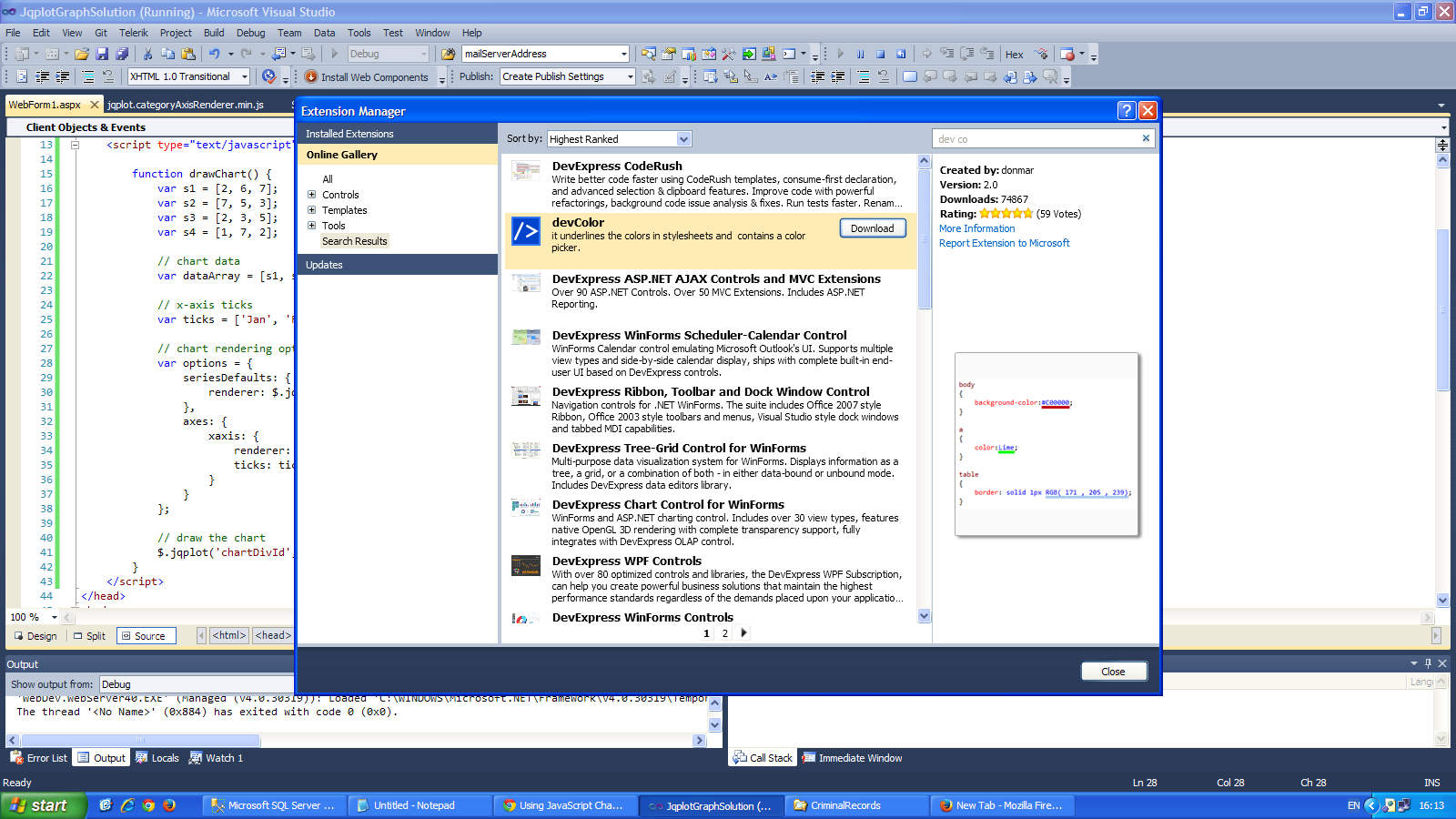
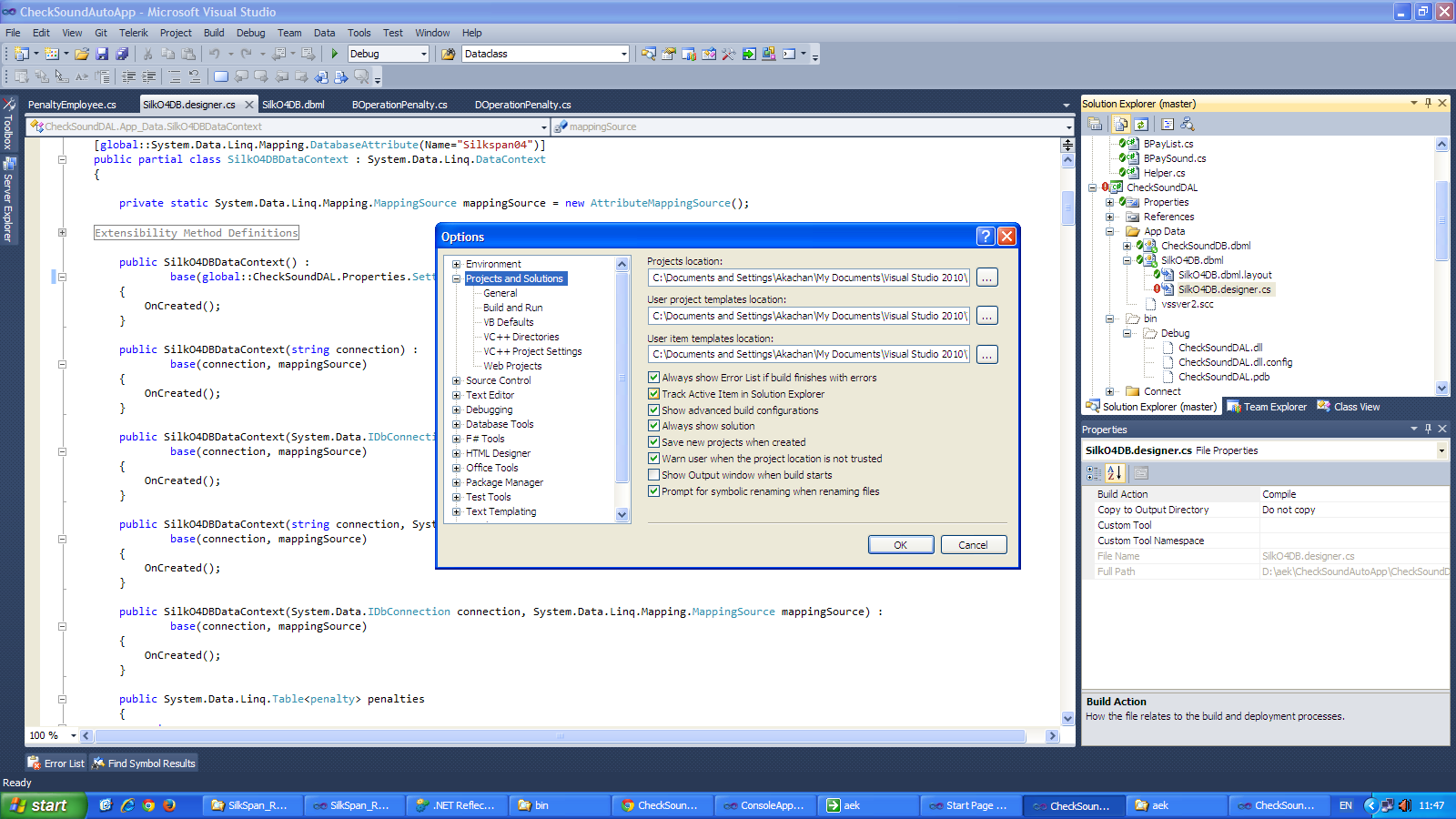
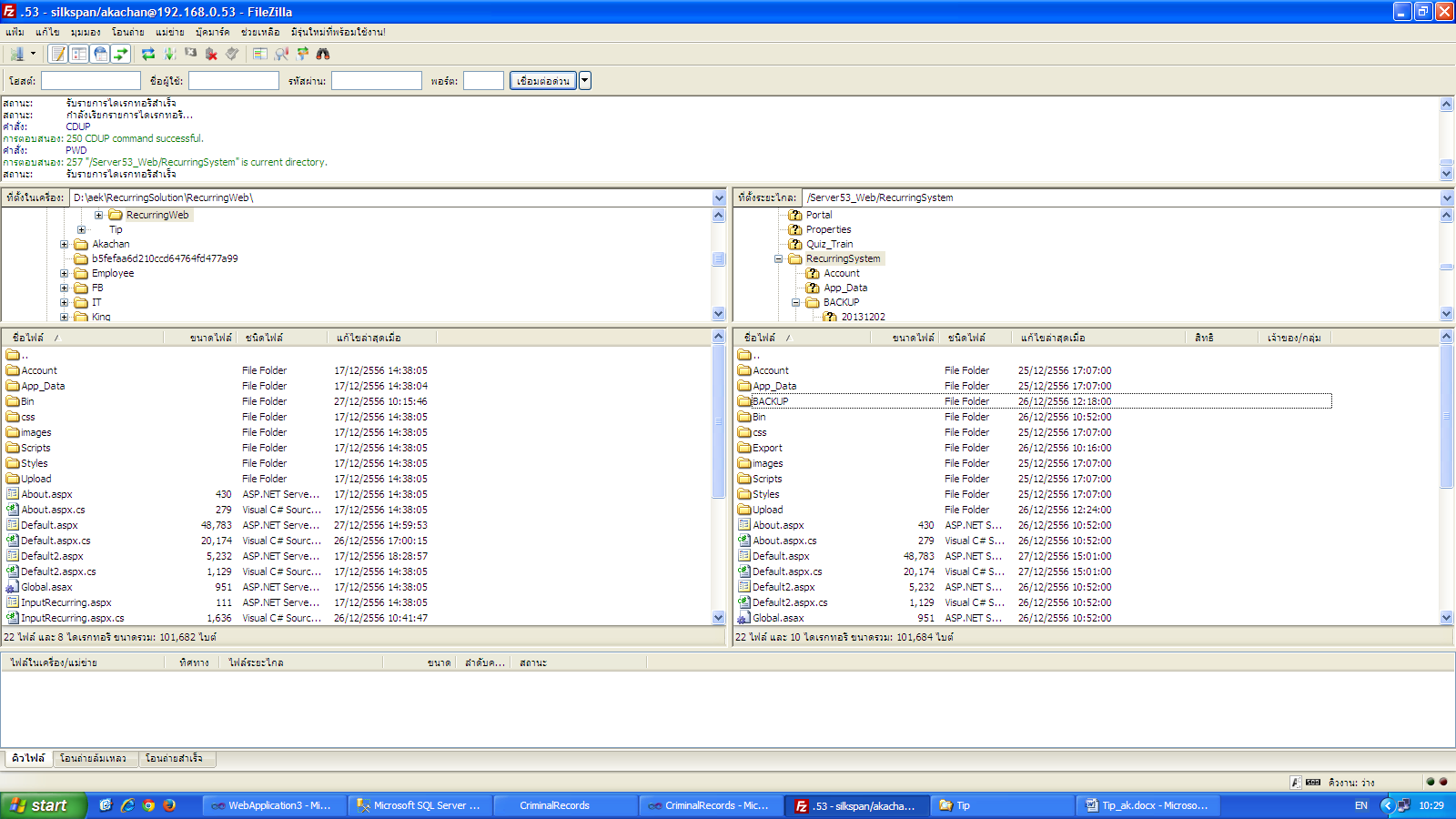
**Tip**

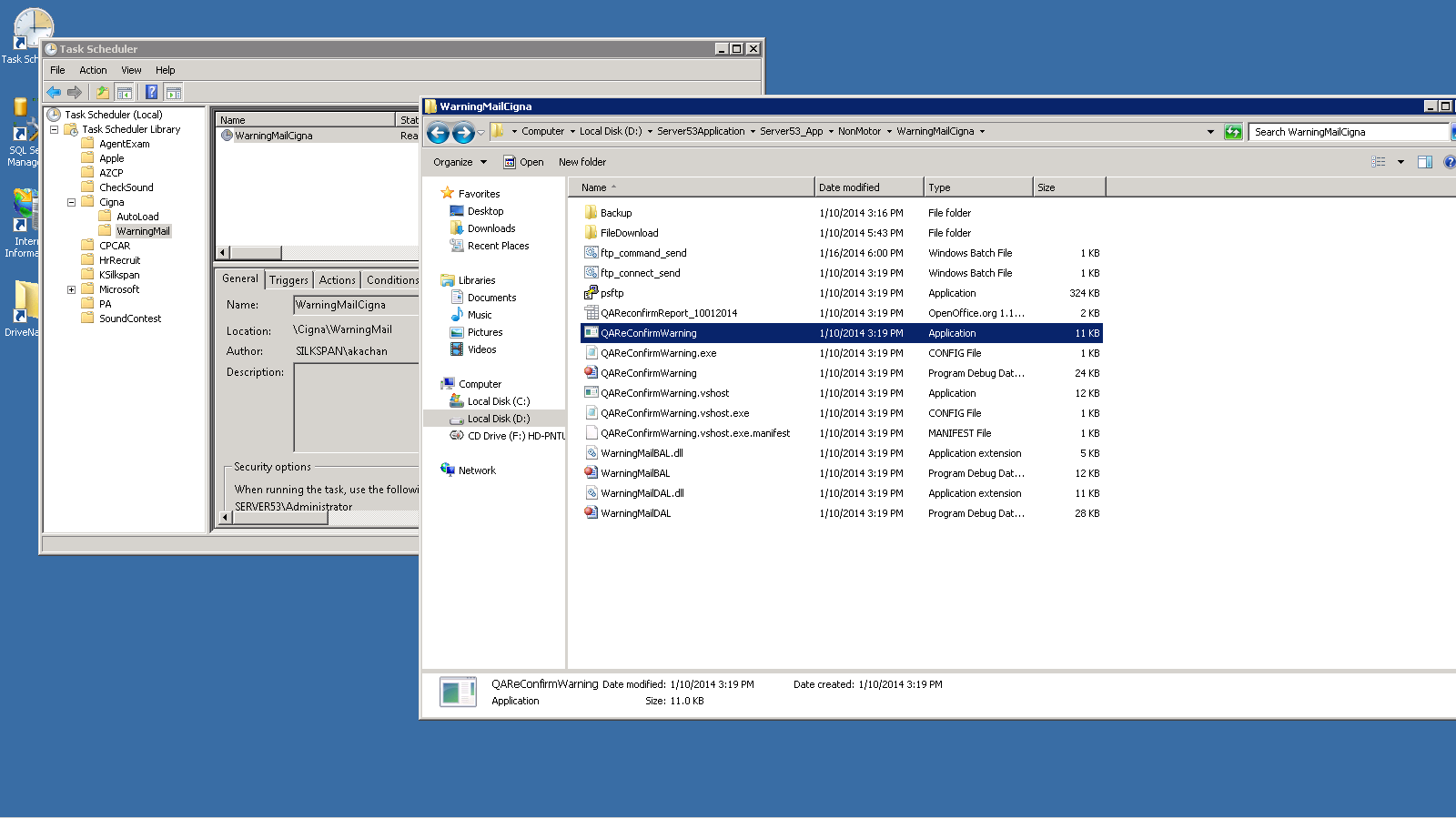
****

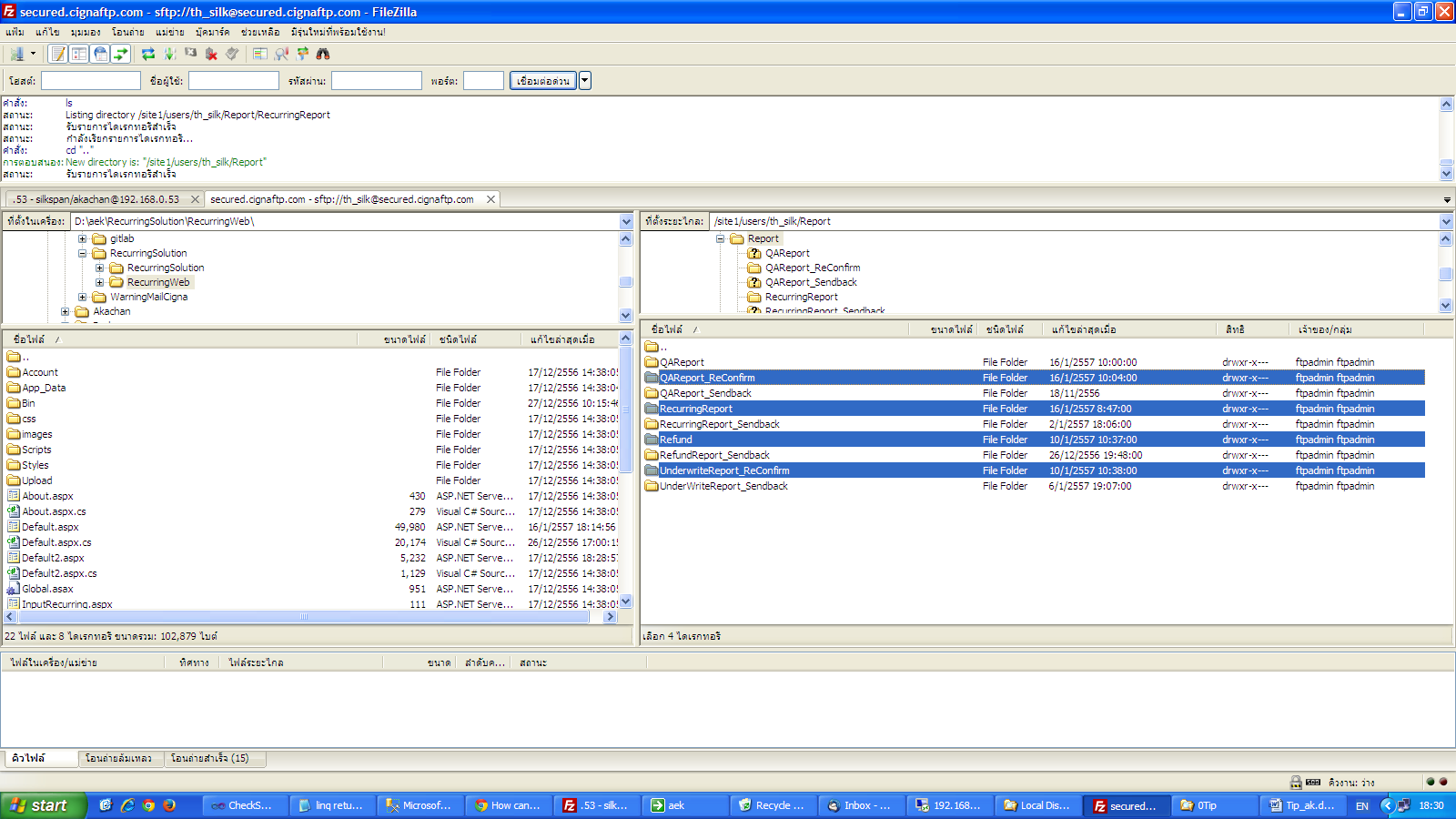
**เพิ่มพอย์เตอร์ ให้จับชื่อไฟล์ฝั่งขวามือใหอัตโนมัติ**

**ติกถูกตั้งแต่ Track Active Item in Solution Explorer**

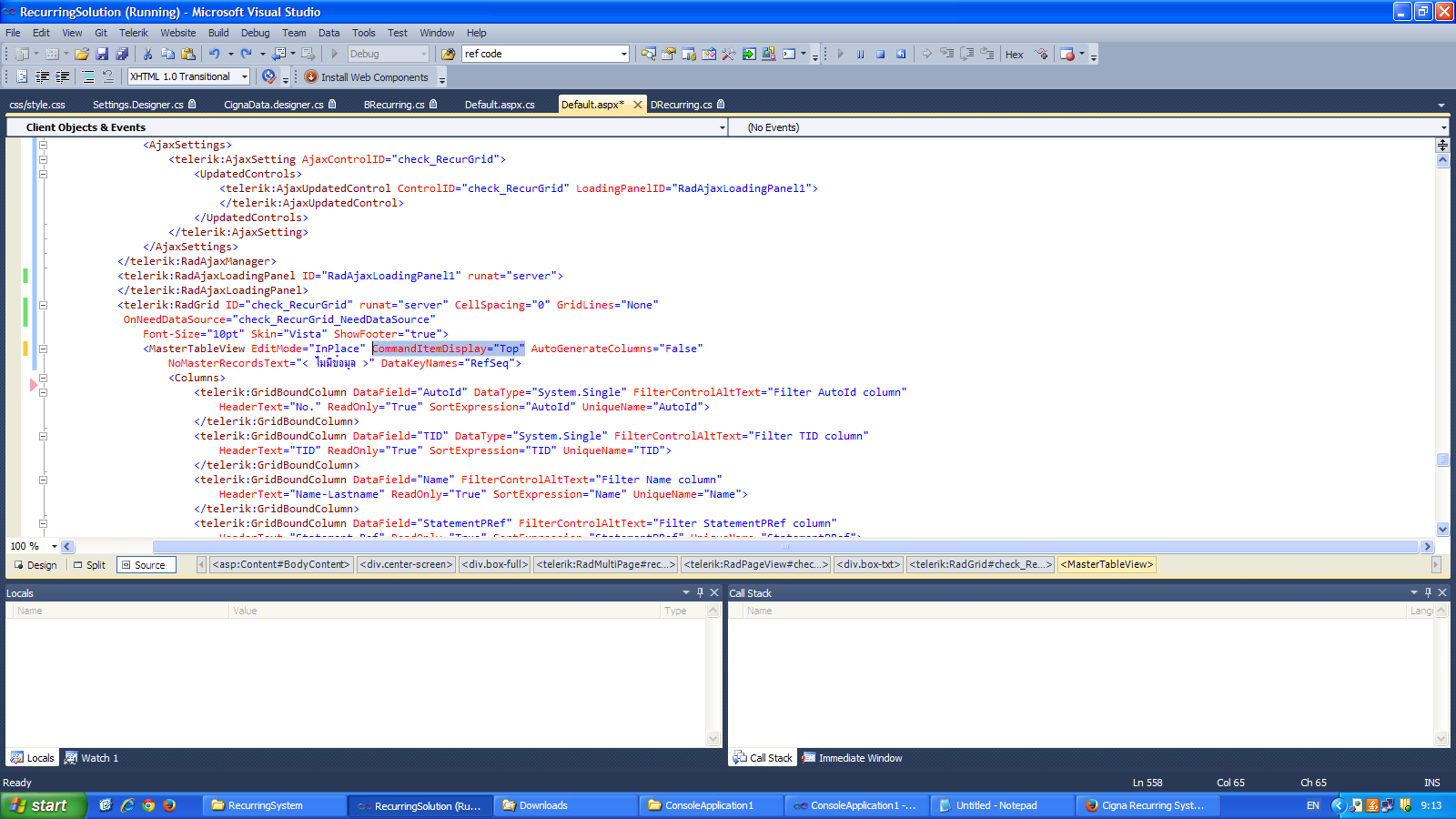
****

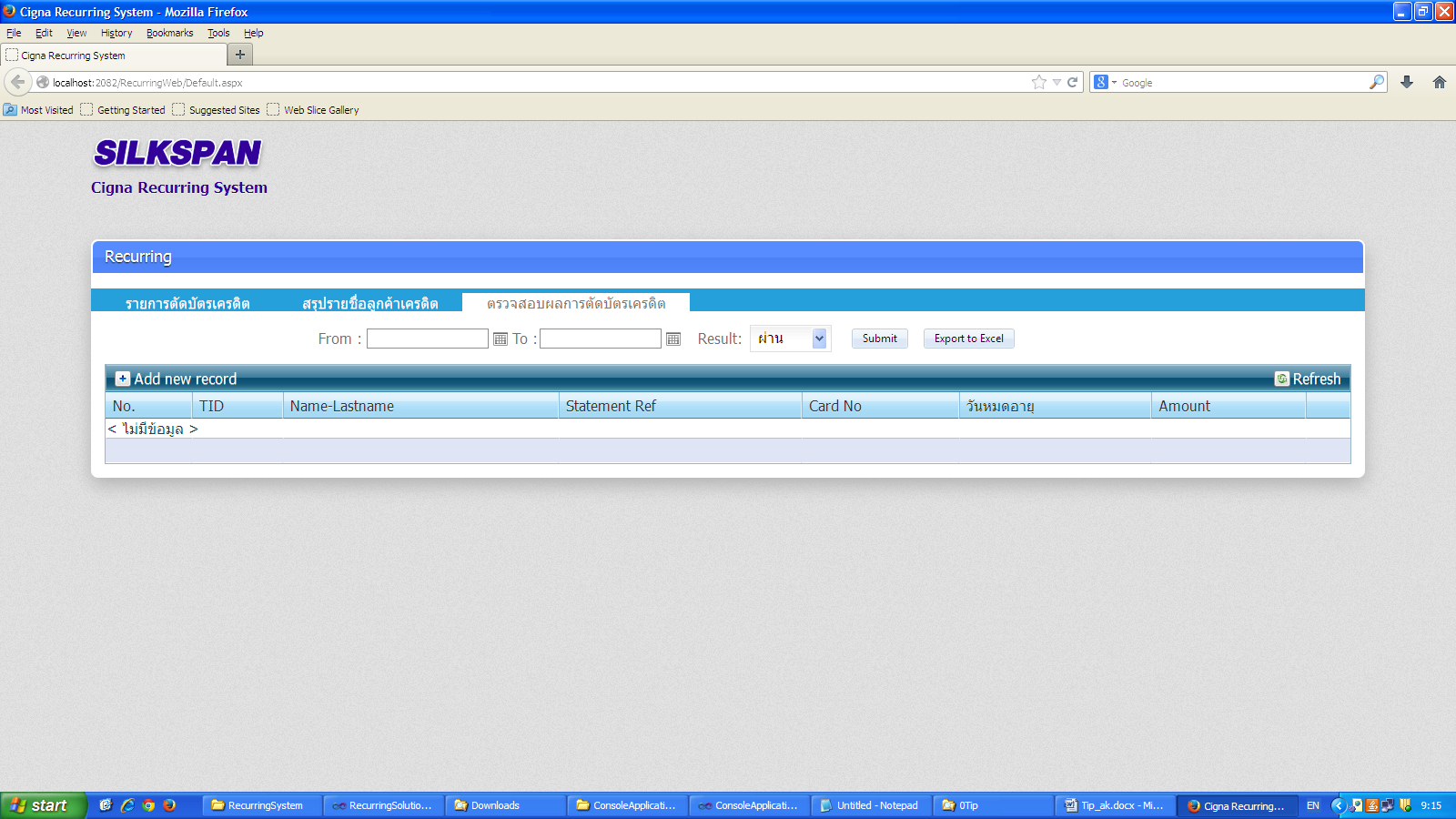
****

****

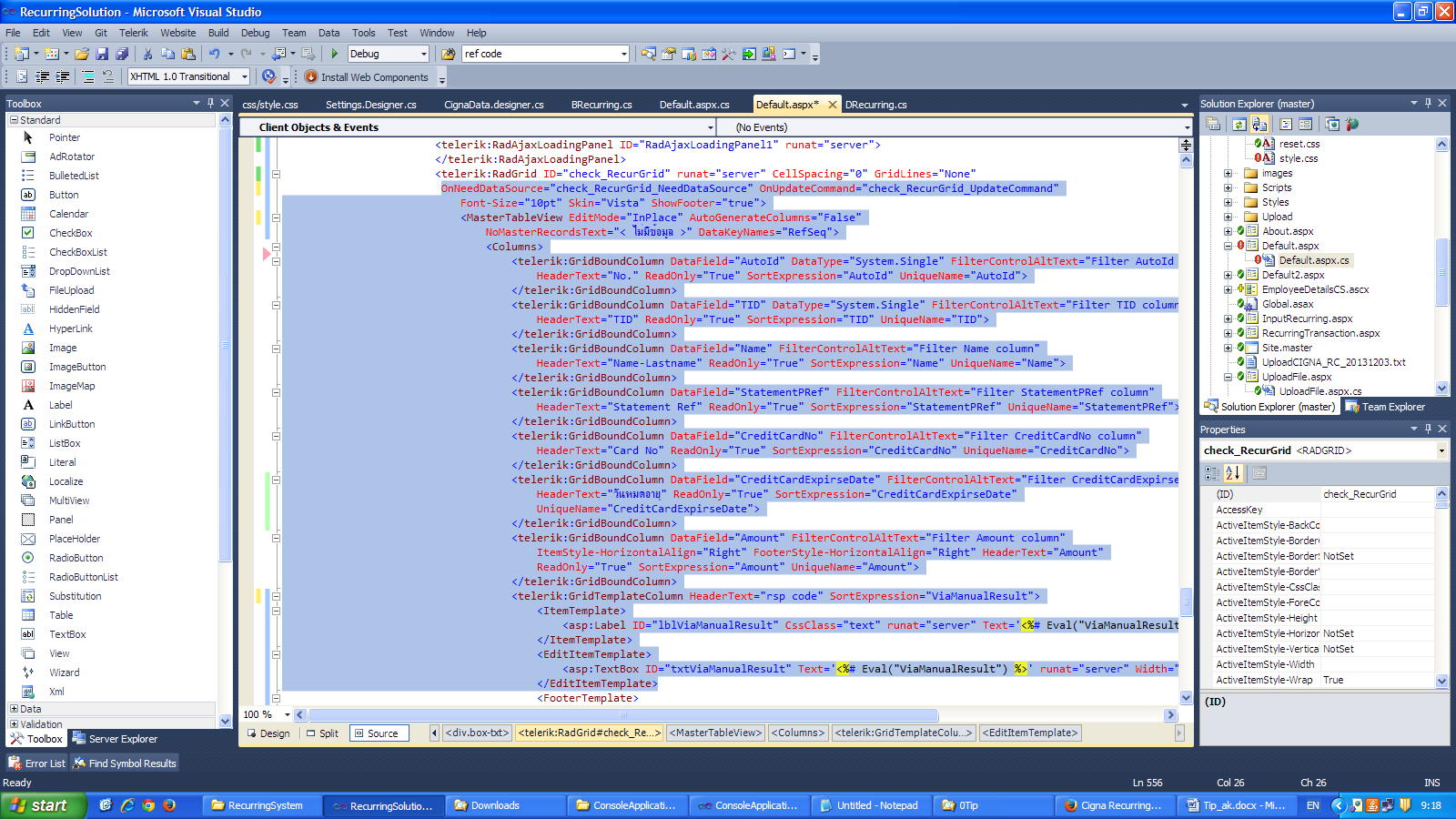
****

**Show add new record**

****

****

**Show Edit**

****

Export Excel

Desing

<telerik:RadGrid ID="grid\_export\_track" Visible="false" runat="server">

</telerik:RadGrid>

Codebehind

protected void btnExport\_Track\_Click(object sender, EventArgs e)

{

DataTable dataTable = new DataTable();

String txtDept = ddlDept.SelectedValue.ToString();

String txtCode = txtHrCode.Text.Trim();

if (txtDept == "All")

{

dataTable = dbCriminal.getData\_Criminal\_Export\_ALL(txtCode);

}

else

{

dataTable = dbCriminal.getData\_Criminal\_Export\_Bydept(txtDept, txtCode);

}

grid\_export\_track.DataSource = dataTable;

grid\_export\_track.DataBind();

//string alternateText = (sender as System.Web.UI.WebControls.ImageButton).AlternateText;

string alternateText = "Html";

grid\_export\_track.ExportSettings.Excel.Format = (GridExcelExportFormat)Enum.Parse(typeof(GridExcelExportFormat), alternateText);

grid\_export\_track.ExportSettings.FileName = "Export\_Criminal\_Record";

grid\_export\_track.ExportSettings.IgnorePaging = true;

grid\_export\_track.ExportSettings.OpenInNewWindow = true;

grid\_export\_track.MasterTableView.ExportToExcel();

}

Export2

<telerik:RadGrid ID="grid\_export\_track" Visible="false" runat="server">

</telerik:RadGrid>

<asp:GridView ID="gvExport" runat="server" AutoGenerateColumns="true"></asp:GridView>

protected void btnExport\_Track\_Click(object sender, EventArgs e)

{

DataTable dataTable = new DataTable();

String txtDept = ddlDept\_Track.SelectedValue.ToString();

String txtCode = txtHrCode.Text.Trim();

if (txtDept == "All")

{

dataTable = dbCriminal.getData\_Criminal\_Export\_ALL(txtCode);

}

else

{

dataTable = dbCriminal.getData\_Criminal\_Export\_Bydept(txtDept, txtCode);

}

grid\_export\_track.DataSource = dataTable;

grid\_export\_track.DataBind();

gvExport.DataSource = dataTable;

gvExport.DataBind();

string alternateText = "Html";

//grid\_export\_track.ExportSettings.Excel.Format = (GridExcelExportFormat)Enum.Parse(typeof(GridExcelExportFormat), alternateText);

////grid\_export\_track.ExportSettings.ExportOnlyData = true;

//grid\_export\_track.ExportSettings.FileName = "Export\_Criminal\_Record";

//grid\_export\_track.ExportSettings.IgnorePaging = true;

//grid\_export\_track.ExportSettings.OpenInNewWindow = true;

//grid\_export\_track.MasterTableView.ExportToExcel();

Export("Export\_Criminal\_Record.xls", gvExport);

}

public static void Export(string fileName, GridView gv)

{

HttpContext.Current.Response.Clear();

HttpContext.Current.Response.AddHeader("content-disposition", string.Format("attachment; filename={0}", fileName));

HttpContext.Current.Response.ContentType = "application/ms-excel";

using (StringWriter sw = new StringWriter())

{

using (HtmlTextWriter htw = new HtmlTextWriter(sw))

{

// Create a form to contain the grid

Table table = new Table();

// add the header row to the table

if (gv.HeaderRow != null)

{

PrepareControlForExport(gv.HeaderRow);

table.Rows.Add(gv.HeaderRow);

}

// add each of the data rows to the table

foreach (GridViewRow row in gv.Rows)

{

PrepareControlForExport(row);

table.Rows.Add(row);

}

// add the footer row to the table

if (gv.FooterRow != null)

{

PrepareControlForExport(gv.FooterRow);

table.Rows.Add(gv.FooterRow);

}

// render the table into the htmlwriter

table.RenderControl(htw);

// render the htmlwriter into the response

HttpContext.Current.Response.Write(sw.ToString());

HttpContext.Current.Response.End();

}

}

}

private static void PrepareControlForExport(Control control)

{

for (int i = 0; i < control.Controls.Count; i++)

{

Control current = control.Controls[i];

if (current is LinkButton)

{

control.Controls.Remove(current);

control.Controls.AddAt(i, new LiteralControl((current as LinkButton).Text));

}

else if (current is System.Web.UI.WebControls.ImageButton)

{

control.Controls.Remove(current);

control.Controls.AddAt(i, new LiteralControl((current as System.Web.UI.WebControls.ImageButton).AlternateText));

}

else if (current is HyperLink)

{

control.Controls.Remove(current);

control.Controls.AddAt(i, new LiteralControl((current as HyperLink).Text));

}

else if (current is DropDownList)

{

control.Controls.Remove(current);

control.Controls.AddAt(i, new LiteralControl((current as DropDownList).SelectedItem.Text));

}

else if (current is CheckBox)

{

control.Controls.Remove(current);

control.Controls.AddAt(i, new LiteralControl((current as CheckBox).Checked ? "True" : "False"));

}

if (current.HasControls())

{

PrepareControlForExport(current);

}

}

}

Export 3

protected void btnExport\_Track\_Click(object sender, EventArgs e)

{

gvExport.DataSource = dataTable;

gvExport.DataBind();

Export("Export\_Criminal\_Record.xls", gvExport);

}

public static void Export(string fileName, GridView gv)

{

HttpContext.Current.Response.Clear();

HttpContext.Current.Response.AddHeader("content-disposition", string.Format("attachment; filename={0}", fileName));

HttpContext.Current.Response.ContentType = "application/ms-excel";

using (StringWriter sw = new StringWriter())

{

using (HtmlTextWriter htw = new HtmlTextWriter(sw))

{

// Create a form to contain the grid

Table table = new Table();

// add the header row to the table

if (gv.HeaderRow != null)

{

PrepareControlForExport(gv.HeaderRow);

table.Rows.Add(gv.HeaderRow);

}

// add each of the data rows to the table

foreach (GridViewRow row in gv.Rows)

{

PrepareControlForExport(row);

table.Rows.Add(row);

}

// add the footer row to the table

if (gv.FooterRow != null)

{

PrepareControlForExport(gv.FooterRow);

table.Rows.Add(gv.FooterRow);

}

// render the table into the htmlwriter

table.RenderControl(htw);

// render the htmlwriter into the response

HttpContext.Current.Response.Write(sw.ToString());

HttpContext.Current.Response.End();

}

}

}

private static void PrepareControlForExport(Control control)

{

for (int i = 0; i < control.Controls.Count; i++)

{

Control current = control.Controls[i];

if (current is LinkButton)

{

control.Controls.Remove(current);

control.Controls.AddAt(i, new LiteralControl((current as LinkButton).Text));

}

else if (current is System.Web.UI.WebControls.ImageButton)

{

control.Controls.Remove(current);

control.Controls.AddAt(i, new LiteralControl((current as System.Web.UI.WebControls.ImageButton).AlternateText));

}

else if (current is HyperLink)

{

control.Controls.Remove(current);

control.Controls.AddAt(i, new LiteralControl((current as HyperLink).Text));

}

else if (current is DropDownList)

{

control.Controls.Remove(current);

control.Controls.AddAt(i, new LiteralControl((current as DropDownList).SelectedItem.Text));

}

else if (current is CheckBox)

{

control.Controls.Remove(current);

control.Controls.AddAt(i, new LiteralControl((current as CheckBox).Checked ? "True" : "False"));

}

if (current.HasControls())

{

PrepareControlForExport(current);

}

}

}

Show & hide column in telerik grid

design

<telerik:GridTemplateColumn HeaderStyle-HorizontalAlign="Center" HeaderStyle-Width="60" UniqueName="edit\_column"

HeaderText="แก้ไข" ItemStyle-HorizontalAlign="Center">

<ItemTemplate>

<asp:CheckBox ID="chkEdit" runat="server" AutoPostBack="true" OnCheckedChanged="chkEdit\_CheckedChanged"

Text="แก้ไข" /></ItemTemplate>

<HeaderStyle HorizontalAlign="Center" Width="60px" />

<ItemStyle HorizontalAlign="Center" />

</telerik:GridTemplateColumn>

Code behind

Show cell

protected void gridCriminal\_Track\_ItemDataBound(Object sender, GridItemEventArgs e)

{

int i = 1;

foreach (GridDataItem Item in gridCriminal\_Track.Items)

{

Item["gv\_No"].Text = i.ToString();

i++;

Label lbSendCriminal\_Track = (Label)Item.FindControl("lbSendCriminal\_Track");

DropDownList ddlPass\_Track = (DropDownList)Item.FindControl("ddlPass\_Track");

ddlPass\_Track.SelectedValue = Item["PassCriminal"].Text.Trim();

DropDownList DDLPass\_Track = (DropDownList)Item.FindControl("ddlPass\_Track");

CheckBox chkSendCriminal\_Track = (CheckBox)Item.FindControl("chkSendCriminal\_Track");

RadDatePicker DateCriminal\_Track = (RadDatePicker)Item.FindControl("DateCriminal\_Track");

LinkButton btnDelete = (LinkButton)Item.FindControl("btnDelete");

TableCell edit\_column = (TableCell)Item["edit\_column"];

TableCell cancel\_column = (TableCell)Item["cancel\_column"];

if (lbSendCriminal\_Track.Text.Trim() != String.Empty)

{

chkSendCriminal\_Track.Checked = true;

}

if ((Session["UserCode"].ToString().Trim() !="admin")&&(Session["UserCode"].ToString().Trim() !="Admin"))

{

edit\_column.Visible = false;

cancel\_column.Visible = false;

}

}

}

Show Column

protected void gridCriminal\_Track\_ItemDataBound(Object sender, GridItemEventArgs e)

{

int i = 1;

if ((Session["UserCode"].ToString().Trim() != "admin") && (Session["UserCode"].ToString().Trim() != "Admin"))

{

(gridCriminal\_Track.MasterTableView.GetColumn("edit\_column") as GridTemplateColumn).Visible = false;

(gridCriminal\_Track.MasterTableView.GetColumn("cancel\_column") as GridTemplateColumn).Visible = false;

}

Popup

System.Web.UI.ScriptManager.RegisterStartupScript(this.Page, typeof(string), "Alert", "alert('บันทึกข้อมูลสำเร็จ');", true);

Show footer and check row in grid export -- Recurring

<telerik:RadGrid ID="check\_RecurGrid" runat="server" CellSpacing="0" GridLines="None"

AllowSorting="True" Font-Names="Tahoma"

Font-Size="10pt" Skin="Vista" ShowFooter="true">

<ClientSettings>

<Selecting CellSelectionMode="None"></Selecting>

<ClientEvents OnRowMouseOver="RowMouseOver" />

</ClientSettings>

<SelectedItemStyle BorderColor="#00CCFF" BorderStyle="None" BorderWidth="0.2" Font-Bold="True" />

<MasterTableView AutoGenerateColumns="False" NoMasterRecordsText="< ไม่มีข้อมูล >">

<CommandItemSettings ExportToPdfText="Export to PDF"></CommandItemSettings>

<RowIndicatorColumn Visible="True" FilterControlAltText="Filter RowIndicator column">

<HeaderStyle Width="20px"></HeaderStyle>

</RowIndicatorColumn>

<ExpandCollapseColumn Visible="True" FilterControlAltText="Filter ExpandColumn column">

<HeaderStyle Width="20px"></HeaderStyle>

</ExpandCollapseColumn>

<Columns>

<telerik:GridClientSelectColumn UniqueName="ClientSelectColumn" HeaderText="CheckRow">

</telerik:GridClientSelectColumn>

<telerik:GridBoundColumn DataField="AutoId" DataType="System.Single" FilterControlAltText="Filter AutoId column"

HeaderText="No." ReadOnly="True" SortExpression="AutoId" UniqueName="AutoId">

</telerik:GridBoundColumn>

<telerik:GridBoundColumn DataField="TID" DataType="System.Single" FilterControlAltText="Filter TID column"

HeaderText="TID" ReadOnly="True" SortExpression="TID" UniqueName="TID">

</telerik:GridBoundColumn>

<telerik:GridBoundColumn DataField="Name" FilterControlAltText="Filter Name column"

HeaderText="Name-Lastname" ReadOnly="True" SortExpression="Name" UniqueName="Name">

</telerik:GridBoundColumn>

<telerik:GridBoundColumn DataField="StatementPRef" FilterControlAltText="Filter StatementPRef column"

HeaderText="Statement Ref" ReadOnly="True" SortExpression="StatementPRef" UniqueName="StatementPRef">

</telerik:GridBoundColumn>

<telerik:GridBoundColumn DataField="CreditCardNo" FilterControlAltText="Filter CreditCardNo column"

HeaderText="Card No" ReadOnly="True" SortExpression="CreditCardNo" UniqueName="CreditCardNo">

</telerik:GridBoundColumn>

<telerik:GridBoundColumn DataField="Amount" FilterControlAltText="Filter Amount column"

HeaderText="Amount" ReadOnly="True" SortExpression="Amount" UniqueName="Amount">

</telerik:GridBoundColumn>

<telerik:GridBoundColumn DataField="ViaManualResult" FilterControlAltText="Filter ViaManualResult column"

HeaderText="rsp code" FooterText="บาท" ReadOnly="True" SortExpression="ViaManualResult" UniqueName="ViaManualResult">

</telerik:GridBoundColumn>

<telerik:GridBoundColumn DataField="ApprovedCode" FilterControlAltText="Filter ApprovedCode column"

HeaderText="Approval" ReadOnly="True" SortExpression="ApprovedCode" UniqueName="ApprovedCode">

</telerik:GridBoundColumn>

<telerik:GridBoundColumn DataField="ReferenceNo" FilterControlAltText="Filter ReferenceNo column"

HeaderText="ref code" ReadOnly="True" SortExpression="ReferenceNo" UniqueName="ReferenceNo">

</telerik:GridBoundColumn>

<telerik:GridBoundColumn DataField="BatchNo" FilterControlAltText="Filter BatchNo column"

HeaderText="Batch" ReadOnly="True" SortExpression="BatchNo" UniqueName="BatchNo">

</telerik:GridBoundColumn>

</Columns>

<EditFormSettings>

<EditColumn FilterControlAltText="Filter EditCommandColumn column">

</EditColumn>

</EditFormSettings>

</MasterTableView>

<ClientSettings EnableRowHoverStyle="true">

<Selecting AllowRowSelect="True"></Selecting>

<ClientEvents OnRowMouseOver="RowMouseOver" />

</ClientSettings>

<FilterMenu EnableImageSprites="false">

</FilterMenu>

</telerik:RadGrid>

Code behind

protected void check\_rBtnSubmit\_Click(object sender, EventArgs e)

{

try

{

check\_ViewData();

check\_RecurGrid.DataBind();//ak

check\_CheckMultiRowSelection();

double sum = 0;

Int16 no = 0;

foreach (GridDataItem item in check\_RecurGrid.MasterTableView.Items)//ak

{

sum += Convert.ToDouble(DataBinder.Eval(item.DataItem, "Amount", "{0}"));

no = Convert.ToInt16(DataBinder.Eval(item.DataItem, "AutoId", "{0}"));

}

GridFooterItem footeritem = (GridFooterItem)check\_RecurGrid.MasterTableView.GetItems(Telerik.Web.UI.GridItemType.Footer)[0];

footeritem["CreditCardNo"].Text = "รวม : " + no.ToString("###") + " รายการ";

footeritem["Amount"].Text = sum.ToString("###.00") + "";

//(footeritem.FindControl("lblAutoId") as System.Web.UI.WebControls.Label).Text = "รวม : " + no.ToString("###") + " รายการ";

//(footeritem.FindControl("lblSumAmount") as System.Web.UI.WebControls.Label).Text = sum.ToString("###.00") + "";

}

catch (Exception)

{

throw;

}

}

private DataTable CheckRowGridToDB(RadGrid grid\_name)

{

DataTable dt = new DataTable();

DataRow dr = null;

int columncount = 0;

foreach (GridColumn column in grid\_name.MasterTableView.Columns)//ak

{

if (!string.IsNullOrEmpty(column.UniqueName) && !string.IsNullOrEmpty(column.HeaderText))

{

if (!check\_CheckColumn(column.UniqueName))

{

columncount++;

dt.Columns.Add(column.UniqueName, typeof(string));

}

}

}

foreach (GridDataItem item in grid\_name.MasterTableView.Items)//ak

{

System.Web.UI.WebControls.CheckBox chk = (System.Web.UI.WebControls.CheckBox)item["ClientSelectColumn"].Controls[0];

if (chk.Checked)

{

dr = dt.NewRow();

for (int i = 0; i < columncount; i++)

{

dr[i] = item[grid\_name.MasterTableView.Columns[i + 1].UniqueName].Text;

}

dt.Rows.Add(dr);

}

}

dr = dt.NewRow();

for (int i = 0; i < columncount; i++)

{

dr[i] = string.Empty;

}

dt.Rows.Add(dr);

return dt;

}

Alert

System.Web.UI.ScriptManager.RegisterStartupScript(this.Page, typeof(string), "Alert", "alert('บันทึกข้อมูลสำเร็จ');", true);

Popup confirm

<asp:LinkButton ID="btnDelete" runat="server" CommandName="Delete" OnClientClick="Confirm()"

CommandArgument='<%# DataBinder.Eval(Container, "DataItem.DataId") %>'>

<asp:Image ID="imgDelete" runat="server" ImageUrl="~/Images/Button/btnMini - Delete.png"

Width="16" />

<asp:Label ID="lblDelete" runat="server" Text="ยกเลิก" />

</asp:LinkButton>

<script type="text/javascript">

function Confirm() {

var confirm\_value = document.createElement("INPUT");

confirm\_value.type = "hidden";

confirm\_value.name = "confirm\_value";

if (confirm("ต้องการยกเลิกข้อมูล?")) {

confirm\_value.value = "Yes";

} else {

confirm\_value.value = "No";

}

document.forms[0].appendChild(confirm\_value);

}

</script>

</asp:Content>

Code behind

if (e.CommandName == "Delete")

{

if (e.CommandArgument.ToString().Trim() != String.Empty)

{

string confirmValue = Request.Form["confirm\_value"];

if (confirmValue == "Yes")

{

GridDataItem Items = e.Item as GridDataItem;

HiddenField HidDataId = Items.FindControl("HidDataId") as HiddenField;

string StrDataId = HidDataId.Value.ToString().Trim();

int xx = dbCriminal.updateCriminal\_isnull(StrDataId);

gridCriminal\_Track\_DataBind();

gridCriminal\_DataBind();

}

}

}

Linq

Code benind

public void summary\_ViewData()

{

string str\_date\_from = string.Empty;

string str\_date\_to = string.Empty;

System.Data.DataTable dt = new System.Data.DataTable();

try

{

if (!string.IsNullOrEmpty(summary\_rDpFrom.SelectedDate.ToString()))

{

str\_date\_from = Formatter.ToStringTH\_start(summary\_rDpFrom.SelectedDate.Value);

mRecurring.StartDate = Convert.ToDateTime(str\_date\_from);//ak

}

if (!string.IsNullOrEmpty(summary\_rDpTo.SelectedDate.ToString()))

{

str\_date\_to = Formatter.ToStringTH\_end(summary\_rDpTo.SelectedDate.Value);

mRecurring.EndDate = Convert.ToDateTime(str\_date\_to);//ak

}

if (!string.IsNullOrEmpty(summary\_rDpFrom.SelectedDate.ToString()) && !string.IsNullOrEmpty(summary\_rDpTo.SelectedDate.ToString()))//ak

{

CultureInfo ci = new CultureInfo(System.Configuration.ConfigurationManager.AppSettings["CultureName"]);

dt = bRecurring.summary\_Recurring(mRecurring, ci.ToString());

Session["data"] = dt;

summary\_RecurGrid.DataSource = dt;//ak

}

//mDataTable.MyTable = dt;

}

catch (Exception)

{

throw;

}

}

Class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using RecurringDAL;

using RecurringModel;

using System.Data;

public DataTable summary\_Recurring(MRecurring mRecur,string culture)

{

DataTable dt = new DataTable();

try

{

dt = dRecurring.summary\_GetData(mRecur, culture);

}

catch (Exception)

{

throw;

}

return dt;

}

Dataaccess

public DataTable summary\_GetData(MRecurring mRecur, string culture)

{

DataTable dtReturn = new DataTable();

dtReturn.Columns.Add("AutoId");

dtReturn.Columns.Add("TID");

dtReturn.Columns.Add("PolicyNumber");

dtReturn.Columns.Add("Name");

dtReturn.Columns.Add("PaymentTerm");

dtReturn.Columns.Add("PolicyYear");

dtReturn.Columns.Add("PaySeq");

dtReturn.Columns.Add("PayDate");

dtReturn.Columns.Add("PayAmount");

DataRow Dr;

var qPeriod = (from s in db.Cigna\_PeriodPay\_Logs

join r in db.Insur\_Cignas on s.PolicyNumber equals r.PolicyNumber

where s.PayDate >= mRecur.StartDate && s.PayDate <= mRecur.EndDate

&& s.PayResult == null //null is not pay

orderby s.PaySeq descending

select new {

r.TID

,r.PolicyNumber

,r.Name ,r.Lastname

,r.PaymentTerm

,s.PaySeq

,s.PayDate

,r.Premium

,s.Seq

,s.PayResult

} ).ToList();

var qData = (from s in qPeriod

where s.PayDate >= mRecur.StartDate && s.PayDate <= mRecur.EndDate

&& s.PayResult == null

group s by new {s.PolicyNumber} into g

select new{

TID = g.Min(p => p.TID)

,PolicyNumber=g.Key

,Name =g.Min(p => p.Name)

,Lastname =g.Min(p => p.Lastname)

,PaymentTerm =g.Min(p => p.PaymentTerm)

,PaySeq =g.Min(p => p.PaySeq)

,PayDate =g.Min(p => p.PayDate)

,Premium =g.Min(p => p.Premium)

,Seq =g.Min(p => p.Seq)

,PayResult = g.Min(p => p.PayResult)

}).ToList();

//var qData1 = (from r in qData1

// group r by r.PolicyNumber into g

// select g.Min(p => p.Seq)).ToList();

foreach (var item in qData)

{

//MRecurring mRecur2 = new MRecurring();

Dr = dtReturn.NewRow();

Dr["AutoId"] = AudoId.ToString();

Dr["TID"] = item.TID;

Dr["PolicyNumber"] = item.PolicyNumber.PolicyNumber;

Dr["Name"] = item.Name+" "+item.Lastname;

Dr["PaymentTerm"] = item.PaymentTerm;

Dr["PaySeq"] = item.PaySeq;

CultureInfo ci = new CultureInfo(culture);

Dr["PayDate"] = string.IsNullOrEmpty(item.PayDate.ToString()) ? DBNull.Value.ToString() :(Convert.ToDateTime(item.PayDate).ToString("dd/MM/yyyy",ci));

Dr["PayAmount"] = item.Premium;

dtReturn.Rows.Add(Dr);

AudoId++;

}

return dtReturn;

}

Report

