Cách tạo cài đặt ClickOne cho phần mềm AsoftClientPrinter | 16 |
| | Bước 1: Mở source code phần mềm AsoftClientPrinter | 16 |
| | Bước 2: Chọn Solution → Properties → Publish | 16 |
| | Bước 3: Chọn Publish Now để release ClickOne | 17 |
| | Bước 4: Chép thư mục publish vào thư mục download đã thiết định Installation Folder URL (Download/AsoftPrinter) | 17 |
| | Bước 5: Cấu hình không kiểm tra file App.Config | 17 |
| | Bước 6: Thay đổi thông tin phần mềm | 18 |

I) Giới thiệu

1) Tổng quan

- ❖ Tài liệu hướng dẫn các bước thực hiện In ấn trên trình duyệt

2) Môi trường phát triển

| | |
|-------------------------------------|--|
| Phát triển trên các trình duyệt web |  |
| Server | IIS 6 trở lên |
| Net Framework | 4.5 |
| Database | SQL2012 R2 |
| Tool hỗ trợ release | VS2012 |
| Phần mềm In | ASOFTClientPrinter |
| Luồn di chuyển phần in |  Print.xlsx |

II) Coding guideline

Nguyên lý hoạt động

Do vấn đề bảo mật, script (Javascript) trên trình duyệt không thể truy xuất các API mức thấp của hệ thống, cụ thể là không thể dùng script trên trình duyệt để điều khiển máy in, chỉ có thể điều khiển máy in bằng một ứng dụng desktop.

Vấn đề có thể được giải quyết bằng cách cài vào máy của người dùng một ứng dụng desktop, viết bằng C#.NET (Winform). Chương trình này sẽ trực tiếp truy xuất và điều khiển các máy in kết nối với máy tính của người dùng.

Chương trình in (WinClient), sẽ chờ lệnh in từ webserver của ERP9.

Người dùng sử dụng ERP 9 trên trình duyệt web sẽ ra lệnh in, Javascript nhận lệnh in này, và gửi lệnh in lên webserver thông qua một ajax request. Server gửi lệnh in này đến WinClient .

Vì cơ chế in như vậy yêu cầu sự tương tác 2 chiều giữa webserver và client, nên cần phải sử dụng web socket (<http://en.wikipedia.org/wiki/WebSocket>).

Trên nền .NET Framework, một thư viện hỗ trợ websocket được xem là tốt nhất hiện nay, đó là SignalR. SignalR hỗ trợ tương tác theo giao thức http giữa web server và client (cả desktop sử dụng .NET và web sử dụng Javascript). Ngoài websocket, SignalR còn hỗ trợ nhiều kiểu giao tiếp 2 chiều khác như long polling, forever frame ...

SignalR thực hiện tất cả các tác vụ phức tạp liên quan đến kết nối và gửi các thông điệp.

Hiện tại, hai chu trình chính cần quan tâm của chương trình in là:

Chu trình kết nối, lưu giữ kết nối:

- Webserver (IIS) được khởi động và đang chạy
- Một chương trình WinClient ở một máy tính nào đó được khởi động (máy tính này phải được nối mạng để có thể truy cập được webserver của ERP 9)
- WinClient gửi một request kết nối đến webserver, sau đó gửi thêm một request bao gồm các thông tin để xác định "danh tính" của WinClient, giúp phân biệt WinClient này với các WinClient khác.
- Server sẽ tìm đối tượng "cũ" trong WinClient ObjectList, nếu không tìm thấy (tức đây là một client mới), server sẽ khởi tạo một đối tượng WinClient Object từ các thông tin trong request, lưu đối tượng này vào danh sách WinClient ObjectList. Nếu đây là một client "cũ" gửi yêu cầu để "kết nối lại", thì server sẽ cập nhật connectionID của client tìm được.

Chu trình thực hiện in:

- Người dùng sử dụng ERP 9 trên web, ra lệnh in.
- Javascript trên trình duyệt nhận sự kiện in, tạo ajax request với các thông tin cần thiết và gửi lên server
- Webserver nhận request này, dựa vào các thông tin để xử lý nghiệp vụ nếu có.
- Webserver tìm WinClient mục tiêu, trong danh sách WinClient ObjectList.
- Webserver gửi thông điệp in đến WinClient tìm được. Thông điệp in này có chứa url để WinClient download tập tin excel, in ra máy in.
- WinClient nhận được lệnh in. Sử dụng url nhận được, WinClient gửi request download tập tin excel, sau đó in file excel này ra máy in.

Một số vấn đề về kỹ thuật code với SignalR

Dùng Nuget để tải thư viện SignalR

Dùng Nuget để tải thư viện SignalR cho project ASP.NET MVC trên webserver và project Window Form của WinClient. Nên dùng cùng phiên bản và nên dùng phiên bản mới nhất (<http://signalr.net/>).

Trong webserver, cần có một (hoặc nhiều lớp) “**Server Hub**”, kế thừa từ lớp Hub trong thư viện của SignalR (`using Microsoft.AspNet.SignalR.Hubs`)

Các lớp Server Hub có thể được đặt trong project MVC hoặc trong một project khác có reference. Hub đóng vai trò trung gian liên lạc giữa các client (cả client web và client desktop) và server và với các client khác.

Trong project MVC phải có một lớp **Startup** với nội dung:

```
using Owin;
using Microsoft.Owin;
[assembly: OwinStartup(typeof(ASOFT.ERP.Startup))]
namespace ASOFT.ERP
{
    public class Startup
    {
        public void Configuration(IAppBuilder app)
        {
            // Any connection or hub wire up and configuration should go here
            app.MapSignalR();
        }
    }
}
```

Khi webserver khởi động, lớp này sẽ tìm tất cả các lớp “Server Hub” để chuẩn bị nhận các kết nối của client thông qua SignalR. Tương tự như ASP.NET MVC tìm tất cả các Controller.

Hub.Clients và HubProxy

Ở server, Hub của SignalR cung cấp một thuộc tính tên là “Clients” để tìm và tương tác với các client, thông qua các lệnh gọi hàm “dynamic”. Hub.Clients có thể tương tác với 1 client riêng lẻ thông qua connection id, hoặc một nhóm client, thông qua group.

Ở client (viết bằng .NET C#), SignalR cung cấp lớp HubConnection để thực hiện kết nối với server, và HubProxy để đóng vai trò làm “đại diện” cho Server Hub ở client.

HubProxy ở client thực hiện 2 chức năng chính là “chờ” lệnh từ server, và “gửi” yêu cầu lên server.

Server Hub thực hiện gọi hàm của client

Ở client

Dùng hàm On của HubProxy để gán hàm xử lý sự kiện - event handler (tương tự như việc gán hàm xử lý sự kiện của Javascript). Cú pháp:

```
HubProxy.On<object, object>(eventName, (para1, para2) => onCommand(para1, para2));
```

Với

eventName: string -> tên sự kiện cần xử lý

onCommand: delegate -> delegate với danh sách tham số giống như danh sách kiểu của hàm On

Hàm On là hàm generic, danh sách kiểu dữ liệu truyền vào <> chính là kiểu của danh sách tham số trong event handler.

Để đơn giản hóa hàm On, lớp ServerCommunicator trong project Window Form cung cấp hàm gán sự kiện SetEventListenerOn với cú pháp:

```
server.SetEventListenerOn("tryPrint", Print);
```

Với

server: một instance của ServerCommunicator

Print: là method với 1 tham số

```
public void Print(object printOptions){}
```

Các hàm SetEventListenerOn khác:

```
SetEventListenerOn2
SetEventListenerOn3
SetEventListenerOn4
```

Dùng cho các method với 2, 3, 4 tham số, vd:

```
server.SetEventListenerOn3("cancelPrint", cancelPrint);
```

cancelPrint là method có 3 tham số, ví dụ:

```
public void cancelPrint(object i, object am, object fine){}
```

Sau khi gán hàm xử lý sự kiện, client sẽ có thể "chờ lệnh" từ server.

Ở server.

Với thuộc tính Clients của lớp Hub, ta thực hiện lệnh "gọi hàm" như sau

```
Clients.Client(WinClient.ConnectionID).tryPrint(parameters);
```

Với

Clients: thuộc tính của lớp Hub

WinClient : một đối tượng WinClient được tạo ra lúc client kết nối lần đầu đến server
Clients.Client(WinClient.ConnectionID) -> lấy ra đối tượng "dynamic" đại diện cho client.

Với đối tượng dynamic này, ta có thể gọi hàm với tên bất kỳ, truyền vào tham số bất kỳ. Tuy nhiên, phải gọi đúng tên hàm mà client đang "chờ", và phải truyền vào danh sách tham số phù hợp.

Client thực hiện gọi một hàm trong Server Hub

Ở Server

Server không cần "chờ" client theo kiểu gán hàm xử lý sự kiện. Chỉ cần viết các public method vào lớp Server Hub, và SignalR sẽ lo phần còn lại. Tương tự như việc tạo Controller và Action của ASP.NET MVC để xử lý request từ trình duyệt. Ví dụ, để chờ client gọi hàm có tên là WinClientConnect, ta khai báo hàm sau, trong lớp Server Hub.

```
public void WinClientConnect(Dictionary<string, object> parameters){}
```

Tham số được truyền vào dưới dạng một Dictionary, để có thể chứa nhiều thông tin trong 1 tham số duy nhất.

Ở Client

HubProxy cung cấp hàm Invoke để client có thể gọi một hàm trong ServerHub, cú pháp đơn giản như sau:

```
HubProxy.Invoke(methodName, para1, para2, ...);
```

Lớp ServerCommunicator cung cấp hàm thay thế:

```
public void CallServerHubMethod(string methodName, params object[] parameterList)
{
    HubProxy.Invoke(methodName, parameterList);
}
```

Với:

methodName: là tên của phương thức cần gọi trong Server Hub

parameterList: danh sách tham số truyền dưới dạng params []

Để tiện lợi và thống nhất, phương thức trong Server Hub sẽ nhận 1 tham số kiểu dictionary, client sẽ gọi hàm và truyền vào 1 tham số kiểu dictionary.

Về cơ bản, dùng hàm Invoke cũng giống như dùng Javascript gửi một ajax request lên server. Tuy nhiên hàm Invoke không hỗ trợ callback, tức là client sẽ không biết được server có nhận được request hay không, có thực hiện request thành công hay không.

Mặc dù vậy, ta vẫn có thể thực hiện cơ chế callback, bằng cách gán hàm xử lý sự kiện cho client, và server sẽ gọi hàm này, sau khi xử lý xong "invoke" của client.

Ví dụ:

Ở client

```
server.SetEventListenerOn("callAfterDone", afterDone_Handler);
server.CallServerHubMethod("requestInfo", data)
```

Ở server

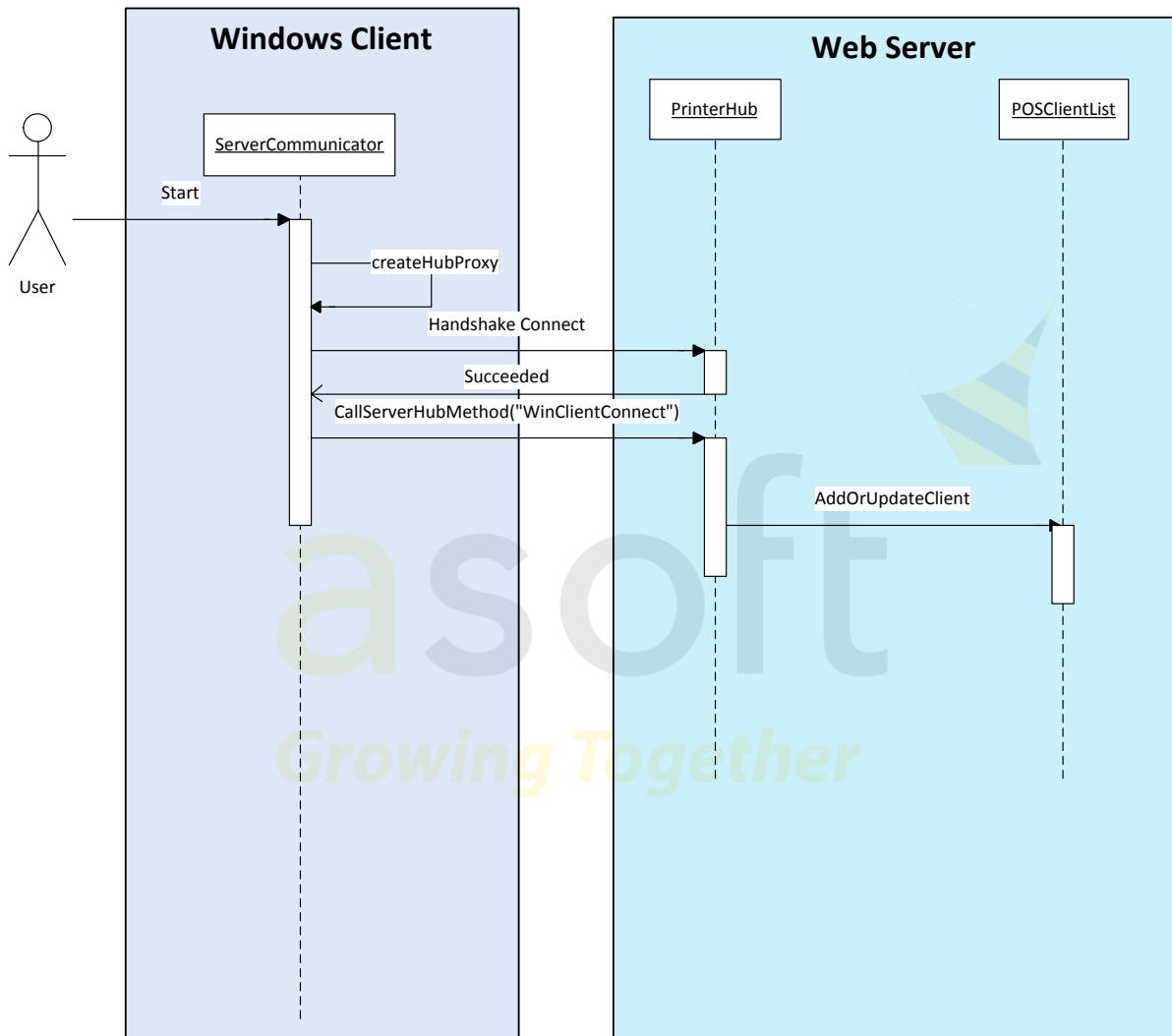
```
public void RequestInfo (Dictionary<string, object> parameters){
    // Process stuffs
    // call the callback of client
    Clients.Caller.callAfterDone(something);
}
```

Trên đây là tóm tắt một số vấn đề liên quan đến việc sử dụng SignalR trong chức năng in. Tài liệu đầy đủ về SignalR có trên <http://www.asp.net/signalr/overview/signalr-20>

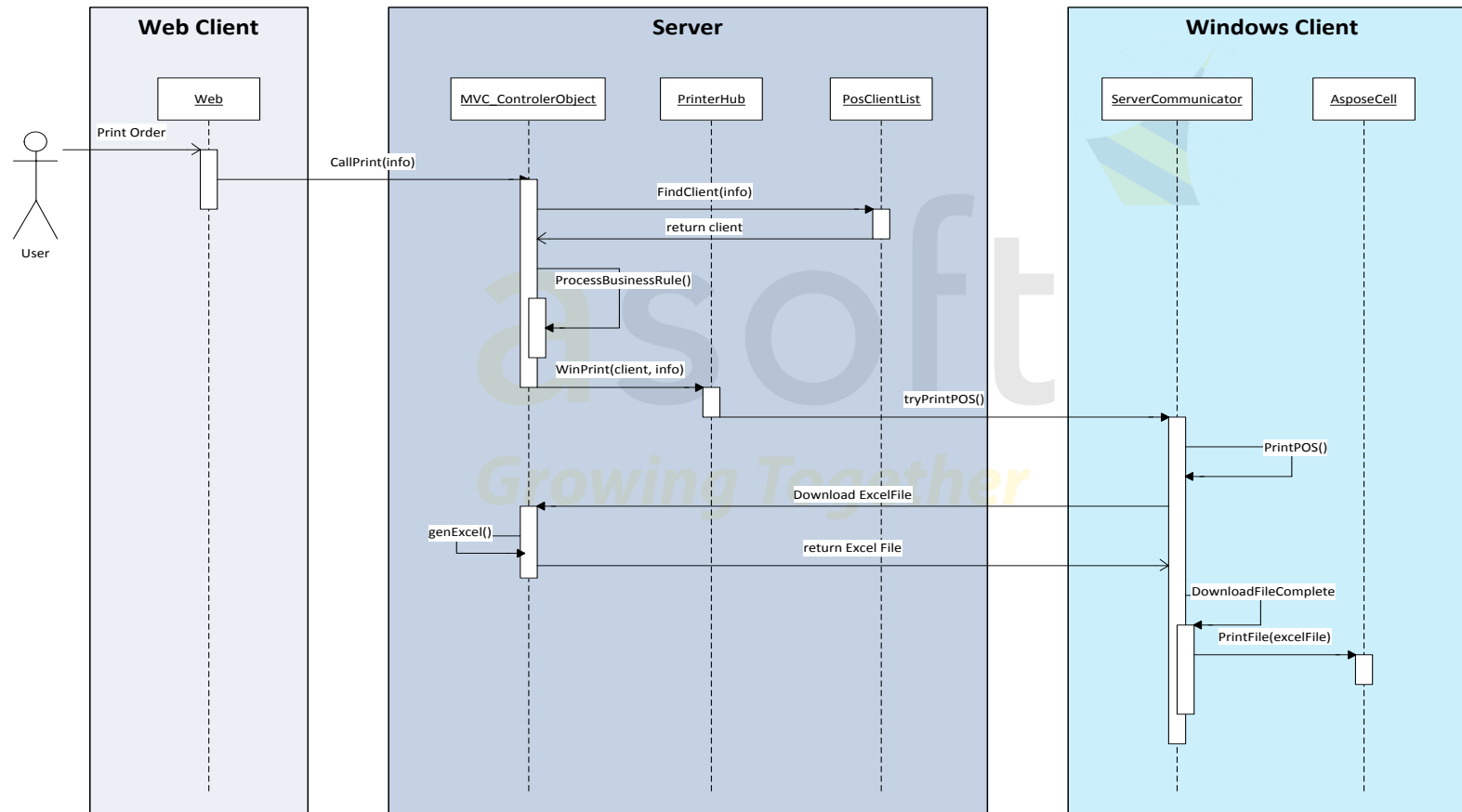
Các sơ đồ thể hiện luồng xử lý in:

Trích từ tập tin: ASoft_ERP90_PrintTechGuideLine_PrintingSequences.vsd

Connect and Persist connection



Printing



asoft 
Growing Together

III) Chạy chương trình ứng dụng in

Bước 1: Start Server (Ví dụ: localhost:3327)

- Download phần mềm tại URL public
- Ví dụ: <http://118.69.183.172:8882/Download/AsoftPrinter/setup.exe>
- Phần mềm được chứa trong thư mục download/AsoftPrinter của web Release
- Mở ứng dụng AsoftPrintClient, cấu hình Server

Bước 2: Thiết lập máy in cho phù hợp với nghiệp vụ - Phải test máy in trước khi chạy chương trình

Bước 3: Copy thông tin IP của máy In, lưu vào thông tin ShopID của cửa hàng

CẬP NHẬT CỬA HÀNG - POSF00101

| THÔNG TIN CỬA HÀNG | THÔNG TIN HÀNG HÓA | THÔNG TIN NGƯỜI DÙNG | THÔNG TIN HÓA ĐƠN | THÔNG TIN PHIẾU CHỨNG TỪ |
|---------------------------|--|---|-------------------|--------------------------|
| Phân loại hàng hóa | 24 Chọn | | | |
| Phân loại đồ ăn | Thức ăn | | | |
| Phân loại thức uống | Thức uống | | | |
| Internet IP | 14.161.5.64;118.69.183.172 | | | |
| Bán theo cột giá | <input checked="" type="checkbox"/> Cột giá | Giá bán 01 | | |
| Bán theo bảng giá | <input checked="" type="checkbox"/> Bảng giá | Chọn loại bảng giá <input type="radio"/> Bảng giá chung <input checked="" type="radio"/> Bảng giá theo thời điểm | | |
| Khuyến mãi hàng tặng hàng | <input checked="" type="checkbox"/> Áp dụng KM | | | |
| Khuyến mãi theo hóa đơn | <input type="checkbox"/> Áp dụng | | | |
| Chọn Kho* | Kho CH 63 MTB | | | |

Trường hợp nhiều IP thì cách nhau bằng dấu ;

Lưu Đóng

Bước 4: Lưu để xác nhận thông tin cài đặt, chương trình sẽ chạy trong process của task Management

IV) Thực hiện in ấn trên web browser

Bước 1: Kiểm tra client đã được kết nối hay chưa?

```
//IPPrint là IP của ứng dụng thiết lập ban đầu (118.169.183.172)

POSWinClient WinClient = POSClient.GetWinClient(shopInfo.IPPrinter)
if (WinClient != null) //Máy in có kết nối
{
    POSF00392 post00392 = new POSF0039BL().GetReportInfo(viewModel.TempAPKMaster);
    List<POST0034> listPost0034s =
    new POST0034BL().GetByApkMaster(viewModel.TempAPKMaster); //Lấy danh sách dữ liệu để in

    if (listPost0034s.Count > 0)
    {
        string url = string.Format(@"POS/POSF0039/PrintWithHub"); //URL download file excel
        FileStreamResult excel = CreateBillExcel(post00392, listPost0034s, shopInfo);
        Print(WinClient, url, excel, POSSettings.PRINTER_DEFAULT3);
        //Thiết lập máy in nào để in
        isWinConnect = true;
    }
}
```

Bước 2: Lấy dữ liệu để thực hiện xuất file excel – Lý do phải lấy trước vì khi start phần in thì không thể lấy các thông tin môi trường để xử lý

Bước 3: Lưu file excel kiểu FileStreamResult để chờ client gọi lệnh in

```
///<summary>
/// Call print
///</summary>
///<param name="WinClient"></param>
///<param name="url"></param>
///<param name="data"></param>
///<param name="printerName"></param>
private void Print(POSWinClient WinClient, string url, object data, string printerName)
{
    //Lưu FileStreamResult vào data của WinClient
    WinClient.Data = data;
    POSF0040Hub.WinPrint(WinClient, POSSettings.ShopInfo.IPPrinter, url, printerName);
}
```

Bước 4: Thực hiện kết nối với Client và gọi hàm In

```
///<summary>
/// Printer
///</summary>
///<param name="WinClient"></param>
///<param name="shopAPK"></param>
///<param name="url"></param>
///<param name="printerName"></param>
public static void WinPrint(POSWinClient WinClient, string shopAPK, string url, string printerName)
{
    var hubContext = GlobalHost.ConnectionManager.GetHubContext<POSF0040Hub>();
    var clients = hubContext.Clients;
    clients.Client(WinClient.ConnectionID).tryPrintPOS(shopAPK, url, printerName);
}
```

Hàm **tryPrintPOS** là hàm dưới ứng dụng ASOFTPrintClient - Program.cs

```
//Print máy POS
server.SetEventListenerOn3("tryPrintPOS", PrintPOS );
```

Bước 5: Client sẽ tiến hành download file excel về client và thực hiện hàm in

```
private static void PrintPOS(object ip, object url, object printerName)
{
    try
    {
        System.Configuration.Configuration config =
        ConfigurationManager.OpenExeConfiguration(ConfigurationUserLevel.None);
        if (config.AppSettings.Settings[(string)printerName] != null)
        {
            printerPOS = config.AppSettings.Settings[(string)printerName].Value;
        }
        //Download file excel
        filePath = server.GetServerFile(
            url.ToString(),
            ip.ToString(),
            POSClient_DownloadFileCompleted
            // Sau khi download file thành công thì thực hiện in
        );
    }
}
```

Bước 6: Hàm sau khi download file thành công sẽ in

```
/// <param name="isPOSPrinter"></param>
private void PrintByAspose(string filePath, string printerName, bool isPOSPrinter)
{
    //Instantiate a workbook.
    //Open an Excel file.
    Workbook workbook = new Workbook(filePath);
    workbook.CalculateFormula();
    //Define a worksheet.
    Worksheet worksheet;
    //Get the second sheet.
    worksheet = workbook.Worksheets[0];
    ImageOrPrintOptions options = new ImageOrPrintOptions();

    if (isPOSPrinter)
    {
        //Zoom file excel lại để cho vừa khổ giấy máy POS
        worksheet.PageSetup.Zoom = 75;
        worksheet.PageSetup.PaperSize = PaperSizeType.PaperJISB6;
        options.OnePagePerSheet = true;
    }
    options.PrintWithStatusDialog = false;

    //options.AllColumnsInOnePagePerSheet = true;
    SheetRender sr = new SheetRender(worksheet, options);
    try
    {
        sr.ToPrinter(printerName);
    }
    catch (Exception)
    {
        if (File.Exists(filePath))
        {
            File.Delete(filePath);
        }
    }
}
```

Máy POS chỉ hỗ trợ khổ giấy 62mm, trong khi Aspose không hỗ trợ khổ giấy này nên phải lấy khổ nhỏ nhất (PaperJISB6) và zoom lại 75%

Bước 7: Kết thúc In

V) Một số chú ý nếu không kết nối được máy in

Phần mềm AsoftClientPrinter khởi động trước server (localhost:3227)

- Địa chỉ IP giữa server và Client chưa trùng với nhau
- Có một máy khác chạy chương trình AsoftClientPrinter
- Trong Web.Config của server chưa có thẻ Json


```
<dependentAssembly>
  <assemblyIdentity name="Newtonsoft.Json" publicKeyToken="30ad4fe6b2a6aeed" culture="neutral" />
  <bindingRedirect oldVersion="0.0.0.0-6.0.0.0" newVersion="6.0.0.0" />
</dependentAssembly>
```
- Nếu server restart IIS thì phải chạy lại chương trình ASOFTPrintClient

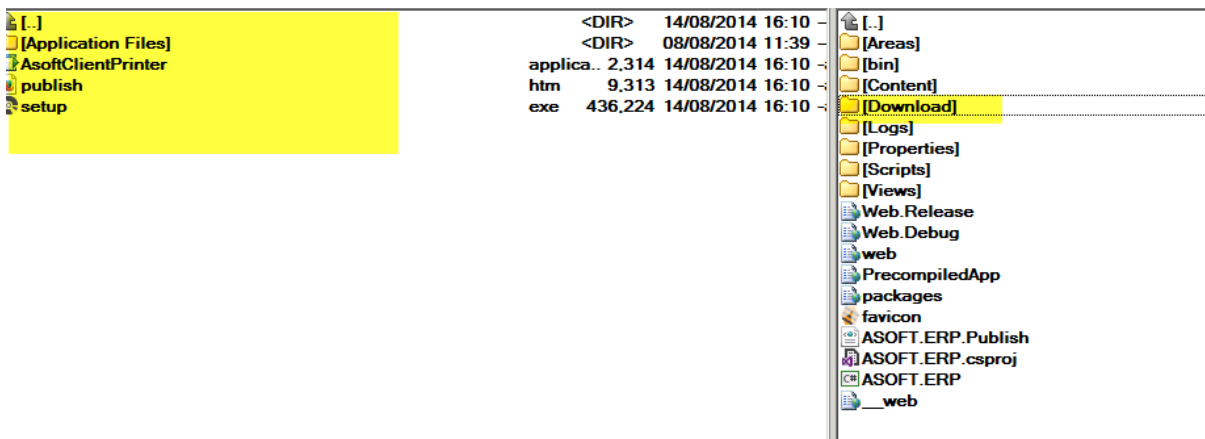
VI) Cách tạo cài đặt ClickOne cho phần mềm AsoftClientPrinter

- Bước 1:** Mở source code phần mềm AsoftClientPrinter
- Bước 2:** Chọn Solution → Properties → Publish

- **Publishing Folder Location** : Thư mục sẽ chứa source sau khi release ClickOne
- **Installation Folder URL** : URL mà user sẽ download phần mềm cũng như nếu phát sinh version mới thì sẽ dựa vào URL này để check version
- **Publish Version**: Nếu tăng version này lên thì máy Client sẽ tự động báo là cần phải update version khi Client restart lại ứng dụng

Bước 3: Chọn **Publish Now** để release ClickOne

Bước 4: Chép thư mục publish vào thư mục download đã thiết định **Installation Folder URL (Download/AsoftPrinter)**



Bước 5: **Cấu hình không kiểm tra file App.Config**

- Cơ chế **ClickOne** sẽ kiểm tra tất cả các thông tin nào thay đổi để báo khác version, do **App.Config** thường xuyên thay đổi vì chứa thông tin kết nối nên cần phải cấu hình không cho ClickOne kiểm tra file này, bằng cách:

Install Mode and Settings

☐ The application is available online only

☒ The application is available offline as well (launchable from Start menu)

Application Files...

Prerequisites...

Updates...

Options...

Publish Version

Application Files

| File Name | Publish Status | Download Group | Hash |
|--------------------------------------|-----------------------|-------------------|----------------|
| AsoftClientPrinter.exe | Include (Auto) | (Required) | Include |
| AsoftClientPrinter.exe.config | Include (Auto) | (Required) | Exclude |
| AsoftClientPrinter.exe.manifest | Include (Auto) | (Required) | Include |
| Aspose.Cells.dll | Include (Auto) | (Required) | Include |
| favicon.ico | Include (Auto) | (Required) | Include |
| Microsoft.AspNet.SignalR.Client.dll | Include (Auto) | (Required) | Include |
| Newtonsoft.Json.dll | Include (Auto) | (Required) | Include |

Chosen Exclude

☐ Show all files

Reset All

OK Cancel

Bước 6: Thay đổi thông tin phần mềm

Vào Option → Thay đổi thông tin

Install Mode and Settings

☐ The application is available online only

☒ The application is available offline as well (launchable from Start menu)

Application Files...

Prerequisites...

Updates...

Options...

Publish Version

Publish Options [?] [X]

| | |
|-------------------------|---|
| Description | Publish language: |
| Deployment | (Default) [v] |
| Manifests | Publisher name: |
| File Associations | Asoft Printer |
| | Suite name: |
| | Asoft Printer |
| | Product name: |
| | Asoft Printer |
| | Support URL: |
| | http://www.asoft.com.vn [v] ... |
| | Error URL: |
| | http://www.asoft.com.vn [v] ... |
| OK Cancel | |

asoft
Growing Together