Telerik & Dev Notes

Contents

Web Resources [[#web-resources|8]]

Telerik Training & Demo Videos [[#telerik-training-demo-videos|8]]

.NET (dotNET) Version for our Projects/Products [[#net-dotnet-version-for-our-projectsproducts|9]]

Maintaining Telerik [[#maintaining-telerik|9]]

Identify Installed Version(s) [[#identify-installed-versions|9]]

Update Telerik Installation [[#update-telerik-installation|9]]

Create Telerik Solution for Legality projects [[#create-telerik-solution-for-legality-projects|10]]

Create a Telerik Solution using the Telerik Wizard [[#create-a-telerik-solution-using-the-telerik-wizard|10]]

VS 2022 [[#vs-2022|10]]

Core vs. Traditional .NET (v4.x) [[#core-vs.-traditional-.net-v4.x|10]]

Older VS Versions [[#older-vs-versions|12]]

~~Create a Telerik Solution without using the Telerik Wizard~~ [[#create-a-telerik-solution-without-using-the-telerik-wizard|16]]

Telerik Systray App [[#telerik-systray-app|19]]

Steps to create a SysTray app [[#steps-to-create-a-systray-app|19]]

Visual Studio Info, Shortcuts & Tricks [[#visual-studio-info-shortcuts-tricks|22]]

Shortcut Keys [[#shortcut-keys|22]]

Add Resources to a Solution (Icons, Graphics, Files, etc.) [[#add-resources-to-a-solution-icons-graphics-files-etc.|22]]

Committing newly added Resources to GIT [[#committing-newly-added-resources-to-git|23]]

Our Source File Structure [[#our-source-file-structure|23]]

TODO Comments [[#todo-comments|24]]

Build [[#build|25]]

Set the Build Output Folder [[#set-the-build-output-folder|25]]

Remove the Namespace from the Build Output (EXE) File Name [[#remove-the-namespace-from-the-build-output-exe-file-name|25]]

Copy the required DLL's to the build folder [[#copy-the-required-dlls-to-the-build-folder|25]]

Deployment [[#deployment|25]]

Telerik Application Deployment [[#telerik-application-deployment|25]]

ILMerge Telerik WinForms assemblies [[#ilmerge-telerik-winforms-assemblies|26]]

Application Icons –Multisize ICO files [[#application-icons-multisize-ico-files|26]]

Add updater [[#add-updater|26]]

Telerik Common Properties [[#telerik-common-properties|27]]

Forms [[#forms|27]]

Borders [[#borders|27]]

TitleBar [[#titlebar|27]]

Position [[#position|28]]

Size [[#size|28]]

AutoScaleMode [[#autoscalemode|29]]

frmMain [[#frmmain|29]]

Telerik Form Templates [[#telerik-form-templates|29]]

RadForm - rfrm [[#radform---rfrm|30]]

RadRibbonForm [[#radribbonform|31]]

RadStatusStrip – rss [[#radstatusstrip-rss|31]]

RadAboutBox [[#radaboutbox|32]]

Add a RadAboutBox to a Project [[#add-a-radaboutbox-to-a-project|33]]

RadShapedForm – rfrmshp [[#radshapedform-rfrmshp|36]]

Control Notes & Naming Conventions [[#control-notes-naming-conventions|36]]

Common Control Properties [[#common-control-properties|36]]

Windows Controls (Non-Telerik) [[#windows-controls-non-telerik|37]]

~~LinkLabel – llbl~~ [[#linklabel-llbl|37]]

* TableLayoutPanel –tlp [[#tablelayoutpanel-tlp|37]]

RadLayoutControl – rloc [[#radlayoutcontrol-rloc|38]]

ToolTip – ttip [[#tooltip-ttip|39]]

PictureBox [[#picturebox|39]]

Telerik – Dialogs [[#telerik-dialogs|39]]

RadColorDialog [[#radcolordialog|39]]

RadMarkupDialog [[#radmarkupdialog|39]]

* RadAboutBox [[#radaboutbox-1|39]]

* RadMessageBox [[#radmessagebox|39]]

RadColorDialog [[#radcolordialog-1|41]]

Telerik – Containers [[#telerik-containers|41]]

* RadDock – rd [[#raddock-rd|41]]

RadPanorama – rp [[#radpanorama-rp|44]]

* RadPageView – rpv [[#radpageview-rpv|44]]

* RadSplitContainer – rsc [[#radsplitcontainer-rsc|48]]

* RadGroupBox – rgb [[#radgroupbox-rgb|50]]

* RadPanel – rpnl [[#radpanel-rpnl|50]]

RadCollapsiblePanel – rcpnl [[#radcollapsiblepanel-rcpnl|51]]

RadScrollablePanel – rspnl [[#radscrollablepanel-rspnl|52]]

* RadSeparator – rs [[#radseparator-rs|52]]

Telerik – Data Controls [[#telerik-data-controls|53]]

RadChartView – rcv [[#radchartview-rcv|53]]

* RadGridView – rgv [[#radgridview-rgv|54]]

RadDataFilter [[#raddatafilter|57]]

RadCardView – rcv [[#radcardview-rcv|58]]

RadDataEntry [[#raddataentry|58]]

RadDataLayout [[#raddatalayout|59]]

RadVirtualGrid – rvg [[#radvirtualgrid-rvg|60]]

RadRangeSelector – rrs [[#radrangeselector-rrs|61]]

RadPivotGrid – rpg [[#radpivotgrid-rpg|61]]

RadPivotFieldList – rpfl [[#radpivotfieldlist-rpfl|62]]

* RadDropDownList – rddl [[#raddropdownlist-rddl|63]]

* CheckedDropDownList – rcddl [[#checkeddropdownlist-rcddl|64]]

* RadListControl – rlc [[#radlistcontrol-rlc|64]]

RadListView – rlv [[#radlistview-rlv|64]]

RadCheckedListBox – rclb [[#radcheckedlistbox-rclb|65]]

RadMultiColumnComboBox – rmcbo [[#radmulticolumncombobox-rmcbo|66]]

* RadTreeView – rtv [[#radtreeview-rtv|67]]

RadBindingNavigator – rbn [[#radbindingnavigator-rbn|70]]

RadMap – rmap [[#radmap-rmap|70]]

RadCarousel – rcar [[#radcarousel-rcar|71]]

Telerik – Gauges [[#telerik-gauges|71]]

RadBulletGraph - rgaugebullet [[#radbulletgraph---rgaugebullet|71]]

RadLinearGauge [[#radlineargauge|72]]

RadRadialGauge [[#radradialgauge|72]]

Telerik – Editors [[#telerik-editors|72]]

* RadAutoCompleteBox – racb [[#radautocompletebox-racb|72]]

* RadBrowseEditor – rbe [[#radbrowseeditor-rbe|73]]

RadCalculatorDropDown – rcdd [[#radcalculatordropdown-rcdd|75]]

RadColorBox – rcb [[#radcolorbox-rcb|75]]

* RadDateTimePicker – rdtp [[#raddatetimepicker-rdtp|76]]

* RadMaskedEditBox – rmeb [[#radmaskededitbox-rmeb|77]]

* RadPropertyGrid – rpg [[#radpropertygrid-rpg|78]]

//RadRichTextBox – (deprecated)// [[#radrichtextbox-deprecated|79]]

RadRichTextEditor - rrte [[#radrichtexteditor---rrte|79]]

RadSpellChecker – rsc [[#radspellchecker-rsc|80]]

* RadSpinEditor – rse [[#radspineditor-rse|81]]

* RadTextBox – rtxt [[#radtextbox-rtxt|81]]

RadTextBoxControl – rtxtbc [[#radtextboxcontrol-rtxtbc|82]]

* RadTimePicker – rtp [[#radtimepicker-rtp|82]]

RadTrackBar – rtb [[#radtrackbar-rtb|83]]

RadPopupEditor – rpe [[#radpopupeditor-rpe|83]]

RadControls – Buttons [[#radcontrols-buttons|84]]

* RadButton – rbtn [[#radbutton-rbtn|84]]

* RadCheckBox – rchk [[#radcheckbox-rchk|85]]

* RadDropDownButton – rddbtn [[#raddropdownbutton-rddbtn|86]]

RadSplitButton – rsbtn [[#radsplitbutton-rsbtn|86]]

* RadRadioButton – rrdo [[#radradiobutton-rrdo|87]]

RadRepeatButton – rrbtn [[#radrepeatbutton-rrbtn|87]]

RadToggleButton – rtbtn [[#radtogglebutton-rtbtn|87]]

RadToggleSwitch – rts [[#radtoggleswitch-rts|88]]

RadControls – Menus & Toolbars [[#radcontrols-menus-toolbars|88]]

RadApplicationMenu – ram [[#radapplicationmenu-ram|88]]

RadContextMenu – rcm [[#radcontextmenu-rcm|89]]

RadContextMenuManager – rcmm [[#radcontextmenumanager-rcmm|89]]

* RadCommandBar – rcb [[#radcommandbar-rcb|90]]

* RadMenu – rm [[#radmenu-rm|94]]

* RadRibbonBar – rrb [[#radribbonbar-rrb|97]]

RadPanorama [[#radpanorama|100]]

RadShortcut (Hotkeys) [[#radshortcut-hotkeys|101]]

RadControls Misc [[#radcontrols-misc|101]]

RadBreadCrumb – rbc [[#radbreadcrumb-rbc|101]]

RadCarousel [[#radcarousel|101]]

RadClock – rclock [[#radclock-rclock|101]]

RadDesktopAlert – rda [[#raddesktopalert-rda|102]]

RadHScrollBar – rhsb [[#radhscrollbar-rhsb|103]]

+-* RadLabel – rlbl [[#radlabel-rlbl|103]]

RadPrintDocument – rpd [[#radprintdocument-rpd|105]]

RadProgressBar – rpb [[#radprogressbar-rpb|105]]

RadWaitingBar – rwb [[#radwaitingbar-rwb|105]]

RadRating [[#radrating|106]]

RadRotator – rrot [[#radrotator-rrot|107]]

RadTitleBar – rtb [[#radtitlebar-rtb|107]]

RadVScrollBar – rvsb [[#radvscrollbar-rvsb|108]]

* RadWizard – rw [[#radwizard-rw|108]]

Telerik – Calendar, Time and Scheduling [[#telerik-calendar-time-and-scheduling|110]]

Related Controls [[#related-controls|110]]

RadCalendar – rc [[#radcalendar-rc|110]]

RadScheduler – rs [[#radscheduler-rs|111]]

RadGanttView - rganttv [[#radganttview---rganttv|113]]

Telerik - Document Processing [[#telerik---document-processing|114]]

RadPdfProcessing [[#radpdfprocessing|114]]

RadSpreadProcessing [[#radspreadprocessing|114]]

RadSpreadStreamProcessing [[#radspreadstreamprocessing|115]]

RadWordsProcessing [[#radwordsprocessing|116]]

RadZipLibrary [[#radziplibrary|116]]

RadPdfViewer – rpdfv [[#radpdfviewer-rpdfv|117]]

RadPdfViewerNavigator – rpdfvn [[#radpdfviewernavigator-rpdfvn|118]]

RadDiagram [[#raddiagram|118]]

Telerik – Themes [[#telerik-themes|119]]

ucTelerikThemeSelector in LS.Common.Presentation [[#uctelerikthemeselector-in-ls.common.presentation|119]]

Code to Set the Current Theme %%**%% [[#code-to-set-the-current-theme|120]]

Enable/Disable the globally set theme for a specific control [[#enabledisable-the-globally-set-theme-for-a-specific-control|120]]

Telerik CAB Enabling Kit [[#telerik-cab-enabling-kit|120]]

Telerik – Coded UI [[#telerik-coded-ui|121]]

Edit UI Elements [[#edit-ui-elements|121]]

Hide Control Borders [[#hide-control-borders|121]]

HTML-like Text Formatting [[#html-like-text-formatting|122]]

Next… [[#next|122]]

User Controls [[#user-controls|122]]

Setup Properties Configurable in the VS IDE [[#setup-properties-configurable-in-the-vs-ide|122]]

Legality User Controls [[#legality-user-controls|124]]

DLL Folders to add Items to ToolBox [[#dll-folders-to-add-items-to-toolbox|124]]

ucFilePath.cs [[#ucfilepath.cs|125]]

ucFilter.cs [[#ucfilter.cs|125]]

ucFolderPath.cs [[#ucfolderpath.cs|125]]

ucFolderPathUserSecurity.cs [[#ucfolderpathusersecurity.cs|125]]

ucFtpConnector.cs [[#ucftpconnector.cs|125]]

ucGridFilter.cs [[#ucgridfilter.cs|125]]

ucDateRangeDays [[#ucdaterangedays|126]]

ucDateRangeMonths [[#ucdaterangemonths|126]]

ucGridSmart.cs [[#ucgridsmart.cs|126]]

ucGridSmart2.cs [[#ucgridsmart2.cs|126]]

ucGridStatus.cs [[#ucgridstatus.cs|127]]

ucGridStatus2.cs [[#ucgridstatus2.cs|128]]

ucItemList.cs [[#ucitemlist.cs|130]]

ucLogData.cs [[#uclogdata.cs|130]]

ucLogViewer.cs [[#uclogviewer.cs|131]]

ucPassword.cs [[#ucpassword.cs|131]]

ucUsernamePassword.cs [[#ucusernamepassword.cs|131]]

ucQueryConfig.cs [[#ucqueryconfig.cs|131]]

ucRichTextEditorHighlighter [[#ucrichtexteditorhighlighter|131]]

ucScheduler.cs [[#ucscheduler.cs|132]]

ucSqlDbName.cs [[#ucsqldbname.cs|132]]

ucSqlEditor.cs [[#ucsqleditor.cs|132]]

ucSqlEditor2.cs [[#ucsqleditor2.cs|133]]

ucSqlEditor3.cs [[#ucsqleditor3.cs|133]]

ucSqlEditor4.cs [[#ucsqleditor4.cs|134]]

ucSqlEditor5.cs [[#ucsqleditor5.cs|134]]

ucTableViewer.cs [[#uctableviewer.cs|135]]

ucTelerikThemeSelector.cs [[#uctelerikthemeselector.cs|135]]

ucTreeFilter.cs [[#uctreefilter.cs|135]]

ucUpdaterSettings.cs [[#ucupdatersettings.cs|135]]

ucDatabaseConnector.cs [[#ucdatabaseconnector.cs|136]]

ucTMDatabaseConnector.cs [[#uctmdatabaseconnector.cs|136]]

ucDbWikiOptions [[#ucdbwikioptions|136]]

ucDbWikiBrowser [[#ucdbwikibrowser|137]]

ucEmailConnector.cs [[#ucemailconnector.cs|137]]

Legality Extended (via inheritance) Controls [[#legality-extended-via-inheritance-controls|139]]

LSRadPropertyGrid - lsrpg [[#lsradpropertygrid---lsrpg|139]]

LS.Common Library [[#ls.common-library|139]]

TMDBConnector & ucTMDatabaseConnector Control [[#tmdbconnector-uctmdatabaseconnector-control|139]]

Add an Existing Form from another project [[#add-an-existing-form-from-another-project|140]]

ChilkatDotNet4.dll Library [[#chilkatdotnet4.dll-library|140]]

C# Documentation [[#c-documentation|140]]

C# Examples Online [[#c-examples-online|140]]

Email Features [[#email-features|140]]

ExceptionReporter [[#exceptionreporter|141]]

Enable/Disable ExceptionReporter [[#enabledisable-exceptionreporter|142]]

Manual Exception test menu option [[#manual-exception-test-menu-option|142]]

LS.Common.EmailSupport [[#ls.common.emailsupport|142]]

LS.Common.EmailSupport.Test [[#ls.common.emailsupport.test|142]]

SoftwareKey/SOLO Licensing [[#softwarekeysolo-licensing|143]]

LS.Common.Licensing [[#ls.common.licensing|143]]

LS.Common.Licensing.Test2 [[#ls.common.licensing.test2|143]]

Tools [[#tools|143]]

ControlSpy [[#controlspy|143]]

Visual Style Builder [[#visual-style-builder|143]]

Theme Viewer [[#theme-viewer|143]]

RadThemeManger [[#_Toc133594718|143]]

ShapeEditor [[#shapeeditor|143]]

Element Hierarchy Editor [[#element-hierarchy-editor|144]]

JustCode [[#justcode|144]]

ActiProSoftware Syntax Editor [[#actiprosoftware-syntax-editor|144]]

Licensing [[#licensing|144]]

GIT Source Control on BitBucket [[#git-source-control-on-bitbucket|144]]

Install GIT for Windows [[#install-git-for-windows|144]]

Install TortoiseGIT on your PC [[#install-tortoisegit-on-your-pc|144]]

Create a Repository and Initial Upload of a New Solution [[#create-a-repository-and-initial-upload-of-a-new-solution|144]]

Initial Download of an Existing Solution from a Repository [[#initial-download-of-an-existing-solution-from-a-repository|148]]

Commit a New Version/Update to an Existing Solution Repository [[#commit-a-new-versionupdate-to-an-existing-solution-repository|149]]

Delete (‘Revert’) changes in the local Repository so you can Pull a fresh copy from the web [[#delete-revert-changes-in-the-local-repository-so-you-can-pull-a-fresh-copy-from-the-web|152]]

Force a Local Repository Push to overwrite the GIT Repository version [[#force-a-local-repository-push-to-overwrite-the-git-repository-version|153]]

Roll the current ‘Master’ version back to some previous log time [[#roll-the-current-master-version-back-to-some-previous-log-time|153]]

Pull the Current Version from an Existing Repository [[#pull-the-current-version-from-an-existing-repository|153]]

Branches - Create [[#branches---create|154]]

Branches – See the Current Branch [[#branches-see-the-current-branch|155]]

Branches – Switch [[#branches-switch|155]]

Branches – Delete [[#branches-delete|156]]

Delete Repository but not files [[#delete-repository-but-not-files|156]]

The Repository Browser [[#the-repository-browser|156]]

Change the Current GIT User in TortoiseGit [[#change-the-current-git-user-in-tortoisegit|156]]

Code How Tos [[#code-how-tos|158]]

Snippets [[#snippets|158]]

propg<tab><tab> [[#propgtabtab|158]]

%%//%% Set Form to load on program Start - C# [[#set-form-to-load-on-program-start---c|159]]

~~%%//%% LinkLabel clicked launches http:%%//%%www.legalitysoftware.com in browser - C#~~ [[#linklabel-clicked-launches-httpwww.legalitysoftware.com-in-browser---c|159]]

%%//%% Open/Display a Form [[#opendisplay-a-form|159]]

%%//%% Open a form in the center of the screen, or parent window [[#open-a-form-in-the-center-of-the-screen-or-parent-window|160]]

%%//%% Show, Hide and Close a form [[#show-hide-and-close-a-form|160]]

%%//%% Programmatically minimize a window [[#programmatically-minimize-a-window|160]]

%%//%% Access the Assembly for info like Product Name [[#access-the-assembly-for-info-like-product-name|160]]

%%//%% Close form/ Exit application [[#close-form-exit-application|161]]

%%//%% ‘Help Online’ in web browser (Help menu) [[#help-online-in-web-browser-help-menu|162]]

%%//%% ‘Legality Software Online’ in web browser (Help menu) [[#legality-software-online-in-web-browser-help-menu|162]]

%%//%% About Dialog (Help menu) [[#about-dialog-help-menu|163]]

%%//%% List all controls in the project to create spec doc [[#list-all-controls-in-the-project-to-create-spec-doc|163]]

%%//%% Use Tooltips with Telerik controls [[#use-tooltips-with-telerik-controls|163]]

%%//%% TreeView - Expand All/Collapse All [[#treeview---expand-allcollapse-all|163]]

%%//%% Open a URL in the default browser [[#open-a-url-in-the-default-browser|164]]

%%//%% Enable/Disable Controls based on RadioButton/Checkbox settings [[#enabledisable-controls-based-on-radiobuttoncheckbox-settings|165]]

%%//%% Multiple lines of text with new line (cr/lf) [[#multiple-lines-of-text-with-new-line-crlf|165]]

Create a Delay or Wait for some time [[#create-a-delay-or-wait-for-some-time|165]]

Application Folder – Open Windows Explorer [[#application-folder-open-windows-explorer|165]]

Control 'Bring to Front'/'Send to back' at runtime [[#control-bring-to-frontsend-to-back-at-runtime|165]]

Create a Collapsible Region [[#create-a-collapsible-region|165]]

Conditional Compilation with #define [[#conditional-compilation-with-define|166]]

Base Template Projects [[#base-template-projects|166]]

LS.Base0 [[#ls.base0|166]]

LS.Base0.0 [[#ls.base0.0|167]]

LS.Base1 [[#ls.base1|167]]

LS.Base1.0 [[#ls.base1.0|167]]

LS.Base3 [[#ls.base3|167]]

Next thing… [[#next-thing|168]]

Web Resources

RadControls for WinForms

http://docs.telerik.com/devtools/winforms/introduction

Support
Main page: http://www.telerik.com/support/winforms
Submit Ticket - Product list: http://www.telerik.com/account/support-tickets/available-support-list
Submit Ticket – WinForms:
http://www.telerik.com/account/support-tickets/contact-support-team.aspx?odid=3559243&pid=523
http://www.telerik.com/account/support-tickets/new-support-ticket.aspx

Forums – http://www.telerik.com/community/forums/winforms.aspx
Knowledge Base – http://www.telerik.com/support/kb/winforms.aspx
Code Libraries – http://www.telerik.com/community/code-library/winforms.aspx

Telerik Training & Demo Videos

http://www.telerik.com/videos/winforms

.NET (dotNET) Version for our Projects/Products

As of Dec 2015 all our projects will move to .NET version 4.5.2.

Per this article: *"As previously announced, starting January 12, 2016 Microsoft will no longer provide security updates, technical support or hotfixes for .NET 4, 4.5, and 4.5.1 frameworks. All other framework versions, including 3.5, 4.5.2, 4.6 and 4.6.1, will be supported for the duration of their established lifecycle."*

Maintaining Telerik

Identify Installed Version(s)

Update Telerik Installation

~~Each year in October we will update all systems with the current year, latest version from the 1st quarter.~~

~~As of 10/2015 we will use v2015.2.728.40 until 10/2016~~

As of 07/28/2021 we are using version: 2020.1.113.40

You can download 'Older Versions' of specific packages from http://www.telerik.com/account/your-products/download-list.aspx

Since we will all be on the same version, all projects will use Telerik from the GAC ("UI for WinForms, v. XXX (Dev) [GAC]")

Here is a video on the process:

\\192.168.93.1\KB_MiM\Telerik\20151013) Telerik Q2 update install each October.mp4

After the new Telerik version install we will need to Upgrade the Common Library (1st) then all our Solutions to use this newest version.

Here is a video on the Telerik Solution ‘Upgrade Wizard’:

\\192.168.93.1\KB_MiM\Telerik\20151013) Run Telerik Solution 'Upgrade Wizard' after installing Q2 update in October.mp4

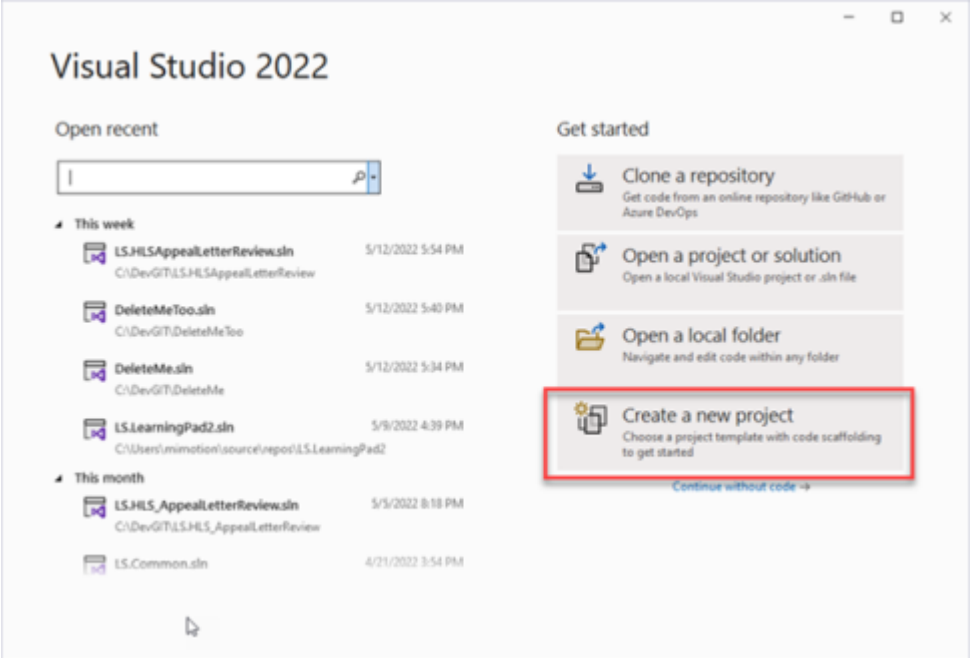
Create Telerik Solution for Legality projects

I'm only building UIs so my work will be here: LS.ProductName\ProjectName.Presentation.

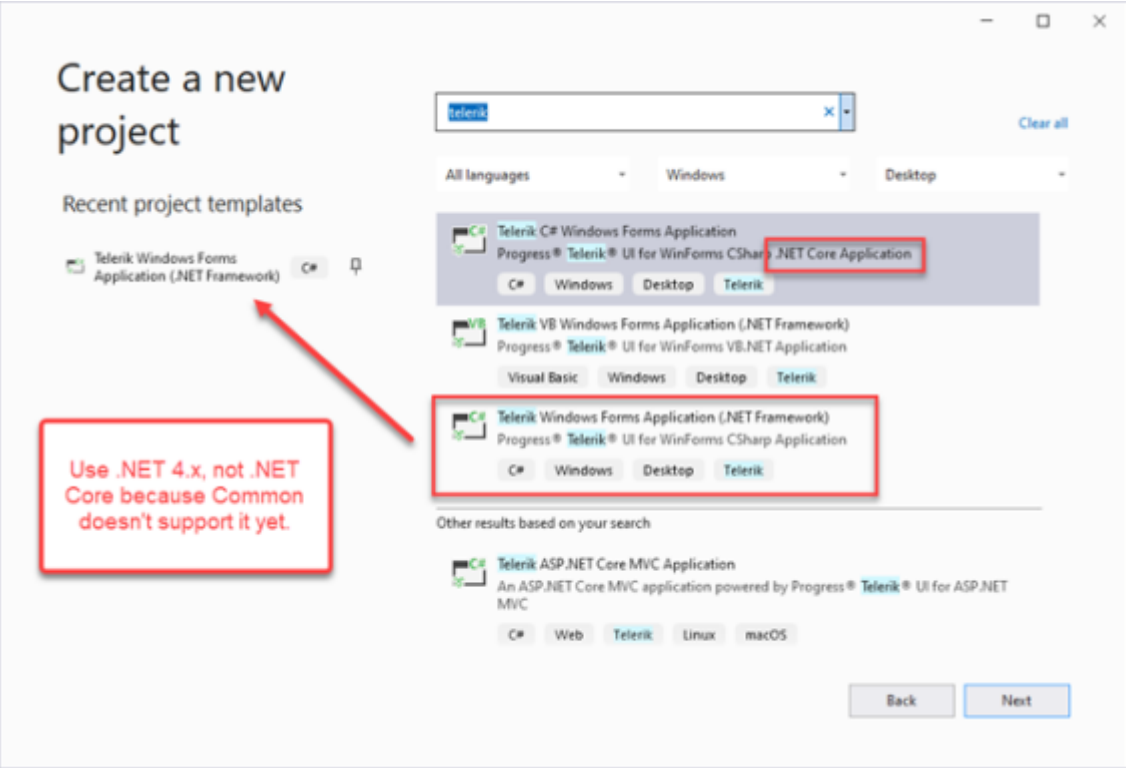
Create a Telerik Solution using the Telerik Wizard

http:%%//%%docs.telerik.com/devtools/winforms/installation-deployment-and-distribution/visual-studio-extensions/create-project

VS 2022



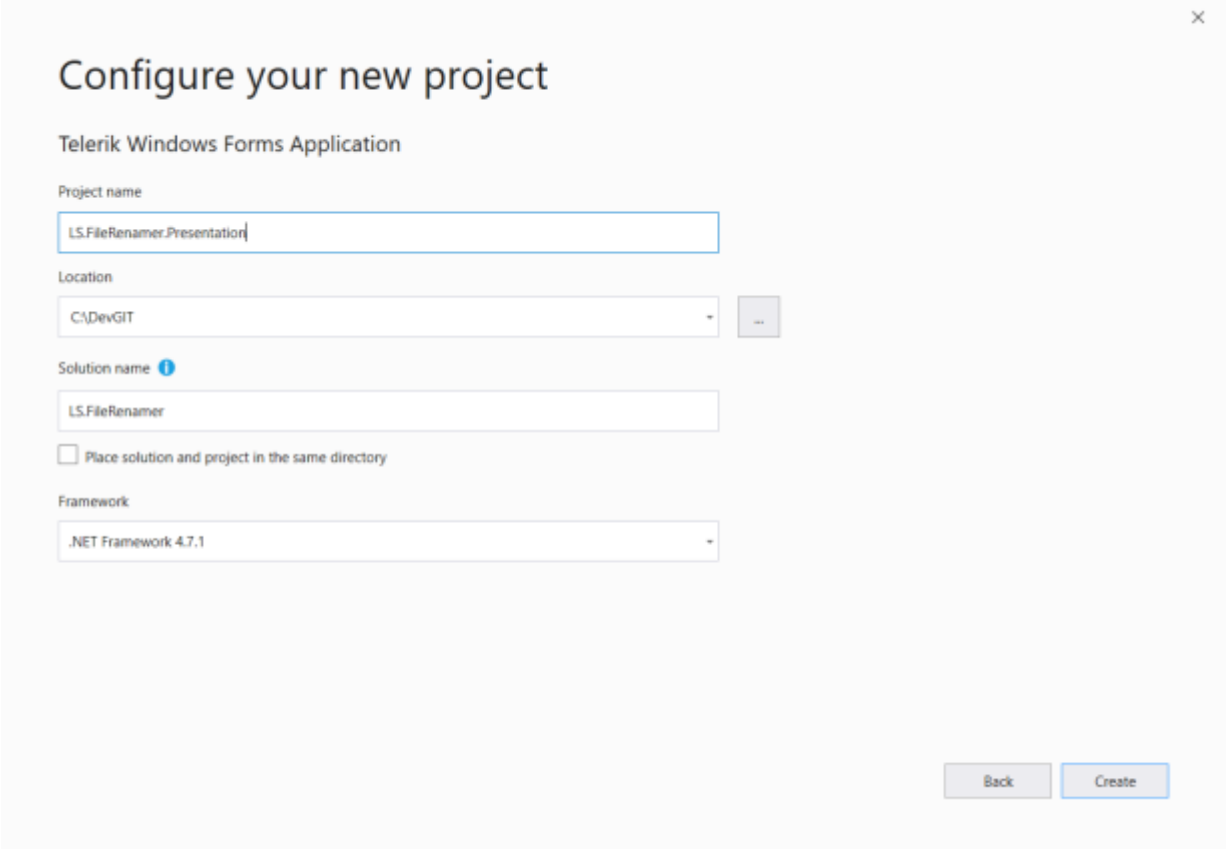
Core vs. Traditional .NET (v4.x)

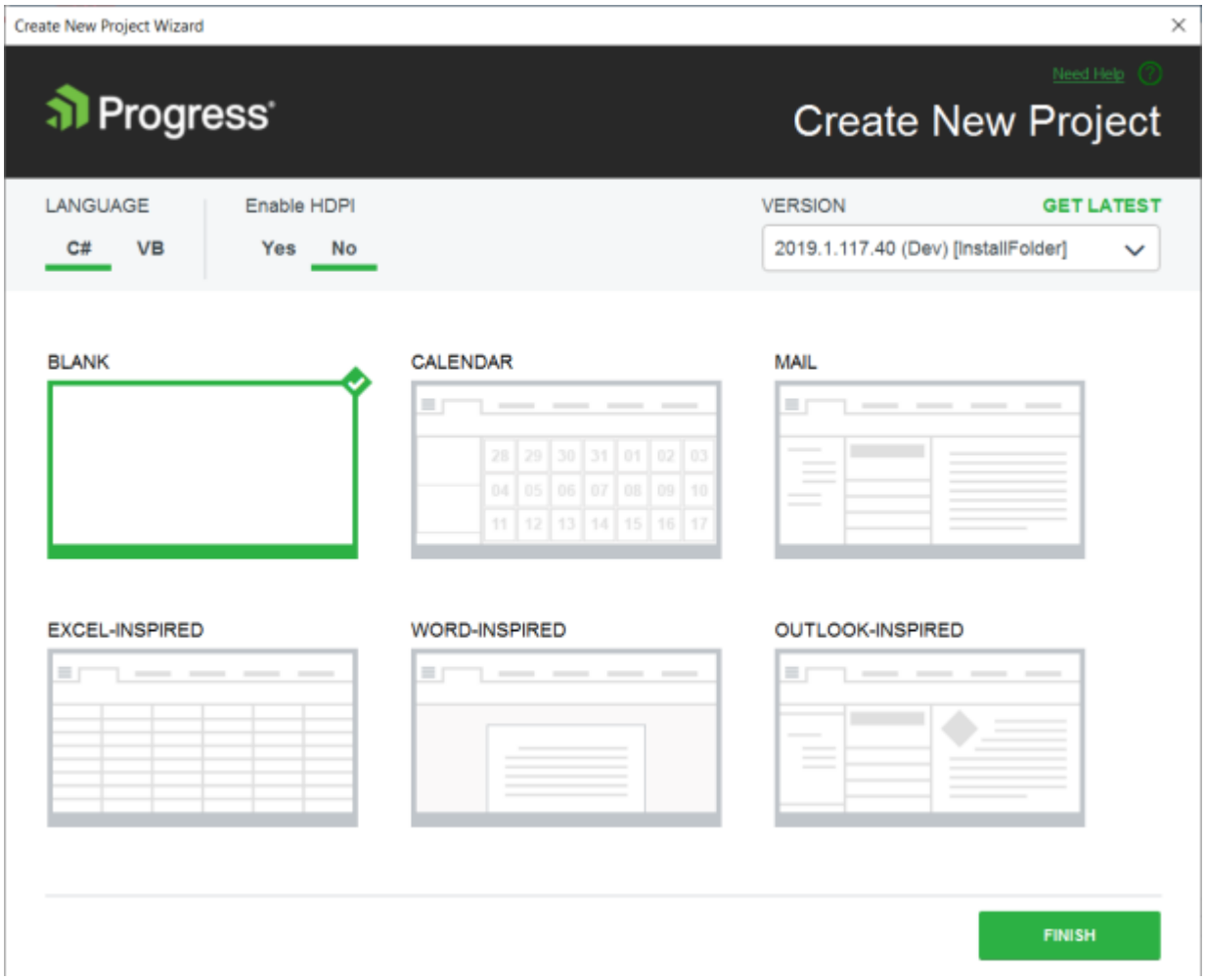


~~UPDATE: Marcel prefers .NET 5 (Core) but you must always use the latest version of the Telerik Nuget packages because they release frequent updates. I can create as regular WinForms project and they can convert it later.~~ ~~**Our Common components are all non-Core anyway so you can NOT use .NET Core for WinForms apps at this time.**~~

~~**UPDATE:** All new projects needs to be created using the .NET Core for WinForms as our commons now supports it.~~

UPDATE: Going back to .NET Framwork projects. There is a bug in VS 2022 that breaks the toolbox for Core controls.

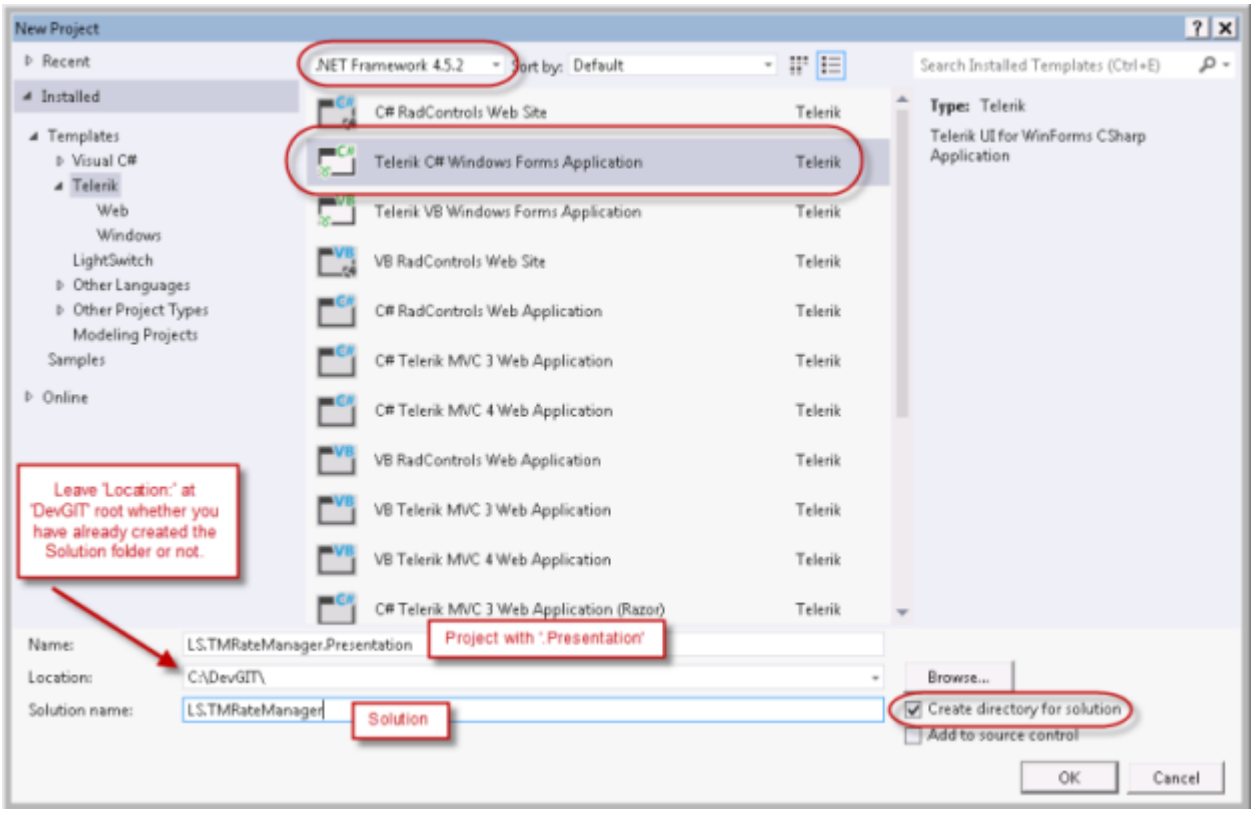

Configure your new project
Telerik Windows Forms Application
Project name
LS.FileRenamer.Presentation
Location
C:\DevGIT
...
Solution name
LS.FileRenamer
Place solution and project in the same directory
Framework
.NET Framework 4.7.1
Back
Create


Create New Project Wizard
Need Help
Progress
Create New Project
LANGUAGE
C#
VB
Enable HDPI
Yes
No
VERSION
GET LATEST
2019.1.117.40 (Dev) [InstallFolder]
BLANK
CALENDAR
MAIL
EXCEL-INSPIRED
WORD-INSPIRED
OUTLOOK-INSPIRED
FINISH

Older VS Versions

1. Click: “Telerik | RadControls for WinForms | Create New Telerik Project”
2. **New Project** dialog

- Choose .NET Framework version from dropdown. (Currently 4.5.2)
- “Templates | Telerik | Windows”
- Choose “C# RadControls Windows Forms Application”
- Enter a Project name like “LS.TelerikControls.Presentation”
- Enter a Solution name like “LS.TelerikControls”
- Enter a root path (“C:\DevGIT”) but don’t specify the solution folder. That will be created when you…
- Check the box to “Create directory for solution”




```
<ol start="3" style="list-style-type: decimal;">
```

```
<li>
```

```
<p>
```

Project Configuration Wizard dialog

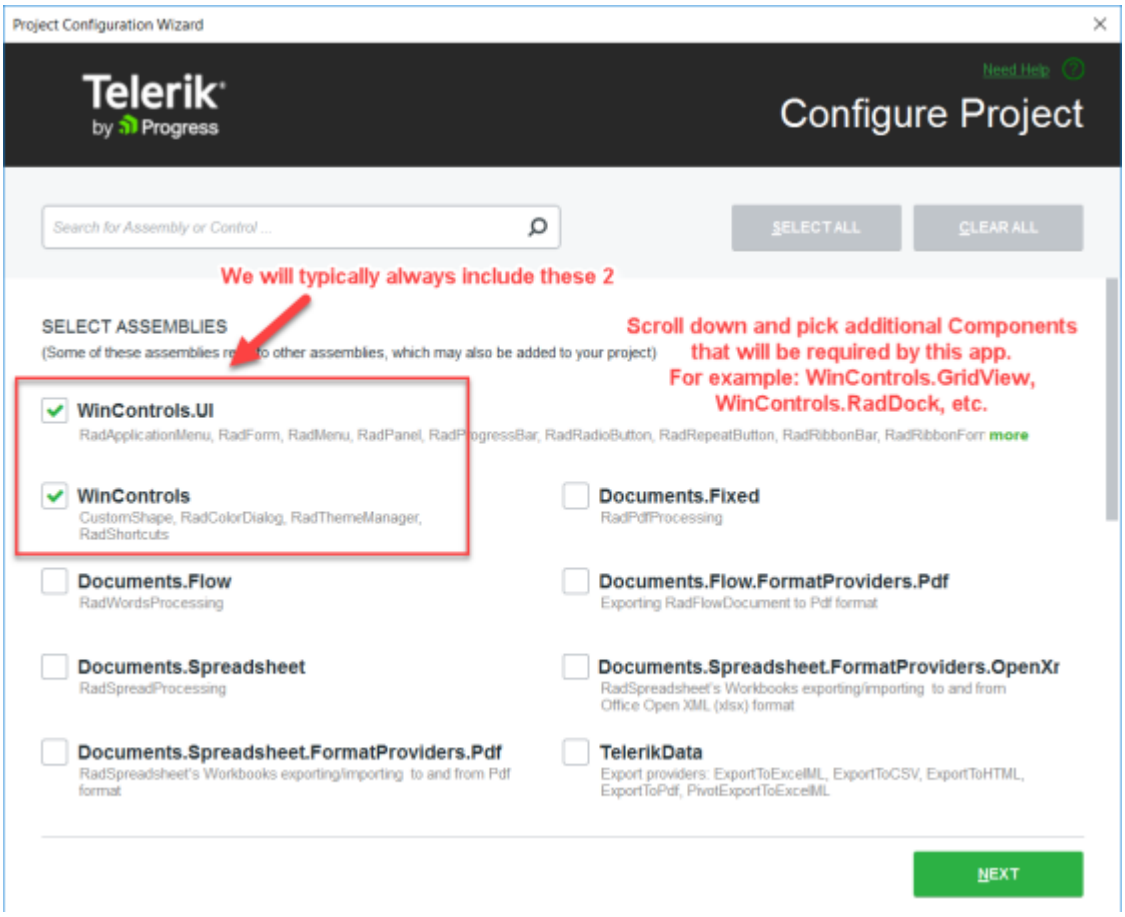
```
</p>
```

```
</li>
```

```
</ol>
```

> Next select the Project and use the Telerik ‘Configuration Wizard’ (“Telerik | UI for WinForms | Configure Project”) to pick the ‘Components’ and ‘Themes’ required by your application.
>
> You can return to this dialog later by: “Telerik | UI for WinForms | Configure Project”.
>
> ### SOMEWHERE WE NEED TO SET THiS. I’m not sure where in the new Telerik version: Choose “UI for WinForms, v. XXX (Dev) [GAC]”. This is our new plan starting with v2015.2.728.40

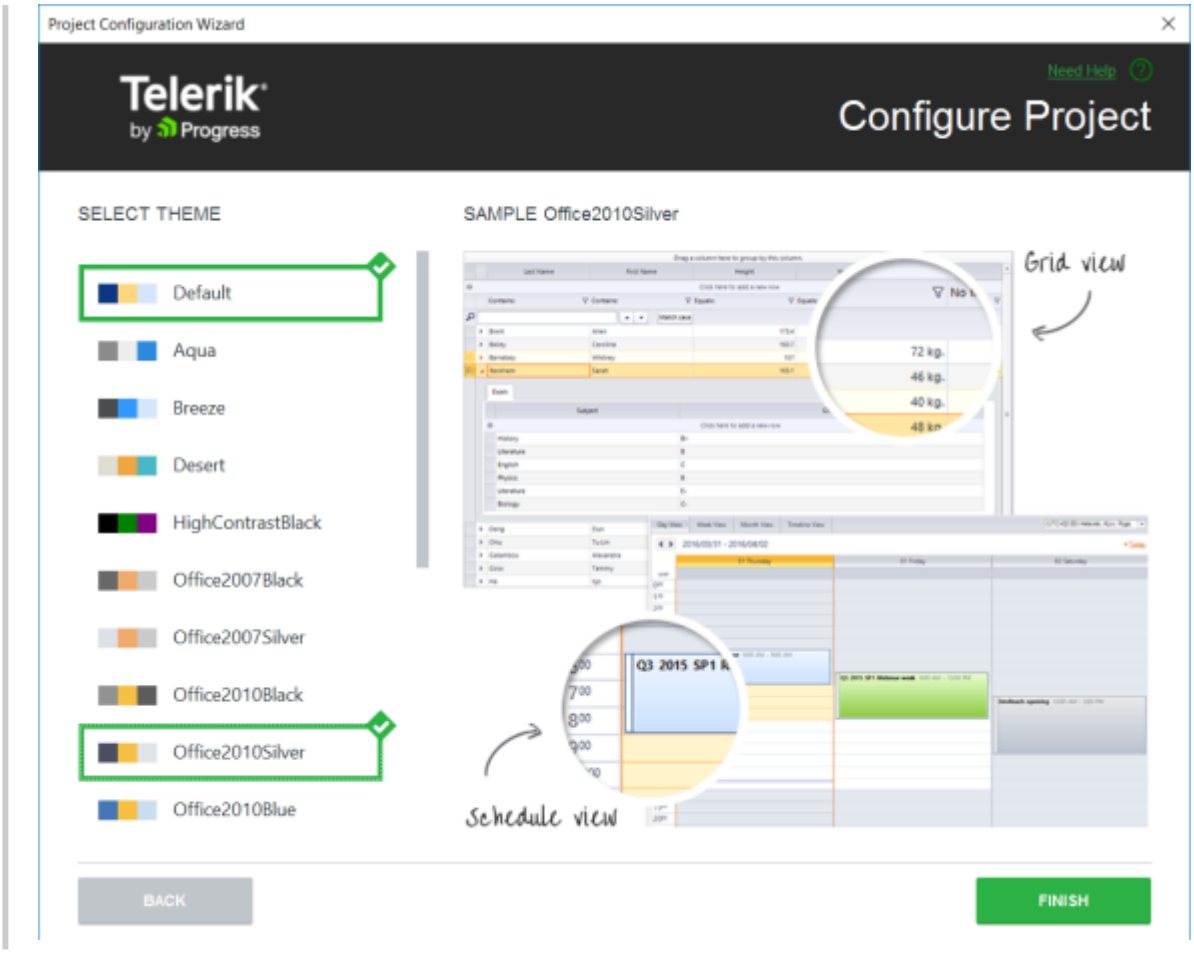
Configure Project

> Choose “UI for WinForms, v. XXX (Dev) [GAC]”. This is our new plan starting with v2015.2.728.40
>
> 
>

Choose Components

- Telerik.WinControls
- Telerik.WinControls.GridView (unless it won't be used)
- Telerik.WinControls.RadDock (unless it won't be used)
- Telerik.WinControls.UI
- *Telerik.Common appears to be included by default*

> **Themes tab**



We use these themes

- Telerik.WinControls.Themes.Desert
- Telerik.WinControls.Themes.Office2010Black
- Telerik.WinControls.Themes.Office2010Blue
- Telerik.WinControls.Themes.Office2010Silver
- Telerik.WinControls.Themes.Office2013Dark
- Telerik.WinControls.Themes.Office2013Light
- Telerik.WinControls.Themes.TelerikMetro
- Telerik.WinControls.Themes.TelerikMetroBlue
- Telerik.WinControls.Themes.Windows7
- Telerik.WinControls.Themes.Windows8

```
<ol start="4" style="list-style-type: decimal;">
```

```
<li>
```

```
<p>
```

Click

```
<HTML></p></HTML><HTML></li></HTML><HTML></ol></HTML>

> //Rename the RadForm1 to rfrmMain//
>
```

```
> //<del>Now you have an **unsaved** Telerik “Project” (not
Solution).</del>//
>
> //<del>It also contains a Windows Form called “Form1”, rather than a
RadForm called “frmMain”.</del>//

<HTML><ol start="5" style="list-style-type: decimal;"></HTML>
<HTML><li></HTML><HTML><p></HTML><del>To save the Solution click “File |
Save All” (or press <Ctrl><Shift><S>)</del><HTML></p></HTML>
<HTML><ol style="list-style-type: lower-alpha;"></HTML>
<HTML><li></HTML><HTML><p></HTML>//<del>Name:
“LS.[ProductProjectName].Presentation”</del>//<HTML></p></HTML><HTML></li></
HTML>
<HTML><li></HTML><HTML><p></HTML>//<del>Location: Where ever you want. Can
move later.</del>//<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>//<del>Solution: “LS.[ProductProjectName]”
- Solution is same as Project, but without the "Presentation", "DataLayer",
etc.</del>//<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>//<del>“Create directory for solution” -
leave checked</del>//<HTML></p></HTML><HTML></li></HTML><HTML></ol></HTML>
<HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>Add or switch to
rfrmMain</del><HTML></p></HTML>
<HTML><ol style="list-style-type: lower-alpha;"></HTML>
<HTML><li></HTML><HTML><p></HTML><del>**Option 1** Click: "Telerik |
RadControls for WinForms | Add RadForm” then choose from the **Add New
Item** dialog</del><HTML></p></HTML>
<HTML><ol style="list-style-type: lower-roman;"></HTML>
<HTML><li></HTML><HTML><p></HTML><del>“Telerik RadForm” or “Telerik
RadRibbonForm”</del><HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>Name it “rfrmMain” and click
<Add></del><HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>Double click the 'Program.cs' in the
Solution Explorer to open it.</del><HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>Open ‘Program.cs’ and in the Main()
function line: "Application.Run(new Form1());"replace the "Form1" with
"rfrmMain" and save ‘Program.cs’.</del><HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>Delete the "Form1.cs" from the
Solution
Explorer.</del><HTML></p></HTML><HTML></li></HTML><HTML></ol></HTML>
<HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>**Option 2** Convert the existing
Windows style Form1.cs to a RadForm</del><HTML></p></HTML>
<HTML><ol style="list-style-type: lower-roman;"></HTML>
<HTML><li></HTML><HTML><p></HTML><del>Expand the ‘Form1’ branch and open the
‘Form1’ with the orange braches icon (the one with no
extension)</del><HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>Change:</del><HTML></p></HTML>
<HTML><ol style="list-style-type: decimal;"></HTML>
<HTML><li></HTML><HTML><p></HTML><del>“public partial class Form1 :
Form”</del><HTML></p></HTML><HTML></li></HTML>
```

```
<HTML><li></HTML><HTML><p></HTML><del>To: “public partial class Form1 :
**Telerik.WinControls.UI.RadForm**”</del><HTML></p></HTML><HTML></li></HTML>
<HTML></ol></HTML>
<HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>Rename main branch (file) ‘Form1’ to
‘rfrmMain’. Confirm the 3 subfiles renamed as
well.</del><HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>Confirm the solution
builds</del><HTML></p></HTML><HTML></li></HTML><HTML></ol></HTML>
<HTML></li></HTML><HTML></ol></HTML>
<HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>Setup rfrmMain<HTML></p></HTML>
<HTML><ol style="list-style-type: lower-alpha;"></HTML>
<HTML><li></HTML><HTML><p></HTML>Open the rfrmMain design view and open the
'Document Outline' view. If you added a RadRibbonForm then it will
automatically include a 'panel1' Panel. You could use this as the main
container, or replace it with a RadDock, etc. If the form doesn't include a
main container then add some type of container (typically RadPanel, or
RadDock if you want dockable panels) as a master container on the form then
dock it to all sides.<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>Set a default form size if you wish, Good
options are:<HTML></p></HTML>
<HTML><ol style="list-style-type: lower-roman;"></HTML>
<HTML><li></HTML><HTML><p></HTML>800x600 rez form size:
750,550<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>1024x768 rez form size:
1000,700<HTML></p></HTML><HTML></li></HTML><HTML></ol></HTML>
<HTML></li></HTML><HTML></ol></HTML>
<HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>Next setup the DLLs and References
//(NOTE: This process suggests setting Copy Local to true, to copy the
installed Telerik DLLs to the build folder, then copying those to the DLLs
folder. This might still cause Telerik version drama, so we might want to
maintain a copy of the preferred version of Telerik DLLs somewhere, then
copy those in to DLLs. Then we would skip steps a-
d.)//</del><HTML></p></HTML>
<HTML><ol style="list-style-type: lower-alpha;"></HTML>
<HTML><li></HTML><HTML><p></HTML><del>Go to the Solution folder
//“LS.[ProductName]”// (not the project) and create a “DLLs”
folder.</del><HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>In the Solution Explorer, expand the
“Project | References” and select all the “Telerik.*” references. In the
“Properties” window, confirm “Copy Local” is set to “True”. //(Also confirm
“Specific Version” is set to “False” ??? Probably not. See Step i
below.)//</del><HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>Click “Build” (F6) to build the
solution. The Referenced DLLs will be copied to the build “\bin\Debug”
folder.</del><HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>Copy all the “Telerik*.dll” files to
the //“LS.[ProductName]\DLLs\”//
folder.</del><HTML></p></HTML><HTML></li></HTML>
```

```
<HTML><li></HTML><HTML><p></HTML><del>Back in theSolution Explorer, expand
the Project | References and select all the Telerik* references and <Del>
(delete) them.</del><HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>Right click the “References” heading
and click “Add Reference…”</del><HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>If there are any existing, unchecked
entries in the “Reference Manager” select them all, right click and click
“Clear Recent References”.</del><HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>Click <Browse>, select all the Telerik
DLLs from the //“LS.[ProductName]\DLLs\”// folder and click <Add>. **This
Project will now use the DLLs stored in the Solution folder. You must also
patch up the references in any other projects added to the Solution
later.**</del><HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>This last step automatically changed
the “Specific Version” setting to
“False”.</del><HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML><del>In the “Properties” window, change the
“Copy Local” setting to “True” because the last step would have changed it
back to False. We do this so the DLLs will be copied to the Build folder
ensuring the build has a a self contained executable version of the
app.</del><HTML></p></HTML><HTML></li></HTML><HTML></ol></HTML>
<HTML></li></HTML><HTML></ol></HTML>

===== Create a Telerik Solution without using the Telerik Wizard =====

<del>Naming Example:</del>

<del>Name: MiM.TelerikControls.Presentation</del>

<del>Location: Where ever you want. Can move later.</del>

<del>Solution: MiM.TelerikControls (Solution is same as Project, but without
the "Presentation", "DataLayer", etc.</del>

<del>Create directory for solution - leave checked</del>

<del>On the "Project Configuration Wizard" dialog (You can return to this
dialog later by: "Telerik | RadControls for WinForms | Configure
Project")</del>

<del>Components</del>

  * <del>Choose "RadControls for WinForms... [InstallFolder]"</del>
  * <del>Telerik.WinControls</del>
  * <del>Telerik.WinControls.GridView (unless it won't be used)</del>
  * <del>Telerik.WinControls.RadDock (unless it won't be used)</del>
  * <del>Telerik.WinControls.UI</del>
  * <del>Telerik.Common</del>

<del>Themes</del>
```

```
  * <del>Telerik.WinControls.Themes.Desert</del>
  * <del>Telerik.WinControls.Themes.Office2010Black</del>
  * <del>Telerik.WinControls.Themes.Office2010Blue</del>
  * <del>Telerik.WinControls.Themes.Office2010Silver</del>
  * <del>Telerik.WinControls.Themes.TelerikMetroBlue</del>
  * <del>Telerik.WinControls.Themes.Windows7</del>
  * <del>Telerik.WinControls.Themes.Windows8</del>

<del>Create frmMain main application form (C#)</del>

  * <del>Click "Telerik | RadControls for WinForms | Add RadForm (or Add
RadRibbonForm)" then choose "Telerik RadForm" (or RadRibbonForm) and click
<Add>.</del>
  * <del>Rename the newly create form to "frmMain.*"</del>
  * <del>Double click the 'Program.cs' in the Solution Explorer to open
it.</del>
  * <del>In the Main() function, find the line: "Application.Run(new
Form1());" and replace the "Form1" with "frmMain"</del>
  * <del>Delete the "Form1.cs" from the Solution Explorer.</del>
  * <del>Open the frmMain design view and open the 'Document Outline' view.
If you added a RadRibbonForm then it will automatically include a 'panel1'
Panel. You could use this as the main container, or replace it with a
RadDock, etc. If the form doesn't include a main container then add some
type of container (typically RadPanel, or RadDock if you want dockable
panels) as a master container on the form then dock it to all sides.</del>
  * <del>Set a default form size if you wish, Good options are:</del>
    * <del>800x600 rez form size: 750x550</del>
    * <del>1024x768 rez form size: 1000x700</del>

<del>Create frmMain main application form (VB)</del>

  * <del>Click "Telerik | RadControls for WinForms | Add RadForm (or Add
RadRibbonForm)" then choose "Telerik RadForm" (or RadRibbonForm) and click
<Add>.</del>
  * <del>Rename the newly create form to "frmMain.*"</del>
  * <del>Right click the Project name and click "Properties" in Solution
Explorer. The Project name is the second line (bolded), not the top line,
which represents the Solution.</del>
  * <del>Change the "Startup Form" from "Form1" to the name of your newly
created Telerik form and click <Save></del>
  * <del>Double click the RadForm in the Solution Explorer to open it.</del>
  * <del>Delete the "Form1.vb" from the Solution Explorer.</del>
  * <del>Open the frmMain design view and open the 'Document Outline' view.
If you added a RadRibbonForm then it will automatically include a 'panel1'
Panel. You could use this as the main container, or replace it with a
RadDock, etc. If the form doesn't include a main container then add some
type of container (typically RadPanel, or RadDock if you want dockable
panels) as a master container on the form then dock it to all sides.</del>
  * <del>Set a default form size if you wish, Good options are:</del>
    * <del>800x600 rez form size: 750x550</del>
    * <del>1024x768 rez form size: 1000x700</del>
```

```
===== Telerik Systray App =====

See this video:
[[file:///\\192.168.93.1\KB_MiM\Telerik\20140820)%20Telerik%20Systray%20App%
20with%20Larry\20140820)%20Telerik%20Systray%20App%20with%20Larry.mp4|\\192.
168.93.1\KB_MiM\Telerik\20140820) Telerik Systray App with Larry\20140820)
Telerik Systray App with Larry.mp4]]

My sample project:
[[file:///\\192.168.93.1\KB_MiM\Telerik\LS.TelerikSystrayApp|\\192.168.93.1\
KB_MiM\Telerik\LS.TelerikSystrayApp]]

==== Steps to create a SysTray app ====

  - You must add a Windows NotifyIcon (call it ‘niMain’) control from the
Toolbox to rfrmMain to make a Systray app.
  - Next add a RadContextMenu (call it ‘rcmMain’) to rfrmMain. This and
niMain won’t appear on the form itself, but in the tray below.
  - Click the tasks arrow button on the ‘rcmMain’ control and click “Edit
Items” and add the menu items you want to to make available from your
Systray icon. Here’s an example from eTMsync:

> {{tdn:image9.png?395x330}} {{tdn:image10.png?234x330}}

<HTML><ol start="4" style="list-style-type: decimal;"></HTML>
<HTML><li></HTML><HTML><p></HTML>Add an icon (must be *.ico) to the
niMain.Icon property.<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>Add click events for the SysTray
(niMain.Items) menu
items.<HTML></p></HTML><HTML></li></HTML><HTML></ol></HTML>

> Select the form that contains the niMain, then choose the rmiItem control
from the properties list.\\
> {{tdn:image11.png?624x464}}
>
> Now double click in the ‘Click’ event to stub out the ‘niMain_MouseDown’
event handler.
>
> {{tdn:image12.png?634x471}}
>
> Insert this code:
>
> %%//%% Note: you must add using Telerik.WinControls.UI; to the top of
rfrmMain.cs
>
> %%//%% pop systray menu on mouse click
>
> private void niMain_MouseDown(object sender, MouseEventArgs e)
>
> {
>
```

```
> %%//%% click left mouse button on SysTray icon to pop the menu
>
> if (e.Button != MouseButtons.Left)
>
> return;
>
> if (rcmMain.DropDown.Visible)
>
> rcmMain.DropDown.ClosePopup(RadPopupCloseReason.CloseCalled);
>
> else
>
> rcmMain.Show(MousePosition);
>
> }

<HTML><ol start="6" style="list-style-type: decimal;"></HTML>
<HTML><li></HTML><HTML><p></HTML>Now hide rfrmMain so the user only sees the
Systray icon. Add this code right after InitializeComponent(); in the public
rfrmMain().\\
%%//%% Hide rfrmMain by default for a SysTray app\\
WindowState = FormWindowState.Minimized; %%//%% minimize rfrmMain\\
ShowInTaskbar = false; %%//%% ...and don't show it on the
TaskBar<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>Open the rfrmMain in design view, select
each of the rcmMain menu items you create from the Properties tool window
dropdown. Switch from Properties to Events and add a ‘Click’ event
handler.<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>### What
next??<HTML></p></HTML><HTML></li></HTML><HTML></ol></HTML>

Per this post you can bind a [[#radcontextmenu-rcm|RadContextMenu]]
(rcmMain) to the NotifyIcon control.
http://www.telerik.com/forums/radcontextmenu-notifyicon

Add an ‘icon’ (must be *.ico) to the Windows NotifyIcon (niMain) control so
it will be visible in the systray.

**Start frmMain hidden**

public frmMain()

{

InitializeComponent();

%%//%% Visible = false; %%//%% hide the form by default - THIS METHOD
DOESN'T WORK. Do this instead:

WindowState = FormWindowState.Minimized; %%//%% minimize frmMain
```

```
ShowInTaskbar = false; %%//%% ...and don't show it on the TaskBar

**Pop the systray menu on mouse click (bind the RadContextmenu to the
NotifyIcon)**

%%//%% pop systray menu on mouse click

void niMain_MouseDown(object sender, MouseEventArgs e)

{

if (e.Button == System.Windows.Forms.MouseButtons.Left) %%//%% click left
mouse button on SysTray icon to pop the menu

rcmMain.Show(MousePosition); %%//%% bind the RadContextmenu to the
NotifyIcon

}

**Show/Hide frmMain on Minimize**

### I’m not sure the following really works. Test it. I had to minimize and
set ShowInTaskbar = false; above.

private void radMenuItem4_Click(object sender, EventArgs e) %%//%% Show
frmMain

{

Visible = true;

this.WindowState = FormWindowState.Normal;

ShowInTaskbar = true; %%//%% ...and show it on the TaskBar

}

private void frmMain_Resize(object sender, EventArgs e) %%//%% Hide frmMain

{

if (WindowState == FormWindowState.Minimized)

Visible = false;

}

====== Visual Studio Info, Shortcuts & Tricks ======

===== Shortcut Keys =====
```

```
Outline View: "View | Other Windows | Document Outline" or <Ctrl><W>, <U>

Code Autoformat <Ctrl><K>,<D> (only works if there are no errors in the
code)

===== Add Resources to a Solution (Icons, Graphics, Files, etc.) =====

Video:
[[file:///\\192.168.93.1\KB_MiM\Telerik\20160204)%20Add%20Icon%20Graphics%20
to%20a%20VS%20Solution%20(Project%20Resource%20File).mp4|\\192.168.93.1\KB_M
iM\Telerik\20160204) Add Icon Graphics to a VS Solution (Project Resource
File).mp4]]

Click the <…> button on the ‘Image’ property of a radCommandBar button item
(for example) and…

{{tdn:image13.png?341x230}}

{{tdn:image14.png?455x256}}

==== Committing newly added Resources to GIT ====

{{tdn:image15.png?503x222}}

===== Our Source File Structure =====

/*

Solution Structure:

- LS.{name}.Library - business logic

- LS.{name} or LS.{name}.Presentation - main application

- LS.{name}.Tests - unit tests

- LS.{name}.{project} - specific implementation of an application concept
that I think should be in its own assembly; for example, database access

Project Structure:

- Contracts / Interfaces - the interfaces; I don't usually use the "I"
prefix as I dislike the (bad) Hungarian notation and also I want *these* to
be the primary concepts in the application

- Models - the (usually) dumb classes, objects that just carry data and
usually have no / very few methods and should have no dependencies

- Implementations / Services - the implementations of the interfaces

*/
```

```
%%//%%===[ Source File Structure ]===

%%//%% TODOs

%%//%% Namespace starts here

%%//%% CLASS STARTS HERE

%%//%% Static methods, if any

%%//%% Events, if any

%%//%% Constructor

%%//%% Public Properties

%%//%% Public Methods

%%//%% <- an empty comment line

%%//%% Private Fields

%%//%% Private Properties

%%//%% Private Methods

%%//%% <- an empty comment line IF there are any event handlers

%%//%% Event Handlers

%%//%% use this ordering:

%%//%% _Load

%%//%% _Shown

%%//%% _Closing

%%//%% _Close

%%//%% CLASS ENDS HERE

%%//%% Namespace ends here

%%//%% DONE TODOs

===== TODO Comments =====

http://www.dotnetperls.com/todo

Create tasks in the source code which appear on the VS ‘Task List’ pane.
```

```
> %%//%%TODO: LARRY - work on version with no bugs

As a coder completes each TODO change the ‘TODO’ to ‘DONE’ and leave the
comment in place to act as future documentation.

These tasks will appear in event handlers and stubbed out methods.

We will keep other tasks that define other functionality in a file named
‘XXXX---Spec-Todos.cs’ in the Project where ‘XXXX’ is the code file that the
Spec-Todos file describes. For example ‘rfrmMain---Spec-Todos.cs’ might
contain TODO entries describing the big steps that must be completed in the
application. It would also contain TODOs describing code for controls which
aren’t responding to an event (i.e. how a form will be setup, or details of
how a GridView is to be configured). Another example of this file could be
‘Program---Spec-Todos.cs’. Here’s anexample of a TODO in a ‘rfrmMain---Spec-
Todos.cs: file:

%%//%%TODO: Marcel - 'ucGridStatusModuleStatusConsole' - Populate the
GridView with these columns in the following order.

%%//%% Each of the values are provided by the plugin (module) via some way
they report their settings and status.

%%//%% Some of these values will be defined on the plugin Options form where
they will be defaulted to design time values.

%%//%% 'Enable' - Checkbox column. Unchecked disables all activity for this
module.

%%//%% 'Category' - A text string name assigned to each category.

%%//%% 'Module' - The module name reported by each plugin.

//NOTE: When adding a ‘XXXX---Spec-Todos.cs’ to a Project right click the
Project | Add | New Item | and choose ‘Code File’ for a blank .cs file, or
simply press <Shift><Alt><C> to add a class.cs file, then delete the stubbed
out contents.//

### At some point we may add date stamps to track the original creation date
and completion date of a TODO.

===== Build =====

==== Set the Build Output Folder ====

  * On the Project menu, click Properties.
  * Click the Build tab.
  * Click the Browse button next to the Output path box and select a new
build output directory.

> Read more here:
```

http://stackoverflow.com/questions/428085/how-do-i-set-a-path-in-visual-studio

==== Remove the Namespace from the Build Output (EXE) File Name ====

Go from something like this: ‘MiM.ForensicDataCollector.Presentation.exe’ to something like this: ‘ForensicDataCollector.exe’

{{tdn:image16.png?489x273|C:\Users\rgray\AppData\Local\Temp\SNAGHTML178b5626.PNG}}

What about our DLL folder plan? How do you adjust References and “Copy Local”?

==== Copy the required DLL’s to the build folder ====

For each Telerik assembly reference in the Solution Explorer, open its context menu, click on properties and in the property grid set the Copy Local property of the reference to True. Thus these Telerik assemblies will be copied to your Release/Bin, Debug/Bin folders

===== Deployment =====

==== Telerik Application Deployment ====

http://www.telerik.com/help/winforms/installation-deployment-and-distribution-application-deployment.html

==== ILMerge Telerik WinForms assemblies ====

http://www.telerik.com/help/winforms/installation-deployment-and-distribution-redestributing-telerik-radcontrols-for-windows.html

When you deploy your application, you may prefer to do it as a single executable rather than an executable referencing many external assemblies. In this case, you need to ILMerge the assemblies with the executable. Here is a link to MSDN from where you can download the ILMerge executable: [[http://www.microsoft.com/download/en/details.aspx?id=17630|Download ILMerge]]

==== Application Icons –Multisize ICO files ====

ICO files can have multiple sizes of the icon image contained within it by using layers. You do this so that the 16x16 size is used for the app's top bar but the 32x32 size version is used when showing open apps via Alt-Tab.

{{tdn:image17.png?545x287}}

Here’s an explaination and some tool options to make multisize ISO files.

http://stackoverflow.com/questions/4354617/how-to-make-get-a-multi-size-ico-

```
file

<del>Here’s a site that will convert PNGs to a multisize ISO file.</del>
Uses Flash so dead now.

<del>http://convertico.org/Multi_Image_to_one_icon/</del>

Upload a larger (<4MB) square aspect ratio PNG file and this will provide a
multi-rez download ICO file.\\
https://www.icoconverter.com/

==== Add updater ====

Need to add a rmi for Check for Updates. The updater.exe will live in the
same location as the application exe file.

private void rmiCheckForUpdates_Click(object sender, EventArgs e)

{

string exeFilePath =
System.Reflection.Assembly.GetExecutingAssembly().Location;

string workPath = System.IO.Path.GetDirectoryName(exeFilePath);

if(File.Exists(workPath + "\\updater.exe"))

{

Process.Start(workPath + "\\updater.exe");

}

else

{

RadMessageBox.Show("Application Update Unavailable",

SystemSettings.AssemblyData.Title, MessageBoxButtons.OK,
RadMessageIcon.Exclamation);

}

}

====== Telerik Common Properties  ======

  *
[[http://www.telerik.com/help/winforms/buttons-mnemonics.html|**UseMnemonic*
*]] is a boolean property which controls whether the & character is
```

```
displayed as the symbol itself or whether it is used to designate a mnemonic
(visualized as an underscore). Its default value is true i.e. mnemonics are
used by default. Specify the character that will be used for a given control
by placing the **& **character before the desired symbol.
  *

====== Forms ======

Use Telerik .NET (4.x) forms to our WinForms projects. LS.Common for
WinForms doesn’t yet support .NET Core.

{{tdn:image18.png?519x360}}

===== Borders =====

> this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None; %%//%%
Control the type of form border

  * None
  * FixedSingle
  * FixedDialog
  * Sizable
  * FixedToolWindow
  * SizableToolWindow

===== TitleBar  =====

> this.ControlBox = false; %%//%% hide the form control buttons
>
> this.Text = String.Empty; %%//%% hide (collapse) the form titlebar
>
> this.Text = " "; %%//%% show the form titlebar but display no text

===== Position =====

**Center a form on screen**

> this.StartPosition = CenterScreen; %%//%% Center the form on screen when
it is opened

  * Manual
  * CenterScreen
  * WindowsDefaultLocation
  * WindowsDefaultBounds
  * CenterParent

CenterToScreen(); %%//%% move the form to center screen while it is already
open

===== Size =====
```

```
For a fixed size window, you still use FormBorderStyle.SizableToolWindow. If
you want to really enforce the size, you could also set MinimumSize equal to
MaximumSize on the form.

**Force a form to a specific size**

int CollapsedWidth = 505; %%//%% default size when collapsed (History is NOT
visible)

int CollapsedHeight = 258;

public rfrmMain()

{

InitializeComponent();

Width = CollapsedWidth; %%//%% set the default collapsed form size

Height = CollapsedHeight;

> …

**Resize a form to display a hidden area based on a Toggle button (from
‘LS.GIT_TM_TimeTracker’)**

> private void rtbtnHistory_ToggleStateChanged(object sender,
StateChangedEventArgs args)
>
> {
>
> if(rtbtnHistory.IsChecked)
>
> {
>
> %%//%% Show History
>
> Width = HistoryWidth; %%//%% Show history
>
> Height = HistoryHeight;
>
> FormBorderStyle = FormBorderStyle.SizableToolWindow; %%//%% allow resize
>
> CenterToScreen();
>
> }
>
> else
>
> {
>
```

```
> %%//%% Hide History
>
> HistoryWidth = Width; %%//%% preserve current size
>
> HistoryHeight = Height;
>
> Width = CollapsedWidth; %%//%% set to default collapsed form size
>
> Height = CollapsedHeight;
>
> FormBorderStyle = FormBorderStyle.FixedToolWindow; %%//%% disable resize
>
> CenterToScreen();
>
> }
>
> }
```

```
===== AutoScaleMode =====

Windows application Forms and UserControls must use the same autoscalemode.
Different settings may cause the Form or UserControl (and their contents) to
scale/draw incorrectly at runtime. This issue may not be visible in Visual
Studio. Always use **AutoScaleMode.None** for the default setting within any
design work.

{{tdn:image19.png?199x77}}

For example:

{{tdn:image20.png?263x288|C:\Users\rober\AppData\Local\Temp\SNAGHTML1e52e6fa
.PNG}}

Use Search/Replace to correct this solution-wide.

{{tdn:image21.png?209x291}}

===== frmMain =====

The primary form they see when the app opens (after any splash screen, etc.
Others are frmAbout, etc.)

===== Telerik Form Templates =====

http://docs.telerik.com/devtools/winforms/forms-and-dialogs/templates/templa
tes

The Visual Studio templates that install with Telerik UI for for WinForms
let you add RadForm and ShapedForm to your application without any coding
steps. There are also two other specialized forms, RadRibbonForm that
contains a built-in RadRibbonBar and RadAboutBox that is an enhanced,
```

```
themeable version of the regular AboutBox.

{{tdn:image22.png?391x273}}

===== RadForm - rfrm =====

http://docs.telerik.com/devtools/winforms/forms-and-dialogs/form/form\\
http://www.telerik.com/help/winforms/forms-and-dialogs-form-getting-started.
html\\
http://www.telerik.com/community/forums/winforms/forms-and-dialogs.aspx

The RadForm control is an extended standard Windows Form that fully supports
the Telerik Presentation Framework (TPF) and the Telerik's theming
mechanism. The control is built of a RadTitleBar component and a border that
can be easily designed in the Visual Style Builder. RadForm also supports
MDI (Multiple Document Interface).

{{tdn:image23.png?374x268}}

//Note: RadForm supports just one shape. See ShapedForm for more advance
shapes.//

  * 'AutoSize' \Layout
  * 'Size' \Layout
  * A good default main form size is 1000x700. SOME OTHER PROPERTY blocks a
new Size entry, so drag the form to resize, then you can enter a specific
size.
  * You can disable resizing a form in a couple of ways:
    * Set the ‘MinimumSize’ and ‘MaximumSize’ to the same value, and set the
‘MaximizeBox’ to False
    * ‘FormBorderStyle’ to ‘FixedSizeToolWindow” (or one of the other
options) and set ‘ControlBox’ to False. (Don’t set the ‘FormBorderStyle’ to
‘None’ or you can’t move the window.
  *

===== RadRibbonForm =====

http://www.telerik.com/help/winforms/ribbonbar-overview.html

http://www.telerik.com/community/forums/winforms/forms-and-dialogs.aspx

http://www.telerik.com/community/forums/winforms/ribbonbar.aspx

The RadRibbonForm control is designed to host a RadRibbonBar control and
mimic the Microsoft Office 2007 UI form style. This control automatically
detects whether it runs under Windows Vista and Desktop Window Manager
Effects are enabled and adjusts itself to make use of these effects just as
Microsoft Office 2007 applications do.

{{tdn:image24.png?357x274}}
```

```
Really the thing to do here is
[[http://docs.telerik.com/devtools/winforms/forms-and-dialogs/ribbonform/get
ting-started|add a RadRibbonForm to the Solution rather than adding a
RadRibbonBar to a regular RadForm]].

A RadForm preloaded with a RadRibbonBar, a “System.Windows.Form.Panel” in
the middle and a RadStatusStrip on bottom. You can drop the panel and
replace with a RadPanel if you want.

See [[#radribbonbar-rrb|RadRibbonBar]] for more info

For a RadRibbonForm the 'Document Outline' view will show:

  * RadRibbonForm1
    * panel1
    * radStatusStrip1
    * radRibbonBar1

//Notice the order. The panel1 is the main container and is followed by the
radStatusStrip1 with the radRibbonBar1 at the bottom. Do not change this
order!//

===== RadStatusStrip – rss  =====

http://www.telerik.com/help/winforms/forms-and-dialogs-statusstrip-overview.
html

http://www.telerik.com/community/forums/winforms/forms-and-dialogs.aspx

RadStatusStrip provides a status area that can be composed of any number of
RadElement types: buttons, repeat buttons, image buttons, labels, panels,
progress bars and separators. The elements can be laid out along the
horizontal or wrapped to additional rows.

{{tdn:image25.gif?449x151}}

The following element types can be added to RadStatusStrip.Items collection:

  * RadLabelElement
  * CommandBarSeparator
  * RadButtonElement
  * RadCheckBoxElement
  * RadImageButtonElement
  * RadProgressBarElement
  * RadRadioButtonElement
  * RadRepeatButtonElement
  * RadSplitButtonElement
  * RadStatusStripPanelElement
  * RadToggleButtonElement
  * RadTrackBarElement
  * RadWaitingBarElement
```

```
**Properties**

  * **LayoutStyle** property handles the arrangement of items when they
exceed the horizontal space in the status bar.
  * **Stack** (the default) arranges items from left to right and the
rightmost items that exceed the horizontal space are not visible.
  * **Overflow** moves items in excess of the horizontal space to the
following line.
  * By default the grip is Hidden. The image below shows **GripStyle** set
to Visible.

If you Dock a panel (or TableLayoutPanel) control to the form and it appears
behind the RadStatusStrip, collapse the panel in the Document Outline, then
drag it above the RadStatusStrip like this:\\
{{tdn:image26.png?247x147|C:\Users\rgray\AppData\Local\Temp\SNAGHTML221ed26e
.PNG}}

===== RadAboutBox =====

http://www.telerik.com/help/winforms/forms-and-dialogs-templates-radaboutbox
-overview.html\\
__http:%%//%%www.telerik.com/products/winforms/aboutbox.aspx__

This isn’t a control in the toolbox. It is a form template you can add to a
project. Like any other form, the RadAboutForm may be resized and configured
in the IDE.

The Telerik version looks just like the standard version that comes with VS.

The form contains a TableLayoutPanel with a logoPictureBox (image) in the
left column, an <OK> on the lower right and the following rows in the right
column:

  * Product Name (RadLabel)
  * Version (RadLabel)
  * Copyright (RadLabel)
  * Company Name (RadLabel)
  * Description (multiline RadTextBox)

These text values are populated from the “Project | Properties | Application
| <Assembly Information> | Title”.

The form title bar is populated from the “Project | Properties | Application
| <Assembly Information>”.

NOTE: The default application icon is also set on the “Project | Properties
| Application” tab.

The logoPictureBox is ‘Size’: 122, 259 with ‘SizeMode’: StretchImage. The
‘logoPictureBox.Image’ property defines the graphic. The default Telerik
sample image size is 120x262.
```

```
==== Add a RadAboutBox to a Project ====

  * Click “TELERIK | RadControls for WinForms | Add RadAboutBox”
  * Name the form “rfrmRadAboutBox” and click <Add>

The code behind “rfrmRadAboutBox”automatically contains some default code.
I’ve modified it here for our purposes so paste this instead):

public rfrmRadAboutBox()

{

InitializeComponent();

> %%//%% Initialize the AboutBox to display the product information from the
assembly information.
>
> %%//%% Change assembly information settings for your application through
either of these options::
>
> %%//%% - Project | Properties | Application | <Assembly Information>
>
> %%//%% - AssemblyInfo.cs

%%//%% Set default row heights

this.tableLayoutPanel.RowStyles[0] = new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute, 20F);

this.tableLayoutPanel.RowStyles[1] = new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute, 20F);

this.tableLayoutPanel.RowStyles[2] = new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute, 20F);

this.tableLayoutPanel.RowStyles[3] = new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute, 30F);

%%//%% Text = SystemSettings.AssemblyData.Product + " by Legality Software";
%%//%% About form title from assembly

Text = "Product Name" + " by Legality Software"; %%//%% About form title
hardcoded

%%//%% radLabelProductName.Text = SystemSettings.AssemblyData.Product;
%%//%% Product name from assembly

radLabelProductName.Text = "Product Name"; %%//%% Product name hardcoded

radLabelVersion.Text = SystemSettings.GetInstallVersion(); %%//%% Shows ver
per installer, not assembly. Shows '???' when run w/o installer
```

```
%%//%% radLabelCopyright.Text = SystemSettings.AssemblyData.Copyright;
%%//%% Pull the Copyright from the Assembly

radLabelCopyright.Text = "Copyright © " +
Convert.ToString(DateTime.Now.Year); %%//%% Make the Copyright always show
the current year

%%//%% radLabelCompanyName.Text = SystemSettings.AssemblyData.Company;
%%//%% pull the company name from the Assembly

radLabelCompanyName.Text = "Legality Software"; %%//%% Hard coded company
name

%%//%% radTextBoxDescription.Text = SystemSettings.AssemblyData.Description;
%%//%% Use the assembly info

radTextBoxDescription.Text = "This is a long line of " + %%//%% Hard coded
description text

"text continued over multiple lines of code." + Environment.NewLine +
Environment.NewLine +

"This line continues after a new line space. Time Matters®";

}

//Insert the Trademark ® by holding <Alt> and typing 0174 on Numpad.//

NOTE: To access the Assembly info (for Product Name, etc), you must start by
adding a Project Reference to:

//‘X:\DevGIT\ls.common\LS.Common.Utilities\bin\Debug\LS.Common.Utilities.dll
’//

Then add using LS.Common; %%//%% provide access to the Assembly to any .cs
file that requires access to the Assembly.

Alternatively, you can also set the defaults in the Form Load event.

private void rfrmRadAboutBox_Load(object sender, EventArgs e)

{

Text = $"About {SystemSettings.AssemblyData.Title}"; %%//%% AboutBox
form.text title from Assembly text ("About %title%")

}

=== Example About Box Description for our custom applications ===

/* Example description for a custom application
```

```
radTextBoxDescription.Text = "This custom application was developed by
Legality Software" + Environment.NewLine + Environment.NewLine +

"for the exclusive use of " + Environment.NewLine + Environment.NewLine +

"Healthcare Legal Solutions, LLC" + Environment.NewLine +
Environment.NewLine + Environment.NewLine + Environment.NewLine +

"Use by any other entity is prohibited without permission from Legality
Software."; %%//%% Hard coded description text

*/

{{tdn:image27.png?488x316}}The RadAboutBox pulls its info from the Assembly
information (from the Project Properties).

Per the comment you can edit the AboutBox content by

> o Changing the Assembly information either directly in the IDE
>
> o By editing the \Properties\AssemblyInfo.cs file.

o Or you could override the Assembly info above and hard code whatever you
want.

**Display the RadAboutBox form like any other:**

private void rmiAbout_Click(object sender, EventArgs e)

{

using (rfrmRadAboutBox rfrmRadAboutBox = new rfrmRadAboutBox()) %%//%%
create an instance of the form

{

rfrmRadAboutBox.ShowDialog(this); %%//%% then show/opens it

}

}

{{tdn:image28.png?348x262}}

Note the text from the Assembly Info dialog above

**Change the AboutBox Picture**

{{tdn:image29.png?720x323}}

In the design view, click the picture box element and view properties.
```

```
Select to change the image and select project resources. Import image like
below.

{{tdn:image30.png?720x452}}

===== RadShapedForm – rfrmshp =====

http://docs.telerik.com/devtools/winforms/forms-and-dialogs/shapedform/shape
dform

The ShapedForm control allows you to design and display a Windows form with
any conceivable shape. You can couple this control with the RadTitleBar
control to easily add forms with a completely custom look and feel.

{{tdn:image31.png?189x186}}

====== Control Notes & Naming Conventions ======

===== Common Control Properties =====

Dock - Property on various controls to fill a space (container)

'Dock' \Layout

Anchor - Anchor to any or all 4 sides of container to auto-resize with
container. Works well within a TableLayoutPanel control.

'Anchor' \Layout

===== Windows Controls (Non-Telerik) =====

==== LinkLabel – llbl ====

**NOTE: Instead of LinkLabel see the <Links> button on the ‘MarkupEditor’
behind the ‘[[#radlabel-rlbl|RadLabel]].Text’ property dropdown. Enter a url
there and the link works with no extra code.**

  * ‘LinkColor’ = 52, 84, 134 to match Telerik Wizard Help link
  * ‘BackColor’ = 208, 225, 251 to match the default theme

[[#linklabel-clicked-launches-
httpwww.legalitysoftware.com-in-browser---c|Here’s code]] to open the
default browser to a site on LinkLabel click

==== * TableLayoutPanel –tlp ====

Not a Telerik control. Use table to specify control layout on form.

Quick Tasks Button

  * 'Add Column', 'Add Row', 'Remove Last Column', 'Remove Last Row' - use
```

```
these to build desired table layout

'Edit Rows and Columns'

'Columns'

<HTML><ul></HTML>
<HTML><li></HTML>> <HTML><p></HTML>'Percent' set to 100% even if another
Column occupies an 'Absolute' amount of space<HTML></p></HTML>
<HTML></li></HTML><HTML></ul></HTML>

'Rows'

<HTML><ul></HTML>
<HTML><li></HTML>> <HTML><p></HTML>**'Absolute' set to 30 pixels high to
hold regular buttons at 24 pixels high**<HTML></p></HTML>

<HTML><ul></HTML>
<HTML><li></HTML><HTML><p></HTML>**Use 40 pixels high for the buttons on
bottom of a form**<HTML></p></HTML><HTML></li></HTML><HTML></ul></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>'Percent' set to 100% even if another Row
occupies an 'Absolute' amount of space<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>'ColumnSpan', 'RowSpan' - allow a control
to span multiple Columns/Rows<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>'Anchor' - align or stretch a control
within a cell (NOTE: Larry prefers
'Dock')<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>'Dock' - stretch the TableLayoutPanel it's
parent container<HTML></p></HTML><HTML></li></HTML><HTML></ul></HTML>

{{tdn:image32.png?406x79}}

%%//%% Columns and rows are 0 based so this references the 2nd column:
TableLayoutPanel1.ColumnStyles[1]

%%//%% Hide an entire TableLayoutPanel - good to hide a subpanel in a single
master tlp cell\\
tableLayoutPanel1.Visible = false;

%%//%% Hide/Show TableLayoutPanel row/column (works for SizeType = Absolute
(pixels) or Percent)\\
tableLayoutPanel1.RowStyles[8].Height = 0; %%//%% Hide row 8
(9<sup>th</sup>) by setting height to 0

tableLayoutPanel1.RowStyles[8].Height = 33; %%//%% Show row 8
(9<sup>th</sup>) by setting height to 33 (good size for a button)

TableLayoutPanel1.ColumnStyles[1].Width = 0; %%//%% Hide column 1
(2<sup>nd</sup>) by setting width to 0
```

```
Create a user control property that shows/hides a column based on the size
of 2 buttons inside a child layout panel.

[Category("Legality Software")]

public bool ViewsExpandCollapseAllVisible

{

get { return tlpQueryListTree.ColumnStyles[2].Width ==
rbtnTreeExpandAll.Width + rbtnTreeCollapseAll.Width + 14; } %%//%% return
true if col currently as wide as buttons + 14px buffer

set { if (value)

tlpQueryListTree.ColumnStyles[2].Width = rbtnTreeExpandAll.Width +
rbtnTreeCollapseAll.Width + 14; %%//%% Show the column (set size to width of
buttons + 14px buffer)

else

tlpQueryListTree.ColumnStyles[2].Width = 0; %%//%% Hide the column (set
width = 0)

}

}

Also see [[#setup-properties-configurable-in-the-vs-ide|Setup Properties
Configurable in the VS IDE]] for more show/hide column/row options as uc
Properties via the IDE

%%//%% Set row/column Size Type to Absolute or Percent

tableLayoutPanel1.RowStyles[0].SizeType = SizeType.Percent;

tableLayoutPanel1.RowStyles[0].SizeType = SizeType.Absolute;\\
TableLayoutPanel1.ColumnStyles[1].SizeType = SizeType.Absolute;

=== Issues Moving Controls to Different TableLayoutPanel Cells ===

Whenever you try to drag a control or block of controls from one
TableLayoutPanel cell to another you may find the controls land in the wrong
cell. Often creating a new tlp rox and dropping the controls there.

This is because the controls you are moving probably have a 'ColumnSpan' or
'RowSpan' property setting that causes the control(s) to occupy more than
one cell. When you drag it VS creates a new tlp row to hold the extra cells
required by the Span property.

The solution is to remove any 'ColumnSpan' or 'RowSpan' settings
```

```
==== RadLayoutControl – rloc ====

http://www.telerik.com/help/winforms/layoutcontrol-overview.html

With RadLayoutControl you can quickly design and arrange your controls in
complex layouts and it will automatically keep the layout consistent at run-
time. RadLayoutControl has an intuitive and straight-forward design time
experience. When resizing the form, it keeps its layout consistent by
proportionally resizing the controls in it while considering their MinSize
and MaxSize settings. RadLayoutControl also allows end-user customizations
and Save/Load layout via the Customize context menu.

{{tdn:image33.png?379x332}}

Main features of the control:

  * Create complex layouts where the controls are automatically resized
according to the current available size.
  * Arrange the controls in the form at runtime - your users will be able to
rearrange the controls according to their needs.
  * Save and load the layout.
  * Group the underlying controls in tabs.
  * Expand/Collapse groups.

==== ToolTip – ttip  ====

Drag this regular Windows (non-Telerik) control to any spot on the form (it
will not be visible). The name doesn’t matter.\\
It will add a ‘**ToolTip on toolTip**’ property to the Rad Controls that
don’t already have a ‘ToolTip’ property.

==== PictureBox  ====

Windows Common Control shows a picture.

  * 'Image' <...> or Quick Tasks Button
    * Choose "(o) Project resource file" then <Import> (copies the picture
into the project)
  * 'SizeMode' = 'AutoSize' to automatically stretch the image.

====== Telerik – Dialogs ======

==== RadColorDialog ====

http://www.telerik.com/help/winforms/forms-and-dialogs-colordialog-overview.
html

==== RadMarkupDialog ====

//RadMarkupEditor// and //RadMarkupDialog// require a reference
to //Microsoft.mshtml// assembly and if you need to use them at Run Time you
```

```
will need to distribute this assembly to the end user computers as well. You
can find further information about that in this //external// resource:

http://msdn.microsoft.com/en-us/library/w0dt2w20.aspx

You do not need to do anything on your developer machine, because Visual
Studio provides the required assembly. Also if your application does not use
the editor or the dialog, you do not need a reference to this assembly.

==== * RadAboutBox ====

http://www.telerik.com/help/winforms/forms-and-dialogs-templates-radaboutbox
-overview.html

This isn’t really a control in the toolbox. It is a form template you can
add to a project.

**For more info see this part of the doc: [[#radaboutbox|RadAboutBox]]**

==== * RadMessageBox ====

http://www.telerik.com/help/winforms/forms-and-dialogs-messagebox-overview.h
tml

http://www.telerik.com/help/winforms/forms-and-dialogs-messagebox-parameters
.html

RadMessageBox displays a modal dialog box that contains a system icon, a set
of buttons, and a brief application-specific message, such as status or
error information. You cannot create a new instance of the RadMessageBox
class. To display a message box, call the static method RadMessageBox.Show.
The title, message, buttons, and icons displayed in the message box are
determined by parameters that you pass to this method.

{{tdn:image34.png?225x115}}

=== Parameters ===

  * Parent - An implementation of System.Windows.Forms.IWin32Window that
will own the RadMessageBox.
  * Text - The text to display in the RadMessageBox.
  * Caption - The text to display in the title bar of the RadMessageBox.
  * Buttons - One of the MessageBoxButtons enumeration values that specifies
which buttons to display in the message box:
    * AbortRetryIgnore
    * OK
    * OKCancel
    * RetryCancel
    * YesNo
    * YesNoCancel
  * Icon - One of the RadMessageIcon enumeration values that specifies which
```

```
icon to display in the message box or a custom Bitmap icon that will be
displayed:
    * RadMessageIcon.None
    * RadMessageIcon.Info
    * RadMessageIcon.Question
    * RadMessageIcon.Exclamation
    * RadMessageIcon.Error
  * defaultBtn - One of the MessageBoxDefaultButton enumeration values the
specifies the default button for the message box:
    * MessageBoxDefaultButton.Button1
    * MessageBoxDefaultButton.Button2
    * MessageBoxDefaultButton.Button2
  * Rtl - RightToLeft settings:
    * RightToLeft.No
    * RightToLeft.Yes
    * RightToLeft.Inherit

=== Details Section ===

RadMessageBox supports details section, shown by specifying the details text
in the Show method parameters:

RadMessageBox.Show("Message text", "Title/Caption Text",
MessageBoxButtons.AbortRetryIgnore, "Details Text");

{{tdn:image35.png?265x132|forms-and-dialogs-messagebox-parameters 001}}

=== Examples ===

using Telerik.WinControls; %%//%% required by RadMessageBox

using System.Windows.Forms; %%//%% required by RadMessageBox button names

private void radButton1_Click(object sender, EventArgs e)

{

RadMessageBox.SetThemeName("Desert");

DialogResult ds = RadMessageBox.Show(this, "Are you sure?", "Title",
MessageBoxButtons.YesNo, RadMessageIcon.Question);

this.Text = ds.ToString();

}

RadMessageBox.Show("I’ll wait till you click <OK>…", "Hold on for a bit",
MessageBoxButtons.OK, RadMessageIcon.None); %%//%% Just hold on a bit

RadMessageBox.Show("Here’s how to stretch a "
```

```
+ "long string across "

+ "multiple lines.", "Help", MessageBoxButtons.OK, RadMessageIcon.None);

if (RadMessageBox.Show("Exit TM Field Utility?", "TM Field Utility",
MessageBoxButtons.YesNo, RadMessageIcon.Question) == DialogResult.No)

e.Cancel = true;

==== RadColorDialog  ====

http://docs.telerik.com/devtools/winforms/forms-and-dialogs/colordialog/colo
rdialog

RadColorDialog is a lightweight UI component that allows users to select
from RGB or HEX color models. The color palettes are skinnable and highly
configurable. It offers a great amount of flexibility in selecting colors
from web, system, and basic colors together with the ability to fine-tune
and pick colors directly from the screen.

{{tdn:image36.png?294x236}}

====== Telerik – Containers ======

Start every form with some kind of container. Could be a RadPanel at
minimum.

==== * RadDock – rd ====

http://www.telerik.com/help/winforms/dock-overview.html

http://www.telerik.com/community/forums/winforms/dock.aspx\\
http://www.telerik.com/help/winforms/dock-object-model-tabs-and-captions.htm
l__\\
__http://www.telerik.com/help/winforms/dock-architecture-and-features-unders
tanding-raddock.html__\\
__Video:
http://www.telerik.com/videos/winforms/getting-started-with-raddock-for-winf
orms\\
Video:
http://www.telerik.com/videos/winforms/introducing-the-new-raddock-for-winfo
rms

Container for collapsable/dockable panels called ‘ToolWindows’ (like the
"Properties" panel in VS). //This is what holds the queries in SQLL - NOT
TRUE???.// Once the container exists you use "Dock New Window to Right
(etc)" from the Quick Tasks Button. Also try “Show Advanced Layout
Designer”.

A new RadDock can directly hold a DocumentContainer and ToolTabStrips which
contain ToolWindow(s) and RadSplitContainers as required. The
```

```
DocumentContainer holds a DocumentTabStrip which in turn holds
DocumentWindow(s). You must add at least one of these before you can add
other objects.

=== SizeMode Info ===

http://www.telerik.com/help/winforms/dock-object-model-example-building-an-a
dvanced-layout-at-runtime.html

http://www.telerik.com/forums/tabstrip-sizeinfo-not-changing-the-size\\
%%//%% You must set both SizeMode and AbsoluteSize properties

window4.TabStrip.SizeInfo.SizeMode = SplitPanelSizeMode.Absolute;

window4.TabStrip.SizeInfo.AbsoluteSize = new Size(150, 0); %%//%% absolutely
150 pixels wide

window2.TabStrip.SizeInfo.SizeMode = SplitPanelSizeMode.Relative;

window2.TabStrip.SizeInfo.RelativeRatio = new SizeF(0, 0.33f); %%//%%
relatively 1/3 of the other window

=== RadDock Properties (parent container) ===

You will probably have to select the Dock control from the VS ‘Document
Outlline’

  * this.radDock1.DocumentTabsAlignment = TabStripAlignment.Left; %%//%%
Show the documentWindow tab strip on the left edge
  * this.radDock1.DocumentTabsTextOrientation %%//%%
  * this.radDock1.DocumentTabsVisible = False; %%//%% Hide the tabs for all
documents
  * this.radDock1.DragDropAllowedStates = All %%//%% Docked, TabbedDocument,
Hidden, AutoHide, Floating

  * **DocumentTabsVisible** – hide document tabs – good if you want to only
show a single document window as the main app window
  * DocumentTabsAlignment – Top, Bottom, etc
  * DocumentTabsTextOrientation
  * MainDocumentContainerVisible – hide all documents in this container
    * this.radDock1.MainDocumentContainerVisible = false;
  * Orientation – Horiz/Vertical
  * **ShowDocumentCloseButton** – close button(s) on each document tab
  * ShowDocumentPinButton – whether DocumentTabStrip instances display a
stickpin button
  * ShowItemToolTips –
  * ShowToolCloseButton – Control whether ToolTabStrip instances will
display Close Button next to each item
  * **SplitterWidth** – default = 3, how prominent is the slplitter bar and
control? Set to 0 for no splitters.
  * ToolTabsVisible
```

```
  * ToolTabsAlignment – Top, Bottom, etc
  * ToolTabsTextOrientation

> {{tdn:image37.png?488x322}}

DocumentContainer – rddc (child container)

Contains all the DocumentWindow objects, which consist of:

> DocumentTabStrip – rddts (grandchild control)
>
> Tabs that represent different 'DocumentWindow' objects and optional
control buttons
>
> Click on the right end (on the ‘X’) to select the tab strip

<HTML><ul></HTML>
<HTML><li></HTML>> <HTML><p></HTML>documentTabStrip1.TabStripVisible =
False; %%//%% Hide the DocumentTabStrip (to hide the doc window tabs and
global close button)<HTML></p></HTML>
<HTML></li></HTML><HTML></ul></HTML>

=== DocumentWindow – rddw (great grandchild control) ===

> Individual Document Windows, like source files in VS
>
> You can put a RadSplitContainer in a documentWindow, but not right in the
RadDock

  * DefaultFloatingSize –

<HTML><ul></HTML>
<HTML><li></HTML>> <HTML><p></HTML>radDock.DocumentButtons = none; %%//%%
Hide the buttons (like close) in the upper right of a
documentWindow<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>**DocumentButtons** options – All, None,
Close, ActiveWindowsList, SystemMenu – buttons on right of DocumentTabStrip.
NOTE: This must be set for each DocumentWindow. It seems like this would be
a DocumentTabStrip property that is global to all DocumentWindows, but
apparently not.<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>radDock.DocumentTabsVisible = false;
%%//%% Hide the documentWindow tab area on the RadDock. If you do, you don’t
need to use ‘DocumentButtons’ = None<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>this.radDock1.ShowDocumentCloseButton =
false; %%//%% Hide the documentWindow close button<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>documentWindow1.CloseAction %%//%%
Controls what happens when you close a window,<HTML></p></HTML>
<HTML></li></HTML>
```

```
<HTML><li></HTML><HTML><p></HTML>Close<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>CloseAndDispose – default is to to destroy
Window on close request<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>Hide – use this if you want to hide a
window and be able to restore
it<HTML></p></HTML><HTML></li></HTML><HTML></ul></HTML>

> **Hide/Show a DocumentWindow**
>
> documentWindow1.CloseAction =
Telerik.WinControls.UI.Docking.DockWindowCloseAction.Hide; %%//%%
CloseAction = Hide, else Window is destroyed on close request
>
> documentWindow1.DockState == DockState.Hidden %%//%% Hide documentWindow
>
> documentWindow1.DockState == DockState.TabbedDocument %%//%% Show
documentWindow
>
> documentWindow1.DockState = documentWindow1.DockState == DockState.Hidden
? DockState.TabbedDocument : DockState.Hidden; %%//%% Toggle Show/Hide
>
> DockWindow[] hiddenWindows =
this.radDock1.DockWindows.GetWindows(DockState.Hidden); %%//%% Get a
collection of all Hidden documentWindows
>
> Here’s another way:
>
> %%//%%Hide document window tab on main form
>
> documentWindow3.TabStripItem.Visibility = ElementVisibility.Collapsed;

ToolTabStrip – rdtwts (child container)

> Header strip above a ToolWindow with optional caption text and control
buttons
>
> //Click on the right end (on the ‘X’) to select the tool window tab
strip//

<HTML><ul></HTML>
<HTML><li></HTML>> <HTML><p></HTML>this.toolTabStrip1.Text = ‘This is
irrelevant!’; %%//%% The Text on the ToolStrip is controlled by the
ToolWindow.Text property<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>this.toolTabStrip1.CaptionVisible =
false; %%//%% Hide the caption of a single ToolTabStrip<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>this.radDock1.ShowToolCloseButton =
false; %%//%% Hide the ToolWindow close button<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>this.AllowedDockState =
```

```
AllowedDockState.AutoHide | AllowedDockState.Docked; %%//%% Force stay
docked and not allow the user to move it<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>this.DockState =
Telerik.WinControls.UI.Docking.DockState.AutoHide; %%//%% Collapse/AutoHide
the ToolWindow (not visible in the IDE)\\
> Options: AutoHide, Docked, Floating, Hidden,
TabbedDocument<HTML></p></HTML>
<HTML></li></HTML><HTML></ul></HTML>

=== ToolWindow – rdtw (grandchild control) ===

> Window dockable to any of the 4 sides by default, like VS Properties
window
>
> See this article on
[[http://www.telerik.com/help/winforms/dock-object-model-allowed-dock-states
.html|AllowedDockStates]] to control floating, etc. And this:
http://www.telerik.com/forums/909245-disable-the-floating-feature
>
> Click in the tool window body (below the ToolTabStrip)

<HTML><ul></HTML>
<HTML><li></HTML>> <HTML><p></HTML>ToolCaptionButtons - None, Close,
AutoHide (the stickpin), SystemMenu, All – buttons on right of
ToolWindow<HTML></p></HTML>
<HTML></li></HTML><HTML></ul></HTML>

> **this.ToolCaptionButtons = ToolStripCaptionButtons.AutoHide**; %%//%%
Shows only the stickpin. Options:

<HTML><ul></HTML>
<HTML><li></HTML>> <HTML><p></HTML>None – no buttons<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>Close – if you close you can’t get it
back<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>Autohide – only show the
stickpin<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>SystemMenu – drop down with multiple
options<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>All<HTML></p></HTML>
<HTML></li></HTML><HTML></ul></HTML>

<HTML><ul></HTML>
<HTML><li></HTML>> <HTML><p></HTML>this.radDock1.ToolTabsAlignment =
TabStripAlignment.Right; %%//%% Show the ToolWindow tab strip on the left
edge<HTML></p></HTML>
<HTML></li></HTML>
```

```
<HTML><li></HTML>> <HTML><p></HTML>this.radDock1.ToolTabsVisible = false;
%%//%% Hide the ToolWindow tab area on the RadDock<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>ContextMenuStrip –shortcut menu to
display on right click<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>**toolWindow.Caption** – Text header that
appears on ToolTabStrip above the ToolWindow.<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>**toolWindow.text** – Text in the expand
button visible when the ToolWindow is unpinned. //(.Caption is the text in
the ToolTabStrip above the ToolWindow.)//<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML><HTML></li></HTML><HTML></ul></HTML>

> %%//%% Only allow ToolWindows to resize, autohide and pin ### Fix this
code sample up – wrong syntax
>
> %%//%% From:
http://www.telerik.com/forums/how-to-disable-toolwindow-floating-after-doubl
e-click-header
>
> Me.ToolWindow1.AllowedDockState =
Telerik.WinControls.UI.Docking.AllowedDockState.All And Not
Telerik.WinControls.UI.Docking.AllowedDockState.Floating

==== RadPanorama – rp ====

http://www.telerik.com/help/winforms/panorama-overview.html

==== * RadPageView – rpv  ====

http://www.telerik.com/help/winforms/pageview-overview.html

http://www.telerik.com/community/forums/winforms/page-view.aspx\\
Video:
[[http://www.telerik.com/videos/winforms/getting-started-with-radpageview|ht
tp:%%//%%www.telerik.com/videos/winforms/getting-started-with-radPageView]]

RadPageViewPage.ViewMode %%//%% Types (ViewMode) of page displays. Options
are:

  *
[[http://www.telerik.com/help/winforms/pageview-stripview-getting-started.ht
ml|Strip]] – regular tabs across one of the 4 edges\\
{{tdn:image38.png?286x231}}
  *
[[http://www.telerik.com/help/winforms/pageview-stackview-overview.html|Stac
k]] – An accordion style stack with the selected page name (header) on top,
the buttons on 1 of 4 sides and the content displayed next to, on-top of the
items’ stack, or next to the selected item (where the buttons move as
```

```
selected).\\
{{tdn:image39.png?286x231}}
    *
RadPageViewPage.ViewElement.[[http://www.telerik.com/help/winforms/pageview-
stackview-customizing-the-stack-position.html|StackPosition]] – specify
which side buttons are on
    *
RadPageViewPage.ViewElement.[[http://www.telerik.com/help/winforms/pageview-
stackviewview-customizing-the-selection-mode.html|ItemSelectionMode]] -
content displayed on-top of the item button stack, or next to the selected
item (where the buttons move as selected)
      * Standard – content above button/item stack. Buttons don't move.
      * ContentWithSelected – content between buttons which move move as
selected (probably best)
      * ContentAfterSelected – content and buttons move here as well but
it's hard to tell which button
    * RadPageViewPage.ViewElement.ItemSizeMode – maybe look at this one too
  *
[[http://www.telerik.com/help/winforms/pageview-outlookview-overview.html|Ou
tlook]] – Buttons stay stacked on the bottom and the pages appear like the
left pane of Outlook\\
{{tdn:image40.png?291x235}}
  *
[[http://www.telerik.com/help/winforms/pageview-explorerbarview-overview.htm
l|ExplorerBar]] – display multiple visible pages at once in a scrolling
mechanis. Offers 3 types of content sizing (FixedLength, AutoSizeToBestFit,
EqualLength) & 2 stack positions (Top, Left). In this case the Headings move
up and down with the data. They even move vertically outside the visible
area possibly leaving the user unsure of context.\\
{{tdn:image41.png?290x234}}
  *
[[http://www.telerik.com/help/winforms/pageview-backstageview-getting-starte
d.html|BackStage]] – options on left with a cute graphic indicator and
selected page on right\\
{{tdn:image42.png?290x234}}

  * Add as a child to an existing container: 1st select the container then
double click the RadPageView control to add
  * Text - choose the parent 'RadPageView' or child 'RadPageViewPage#' from
the Property panel dropdown to change properties for each page button
  * The 'Pages' collection (in the Property Grid) defines the individual
pages at design time

Quick Tasks Button

  * 'Add Page' to add pages
  * 'Strip Buttons' = "None" to hide <X> (like titlebar)
  * 'Item content Orientation' list of tabs appear vertical, horizontal or
auto next to page
  * 'Strip Alignment' Tabs should appear on which edge?
  * 'Item Fit Mode' controls size of Page Tabs
```

```
  * 'View Mode' controls layout of pages providing tabs, or Outlook style
accordian

**Scrollable Page Contents in non-Strip View Mode (as in dbEntreé)**

  * A good way to create scrollable sections within a non-Strip (standard
tabs) View Mode is to create a user control and size it vertically to
contain all the required controls. This enables you to scroll up/down in the
user control in the IDE to edit the sub-controls.
  * Next Dock Fill a RadScrollablePanel in the RadPageView Page.
  * Finally Top, Left, Right Anchor the user control in the
RadScrollablePanel (don’t Dock it) and they scrollable panel will adjust to
the user control height.
  * //Remember to Left, Right anchor the controls in the user control is
they adjust with the width of the PageView.//

=== Page Visibility ===

NOTE: You cannot hide the PageView tabs by changing a design time property.
You must edit the code to set the 'ElementVisibility.Collapsed' property for
each RadPageView page as shown here:

**Hide all RadPageView Pages/tabs**

%%//%% Stick this right after “InitializeComponent();” on the form or UC.

%%//%% Hide all RadPageView 'Page' tabs

foreach (Telerik.WinControls.UI.RadPageViewPage item in radPageView.Pages) {

item.Item.Visibility = Telerik.WinControls.ElementVisibility.Collapsed;

}

**Hide any RadPageView Page/tab with 'Name' = "Hidden" or where 'Text'
contains "Hidden"**

%%//%% Hide any RadPageView Page with 'Name' = "Hidden" or where 'Text'
contains "Hidden" (just the tab, not content)

foreach (Telerik.WinControls.UI.RadPageViewPage item in radPageView.Pages )
{

if (item.Text.ToLower().Contains("hidden") == true ||
item.Name.ToLower().Contains("hidden") == true)

item.Item.Visibility = Telerik.WinControls.ElementVisibility.Collapsed;
%%//%% Must use lower case text

}
```

```
**Hide a specific RadPageView Page/tab by name**

foreach (Telerik.WinControls.UI.RadPageViewPage item in radPageView.Pages)
%%//%% Hide a RadPageView Page with a specific 'Name'

if (item.Name.Contains("rpvpContactSupport") == true)

item.Item.Visibility = Telerik.WinControls.ElementVisibility.Collapsed;

**Or multiple pages…**

foreach (RadPageViewPage item in rpvListViewSettings.Pages) %%//%% Hide a
RadPageView Page(s) with specific 'Name'(s)

if (item.Name.Contains("rpvpListSavedViews") ||
item.Name.Contains("rpvpListToolbar"))

item.Item.Visibility = Telerik.WinControls.ElementVisibility.Collapsed;

**Hide a specific RadPageView Page/tab with specific tab text**

foreach (Telerik.WinControls.UI.RadPageViewPage item in radPageView.Pages)
%%//%% Hide a specific RadPageView Page/tab with specified tab text

if (item.Text.Equals("Contact Support") == true)

item.Item.Visibility = Telerik.WinControls.ElementVisibility.Collapsed;

=== Page Selection ===

**Select the currently configured DefaultPage**

rpvListViewSettings.SelectedPage = rpvListViewSettings.DefaultPage; %%//%%
select the currently configured default page

**Select the 1st Page on the RadPageView by default**

%%//%% Include the following right after “InitializeComponent();” on the
form or UC.

radPageView.SelectedPage = radPageView.Pages[0]; %%//%% Select the 1st Page
on the RadPageView by default

%%//%% This isn't necessary if you use the radRibbonBar1_CommandTabSelected
method to link RadRibbonTabs to specified RadPages (by Tag)

%%//%% If you use that method the page will be selected automatically when
the 'Select the 1st tab on the RadRibbonBar by default' code executes

**Select the 1st Visible (not hidden) Page on the RadPageView by default**
```

```
bool PageSelected = false; %%//%% default to no page selected

foreach (Telerik.WinControls.UI.RadPageViewPage item in radPageView.Pages)

{

if (item.Item.Visibility == Telerik.WinControls.ElementVisibility.Visible)

{

radPageView.SelectedPage = item; %%//%% Select the 1st visible Page on the
RadPageView by default

PageSelected = true; %%//%% there is an active page so don’t hide the
RadPageView

break;

}

}

radPageView.Visible = PageSelected; %%//%% hide the RadPageView container if
no page is visisble

**Select the specified Page (by ‘text’ name) on the RadPageView by default**

%%//%% Include the following right after “InitializeComponent();” on the
form or UC.

%%//%% Select the specified Page (by ‘text’ name) on the RadPageView by
default

SetSpecifiedPageViewByText(rpvLogViewer,"All"); %%//%% Select the 1st Page
('.Text' = "All") on the RadPageView by default

%%//%% Then follow with this

%%//%% Select RadPageView Page by specific name ('.Text' value)

private void SetSpecifiedPageViewByText(RadPageView rpv, string SearchValue)

{

foreach (Telerik.WinControls.UI.RadPageViewPage item in rpv.Pages) %%//%%
Hide a specific RadPageView Page/tab with specified tab text

if (item.Text.Equals(SearchValue, StringComparison.CurrentCultureIgnoreCase)
== true)

rpv.SelectedPage = item;
```

```
}

### Add example of selecting rpvp Page by RadioButton selection

Create an entry based on this video:
[[file:///\\192.168.93.1\KB_MiM\Telerik\20160110)%20Radio%20Buttons%20control%20PageView%20Pages%20(tabs).mp4|\\192.168.93.1\KB_MiM\Telerik\20160110)
Radio Buttons control PageView Pages (tabs).mp4]]\\
This is from the TreeDataExtracter project

### Add example of selecting (syncing) rpvp Pages with the currently
selected radRibbonBar1 CommandTabs

RadPageViewPage – rpvp

RadPageView Child Pages

Add Pages

> Select RadPageView | Pages: | Add Page (Remove is here too)

Edit Page Properties

> Click the Page tab, then click the page body to see Properties

<HTML><ul></HTML>
<HTML><li></HTML>> <HTML><p></HTML>Name \Design<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>Text \Appearance<HTML></p></HTML>
<HTML></li></HTML><HTML></ul></HTML>

==== * RadSplitContainer – rsc  ====

http://www.telerik.com/help/winforms/splitcontainer-overview.html

http://www.telerik.com/community/forums/winforms/split-container.aspx\\
[[http://www.telerik.com/help/winforms/splitcontainer-building-a-layout-of-radsplitcontainers-programmatically.html|Building a layout of
RadSplitContainers programmatically]]

Container for resizable panels. Once the container exists you use "Add
Panel" from the Quick Tasks Button.

{{tdn:image43.png?539x230}}

  * Add as a child to an existing container: 1st select the container then
double click the RadSplitContainer control to add

Quick Tasks Button

  * 'Horizontal'/'Vertical' link toggles direction of split
```

```
(‘**Orientation’** property)
  * 'Add Panels' to add Panels (use 2 RadSplitContainer controls to make 3
panes with both a horiz and vert split
  * 'Remove Panel' is here too

The container SplitContainerLayoutStrategy instance handles all layout
requests from its owning container. Each SplitPanel instance has a member of
type SplitPanelSizeInfo, which is used by the strategy to determine the size
and position of this panel on its hosting container. The ‘**SizeMode’**
property (under ‘SizeInfo’) offers four different sizing modes per panel
basis:

  - **Auto**: a panel is auto-sized, based on the available remaining size
of the container.
  - **Relative**: a panel occupies amount of space, relative to the
container’s available size. This amount is specified through the SplitPanel
SizeInfo.RelativeRatio property.
  - **Absolute**: Absolute size is used for a panel. The amount of size is
specified through the SplitPanel SizeInfo.AbsoluteSize property.
  - **Fill**: Special mode currently used by the MainDocumentContainer. When
a RadSplitContainer has a descendant(s) with this special size mode, all
panels are sized absolutely, leaving the available space for all the “Fill”
panels.

Properties

  * ‘SplitContainerElement’ – Gets the split container element.
  * ‘SplitPanels’ – Gets the collection that contains the split panels.
  * ‘Orientation’ – Gets or sets the splits panels orientation.
  * ‘**SplitterWidth**’ – Gets or sets the splitters width. Does not resize
the ‘SpliterButtons’ if they are enabled.
  * ‘**EnableCollapsing**’ – This property enables the collapsing of the
panels by double clicking the splitter element.
  * ‘**UseSplitterButtons**’ – This property shows the buttons which can be
used for splitter collapsing or expanding.
  * Replace the SplitterButtons – Click the SplitContainer Tasks button,
choose ‘Edit UI Elements’ then
    * Collapse - Set an image in the 1<sup>st</sup> ‘RadButtonElement’
instance: SplitContainerElement | SplitterElement | RadButtonElement |
ImageAndTextLayputPanel | ImagePrimative
    * Restore - Set an image in the 2<sup>nd</sup> ‘RadButtonElement’
instance: SplitContainerElement | SplitterElement | RadButtonElement |
ImageAndTextLayputPanel | ImagePrimative
    * Set the SplitterWidth to fit
    * //NOTE: The original arrow graphic will still be shown behind the new
graphic so this may not be the best method. More on this here:
http://www.telerik.com/forums/custom-splitter-image//

SplitPanel – rscp

‘Collapsed’ – True = hide the current panel
```

```
=== Splitter Control Code Samples ===

%%//%% Lock the Splitter the user can not resize the panels (add to the Load
event of form)

this.radSplitContainer.Splitters[0].Fixed = true;

%%//%% Hide & Disable the Splitter (add to the Load event of form)

%%//%% NOTE: This hides the splitter buttons but does not hide or diable the
splitter itself.

%%//%% See: http:%%//%%www.telerik.com/forums/disable-splitter

this.radSplitContainer.Splitters[0].Visibility =
Telerik.WinControls.ElementVisibility.Collapsed;

Change Splitter Collapse direction

http://www.telerik.com/forums/change-splitpanel-collapse-direction

%%//%% Set the Splitter Width dynamically

radSplitContainer.SplitterWidth = 20;

SplitPanel AutoSize

http://www.telerik.com/forums/splitpanel-autosize

Capture the click event on any splitter in a RadSplitcontainer, or the click
event on a specific splitter in a RadSplitcontainer\\
http://www.telerik.com/forums/splitter-click-event

Save and Restore SplitPanel Layout\\
http://www.telerik.com/forums/radsplitcontainer

==== * RadGroupBox – rgb  ====

http://www.telerik.com/help/winforms/panels-and-labels-groupbox-overview.htm
l

http://www.telerik.com/community/forums/winforms/panels-and-labels.aspx

RadGroupBox control is a group box control with advanced styling options.
The primarily usage of this control is to hold a single radio buttons group.

{{tdn:image44.png?277x174}}

Change the border:
http://www.telerik.com/forums/how-to-change-the-border-color-of-radgroupbox-
and-contentpanel-of-tabitem
```

```
==== * RadPanel – rpnl ====

http://www.telerik.com/help/winforms/panels-and-labels-panel-overview.html

http://www.telerik.com/community/forums/winforms/panels-and-labels.aspx

{{tdn:image45.png?246x106}}

This screenshot demonstrates the use of a border, border width, background
with a linear gradient, and Html-like formatted text. Refer to [[#html-like-
text-formatting|HTML-like Text Formatting]] for further details.

  * Group collections of controls allowing you to place them anywhere you
wish
  * Container to force mutual exclusivity on RadRadioButton groups
  * Hide the border options:
    * Click SmartTag and set “Edit UI Elements | BorderPrimative |
ShouldPaint” to ‘False’, or use this code:

> this.rpnlPanel.PanelElement.PanelBorder.ShouldPaint = false;

  * Or to hide some/selected borders using this code:

> %%//%% Hide some/selected RadPanel borders\\
> this.rpnlPanel.PanelElement.PanelBorder.BoxStyle =
BorderBoxStyle.FourBorders; %%//%% enable border segments
>
> this.rpnlPanel.PanelElement.PanelBorder.TopWidth = 0; %%//%% hide the top
border
>
> this.rpnlPanel.PanelElement.PanelBorder.BottomWidth = 0;
>
> this.rpnlPanel.PanelElement.PanelBorder.LeftWidth = 0;
>
> this.rpnlPanel.PanelElement.PanelBorder.RightWidth = 0;\\
> %%//%% From:
http://www.telerik.com/forums/can-radpanel-only-have-bottom-border-or-top-et
c

==== RadCollapsiblePanel – rcpnl  ====

http://www.telerik.com/help/winforms/panels-and-labels-collapsible-panel-ove
rview.html

An expandable panel which can host controls in its content area. The content
area can be collapsed, allowing space savings in a WinForms application.
RadCollapsiblePanel also gives you control over its ExpandDirection for even
greater flexibility.

{{tdn:image46.gif?361x217}}
```

```
  * ExpandDirection - expand direction.
  * EnableAnimation - use animation to expand or collapse the control.
  * ContentSizingMode -
  * IsExpanded - Get/Set whether the control is expanded.
  * ShowHeaderLine Gets or sets a value indicating whether to show a line
primitive in the header.
  * HorizontalHeaderAlignment -
  * VerticalHeaderAlignment -
  * HeaderText –
  * AnimationInterval -
  * AnimationFrames -
  * AnimationEasingType -
  * AnimationType -

==== RadScrollablePanel – rspnl ====

http://www.telerik.com/help/winforms/panels-and-labels-radscrollablepanel-ov
erview.html

This control is basically just a RadPanel with 2 themable scrollbars to
provide more space for child controls.

The scrollbars only appear when a control is outside the viewable area.

The controls resize when the scroller appears so the scrollbar doesn’t
obscure part of the controls.

It automatically creates a parent radScrollablePanel.PanelContainer for the
ScrollablePanel.

To test the scroller just add some control then set its location outside the
viewable area. The scrollbar(s) will appear in the IDE preview.

{{tdn:image47.png?319x400}}

==== * RadSeparator – rs  ====

http://docs.telerik.com/devtools/winforms/panels-and-labels/separator

RadSeparator is a control that gives you the ability to divide your forms
into logical parts.

{{tdn:image48.png?297x123}}

  * Orientation - gets or sets the control orientation to Vertical or
Horizontal
  * ShadowOffset - gets or sets the offset of the both lines, both
horizontal and vertical
  * ShowShadow - enables/disables the second line
  * SeparatorElement - the element that holds the lines. Gives you the
ability to access and customize them
```

```
====== Telerik – Data Controls ======

==== RadChartView – rcv ====

http://www.telerik.com/help/winforms/chartview-overview.html\\
http:%%//%%www.telerik.com/support/code-library/winforms/chart\\
VIDEO:
http://www.telerik.com/videos/winforms/getting-started-with-radchartview-for
-winforms\\
VIDEO:
http://www.telerik.com/videos/winforms/what-is-new-in-q3-2012-radcontrols-fo
r-winforms (19:12 - 24:12)

RadChartView is a versatile charting component that offers drawing
capabilities, user interaction and real-time updates. Its intuitive object
model and public API allow complex charts to be easily setup either in
design-time or through code. The control is completely data aware and may
work in bound or unbound mode, depending on the requirements. Chart types
(or series) are organized in hierarchies, depending on the coordinate
system, used to plot data points – for example we have CartesianArea (using
Cartesian coordinate system) and PolarArea (using polar or radial coordinate
system) and PieArea.

{{tdn:image49.png?268x188}}

//NOTE: There is also a RadChart control, which is being phased out. Don’t
use it.
[[http://www.telerik.com/help/winforms/chartview-radchartview-vs.-radchart.h
tml|Here’s a comparison]].//

Data point Tooltips -
http://www.telerik.com/help/winforms/chartview-features-tooltip.html

Chart Title -
http://www.telerik.com/help/winforms/chartview-features-title.html

Trackball (threshold indicator line ) –
http://www.telerik.com/help/winforms/chartview-features-trackball.html

Smart labels -
http://www.telerik.com/help/winforms/chartview-features-smart-labels.html

Drill down -
http://www.telerik.com/help/winforms/chartview-features-drill-down.html

Changing colors/palettes -
http://www.telerik.com/help/winforms/chartview-customization-palettes.html

%%//%% Apply a predefined palette of colors\\
this.radChartView1.Area.View.Palette = KnownPalette.Metro;
```

```
%%//%% Apply a specific palette entry to a specific line. Default is to
rotate through the palette colors.

lineSeria.Palette = KnownPalette.Flower.GlobalEntries[0];

lineSeria.Palette = new PaletteEntry(Color.Yellow, Color.Red);

%%//%% Define your own palette by inheriting from ChartPalette and creating
a collection of palette entries\\
public class CustomPalette : ChartPalette

{

public CustomPalette()

{

this.GlobalEntries.Add(Color.Yellow, Color.Red);

this.GlobalEntries.Add(Color.Yellow, Color.Blue);

}

}

%%//%% Then set the custom palette:\\
this.radChartView1.Area.View.Palette = new CustomPalette();

Custom rendering -
http://www.telerik.com/help/winforms/chartview-customization-custom-renderin
g.html

Chart Animation - http://www.telerik.com/forums/animated-chart\\
“RadChartView contains a mechanism that allows an animation and the
animation is built-in the theme. You can create your own animations by
either using
our [[http://www.telerik.com/help/winforms/tpf-animations.html|TPF
Animations]] or you can
use [[http://blogs.telerik.com/winformsteam/posts/12-06-21/put-some-css-magi
c-to-your-radcontrols-for-winforms.aspx|CSS like definition.]] You can see
examples of these approaches in our Demo application that is installed with
the controls. You can see a TPF animation in the
example// RadChartView>>Chart Types>>Line & Area// and a CSS animation in
the example //RadChartView>>Chart Types>>Bar.”//

==== * RadGridView – rgv  ====

http://www.telerik.com/help/winforms/gridview-overview.html\\
http://www.telerik.com/community/forums/winforms/gridview.aspx\\
http:%%//%%www.telerik.com/support/code-library/winforms/gridview\\
Video (Webinar):
```

```
http://www.telerik.com/videos/winforms/radgridview-for-winforms-webinar\\
Video:
[[http://www.telerik.com/videos/winforms/radgridview-for-winforms-hierarchy-
overview|RadGridView for WinForms Hierarchy Overview]]\\
Video:
[[http://www.telerik.com/videos/winforms/binding-radgridview-for-winforms-to
-a-self-referencing-hierarchy|Binding RadGridView for WinForms to a Self
Referencing Hierarchy]]

===  ===

=== RadGridView Config by XML ===

See: '\\192.168.93.1\KB_MiM\Telerik\20170307) Demo of RadGridView Config by
XML.mp4'

For how to setup user control grid columns and save them via XML.

  - Create an extra form called: 'rfrmGridViewConfig-controlName' where
'controlName' is the name of the actual ucStatusGrid, ucItemList, etc. This
form will never be used by the actual application.
  - Put a plain RadGridView on this form and configure the columns as
required.
  - Now <Save> your settings\\
{{tdn:image51.png?506x393}}
  - Save the file in the Project folder and name it like this:
'rfrmGridViewConfig-controlName.xml' where 'controlName' is the name of the
actual ucStatusGrid, ucItemList, etc. NOTE: This naming format is different
that what I did in the video. There I put the control name 1<sup>st</sup>.
We can debate that…
  - Next you can configure the grid in the uc based on the XML settings at
runtime.
  - ### I think the ucItemList has a custom property to select this XML
file. We need to setup the same for our various other ucs (GridStatus, etc)

=== Quick Tasks Button ===

  *
'[[http://www.telerik.com/help/winforms/gridview-columns-gridviewcolumn.html
|Columns]]' collection
    * rgvtxt –
[[http://www.telerik.com/help/winforms/gridview-columns-gridviewtextboxcolum
n.html|Textbox column]]
    * rgvbrow –
[[http://www.telerik.com/help/winforms/gridview-columns-gridviewbrowsecolumn
.html|Browse column]]
    * rgvcalc –
[[http://www.telerik.com/help/winforms/gridview-columns-gridviewcalculatorco
lumn.html|Calculator column]]
    * rgvchk –
[[http://www.telerik.com/help/winforms/gridview-columns-gridviewcheckboxcolu
```

```
mn.html|Checkbox column]]
    * rgvcolor –
[[http://www.telerik.com/help/winforms/gridview-columns-gridviewcolorcolumn.
html|Color column]]
    * rgvcbo –
[[http://www.telerik.com/help/winforms/gridview-columns-gridviewcomboboxcolu
mn.html|Combobox column]]
    * rgvcmd –
[[http://www.telerik.com/help/winforms/gridview-columns-gridviewcommandcolum
n.html|Command column]]
    * rgvdatetime –
[[http://www.telerik.com/help/winforms/gridview-columns-gridviewdatetimecolu
mn.html|Date Time column]]
    * rgvdec –
[[http://www.telerik.com/help/winforms/gridview-columns-gridviewdecimalcolum
n.html|Decimal column]]
    * rgvurl –
[[http://www.telerik.com/help/winforms/gridview-columns-gridviewhyperlinkcol
umn.html|Hyperlink column]]
    * rgvimg –
[[http://www.telerik.com/help/winforms/gridview-columns-gridviewimagecolumn.
html|Image column]]
    * rgvmask –
[[http://www.telerik.com/help/winforms/gridview-columns-gridviewmaskboxcolum
n.html|MaskBox column]]
    * rgvrmcbo –
[[http://www.telerik.com/help/winforms/gridview-columns-gridviewmulticombobo
xcolumn.html|MultiCombobox column]]
    * rgvrate –
[[http://www.telerik.com/help/winforms/gridview-columns-gridviewratingcolumn
.html|Rating column]]
  * ‘Enable Adding’
  * ‘Enable Editing’
  * ‘Enable Column Reordering’
  * ‘Open Property Builder’ – Lots of config here…
    * MasterTemplate | Columns
      * <New> add various types of columns to the grid. Then select each
column and set properties in the bottom pane.
      * ‘RadGridView | Advanced | Appearance | ShowRowHeaderColumn’ –
True/False show/hide the small block column header for the row to the left
of all other columns.
  * ‘Best Fit Columns’ I think this is only impacts at deisgn time and does
not sets a runtime property. I think you have to run code to do it at
runtime.
  * //‘AutoSize’ – this one actually is what you want, rather than ‘Best Fit
Columns’//
  * ‘Dock in parent container’
  *

**Properties**
```

```
  * ‘ReadOnly’ = True - User can’t edit records
  * ‘SelectionMode’ – ‘CellSelect’ or ‘FullRowSelect’
  * ‘MultiSelect’ – True – user can select more than one row at a time
  * ‘ShowRowHeaderColumn’ = ‘Falsse’ – controls the small block in the
header to the extreme left of all other columns.
  * ‘AutoSizeColumnMode’ = ‘Fill’ – expands columns to fill the space

> radGridView1.AutoSizeColumnsMode = GridViewAutoSizeColumnsMode.Fill;

  * ‘AutoSizeRows’ – expand rows (if a column has ‘Multiline’ = ‘True’??)
  * ‘EnableHotTracking’ – set to False or the grey beveled gradient will be
lost (leaving only the background color) as you mouse over grid data
  * ‘EnableAlternatingRowColor’ – set True to get a greenline report style
  * ‘HideSelection’ – set True for selected item to remain highlighted when
the control loses focus
  *

Text in a Hyperlink column has this font color: #113E89 (RGB 17, 62, 137)

=== Grid Column Resizing (Marcel) ===

rgvLog.MasterTemplate.Columns["Message"].AutoSizeMode =
BestFitColumnMode.DisplayedDataCells; %%//%% resizing is based on the values
*visible on the screen* and works the same even with manual resizing
(double-click on the column edge)

rgvLog.MasterTemplate.Columns["Message"].AutoSizeMode =
BestFitColumnMode.AllCells; %%//%% resizing is based on the values in all
rows, even those that are not currently visible

=== Change the Backcolor of a Specific Column (Telerik Support) ===

Use the ViewCellFormatting event (the color depends on your current form
backcolor)

void radGridView1_ViewCellFormatting(object sender, CellFormattingEventArgs
e)

{

e.CellElement.DrawFill = true;

if (e.Column.Name == "SeparatorColumn")

{

e.CellElement.DrawFill = true;

e.CellElement.BackColor = Color.FromArgb(191, 219,
255);%%//%%Color.FromKnownColor(KnownColor.Control);
```

```
e.CellElement.DrawBorder = false;

e.CellElement.GradientStyle = GradientStyles.Solid;

}

else

{

e.CellElement.ResetValue(LightVisualElement.DrawFillProperty,
ValueResetFlags.Local);

e.CellElement.ResetValue(LightVisualElement.DrawBorderProperty,
ValueResetFlags.Local);

e.CellElement.ResetValue(LightVisualElement.GradientStyleProperty,
ValueResetFlags.Local);

e.CellElement.ResetValue(LightVisualElement.BackColorProperty,
ValueResetFlags.Local);

}

}

=== GridBrowseEditor allow text editing and paste ===

Subscribe to the CellEditorInitialized event and set the
GridBrowseEditorElement.ReadOnly property to false. Thus, you will allow the
user to enter custom text and perform paste operation as well: subscribe to
the CellEditorInitialized event and set the GridBrowseEditorElement.ReadOnly
property to false. Thus, you will allow the user to enter custom text and
perform paste operation as well:

**private void **radGridView1_CellEditorInitialized(**object **sender,
GridViewCellEventArgs e)

{

    GridBrowseEditor browseEditor = e.ActiveEditor **as **GridBrowseEditor;

    **if **(browseEditor!=**null**)

    {

        GridBrowseEditorElement element =
browseEditor.EditorElement **as **GridBrowseEditorElement;

        element.ReadOnly = **false**;
```

```
    }

}

=== Jump to Last Row ===

Scroll to the bottom by passing the rows count to ScrollToRow as an integer
parameter:

**private void** radButton1_Click(**object** sender, EventArgs e)

{

    radGridView1.TableElement.ScrollToRow(radGridView1.Rows.Count-1);

}

=== # Group Summaries ===

From the Demo app:

{{tdn:image52.png?720x399}}

==== # RadDataFilter ====

http://docs.telerik.com/devtools/winforms/datafilter/overview

Building complex filter expressions is a breeze with RadDataFilter. The
control allows specifying expressions based on the fields.properties
available in the data source. The intuitive UI of the control is designed to
facilitate the end-user while empowering the developer to create related
expressions with a few clicks.

{{tdn:image53.png?503x200}}

==== RadCardView – rcv  ====

http://docs.telerik.com/devtools/winforms/cardview/overview\\
http://www.telerik.com/forums/winforms/listview

RadCardView is a powerful control providing means for displaying and editing
text data organized in a card layout. It incorporates a RadLayoutControl
allowing modifications of the layout in the Visual Studio designer as well
as at run-time.

{{tdn:image54.png?455x237}}

==== # RadDataEntry ====

http://docs.telerik.com/devtools/winforms/dataentry/dataentry
```

```
RadDataEntry provides an easy way to display and edit arbitrary business
objects in a form layout. The built-in editors are generated by default, so
that a fully operational **CRUD support may be achieved with a single line
of code** - just binding to the business object or to a collection of
objects. In order to further enhance RadDataEntry, it can be used in
combination with RadBindingNavigator or any other collection navigation
control.

{{tdn:image55.png?278x224}}

=== Hide the border ===

(DataEntry).PanelElement.Border.Visibility = ElementVisibility.Collapsed;

=== DataValidation ===

Includes the ability to validate data per field.

Example from the Demo app:

{{tdn:image56.png?720x391}}

==== # RadDataLayout ====

https://docs.telerik.com/devtools/winforms/controls/datalayout/overview

The RadDataLayout control provides means for displaying data in a highly
customizable layout by automatically creating the items and editors and
further allowing easy creating of complex layouts at design-time as well as
at run-time.

{{tdn:image57.png?400x264}}

==== RadVirtualGrid – rvg  ====

http://docs.telerik.com/devtools/winforms/virtualgrid/overview

RadVirtualGrid is a grid component developed on top of Telerik Presentation
Framework which provides a convenient way to implement your own data
management operations and optimizes the performance when interacting with
large amounts of data.

{{tdn:image58.gif?488x388}}

**Key features**

  * Easily customizable appearance and theming mechanism
  * Outstanding performance
  * Editing mechanism
  * Hierarchical data presentation
  * Sorting
```

```
  * Filtering
  * Paging
  * Selection and navigation
  * Flexible API

==== RadRangeSelector – rrs  ====

http://docs.telerik.com/devtools/winforms/rangeselector/overview

RadRangeSelector provides an elegant solution for end-users to select range
(in percentages) and these percentages could be mapped to any kind of
visually represented data. Developers can easily set the associated object
that will be used as background of RadRangeSelector. The associated object
should confront with specific interfaces thanks to which it will be able to
communicate with RadRangeSelector. Currently, RadRangeSelector works out of
the box together with RadChartView.

{{tdn:image59.png?406x88}}

==== RadPivotGrid – rpg ====

http://www.telerik.com/help/winforms/pivotgrid-overview.html

http://www.telerik.com/community/forums/winforms/pivotgrid.aspx\\
VIDEO:
http://www.telerik.com/videos/devcraft/getting-started-with-radpivotgrid-for
-winforms\\
VIDEO:
[[http://www.telerik.com/videos/winforms/visualizing-kpis-with-radpivotgrid-
for-winforms|Visualizing KPIs (Key Performance Indicator) With RadPivotGrid
for WinForms]]\\
VIDEO:
[[http://www.telerik.com/videos/winforms/what-is-new-in-q3-2012-radcontrols-
for-winforms|What is new in R3 2012 Telerik UI for for WinForms]] ()

RadPivotGrid for WinForms is a control which provides functionality similar
to the functionality of PivotTables in MS Excel. It takes large chunks of
data and summarizes it in a human readable way by the help of aggregates and
field descriptors. The end-user can easily get an aggregated view of the
data that would best suit their needs by dragging and dropping the items of
the field descriptors and the aggregates. RadPivotGrid can also sort the
data and show subtotals and grand totals at the end or at the beginning of
the summarized data. It supports the UI virtualization available in
RadGridView, so it can easily handle large data sets bringing to you top
performance and low memory footprint even in such scenarios.

{{tdn:image60.png?556x311}}

Here is a list of the supported features:

  * Grouping data by row and column descriptors
```

```
  * Aggregating data by aggregate descriptors
  * Sorting group values
  * Report filters
  * Drag and drop of descriptors and aggregates
  * Field chooser dialog
  * Row/Column SubTotals and GrandTotals positions
  * Row/Column Headers layouts – Tabular and Compact
  * EmptyValueString, ErrorValueString, FormatString
  * Design Time support for data binding
  * UI virtualization
  * Column/row resizing
  * Tooltips
  * Context menus
  * Best Fit
  * Copy data to clipboard
  * Cell selection
  * Formatting cells

==== RadPivotFieldList – rpfl  ====

https://docs.telerik.com/devtools/winforms/controls/pivotgrid/overview.html
- Overview & Example Apps
http://www.telerik.com/help/winforms/pivotgrid-radpivotfieldlist.html

http://www.telerik.com/community/forums/winforms/pivotgrid.aspx

Similar to the functionality of PivotTables in MS Excel.

{{tdn:image60.png?448x250|pivotgrid-overview 001}}

^RadPivotFieldList
^Value Options Dialog
^Label Options Dialog
^
|{{tdn:image61.png?138x299|pivotgrid-radpivotfieldlist
001}}|{{tdn:image62.png?168x224|pivotgrid-radpivotfieldlist
002}}|{{tdn:image63.png?147x248|pivotgrid-radpivotfieldlist 003}}|

==== * RadDropDownList – rddl ====

http://www.telerik.com/help/winforms/dropdown-and-listcontrol-dropdownlist-o
verview.html

http://www.telerik.com/community/forums/winforms/dropdownlist-and-listcontro
l.aspx\\
Video:
http://www.telerik.com/videos/winforms/getting-started-with-raddropdownlist

**This is like a basic combobox**

{{tdn:image64.png?165x151}}
```

```
**NOTE: If you need a dropdown with checkboxes use the CheckedDropDownList**

Quick Tasks Button

  * 'Edit Items' - to add/manage items
  * DropDownListElement items:
    * 'SelectedItems' -1 = nothing unselected, 0 = 1st list element, etc.
NOTE: This may revert to -1 if you edit the list elements.'DropDownStyle' -
'DropDownList' list items are read only. 'DropDown' you are allowed to edit
list items.
    * DropDownSizingMode = RightBottom – add a size handle to the bottom
right of the list for manual resize
    * DropDownWidth – force the width, in pixels. Does not automatically
resize, but may be combined with DropDownSizingMode
    * AutoSizeItems - ?? ### test this one
    * DropDownMinSize – you can set the width (leaving Height=0) to a value
to make the item list wider than the main control.
  * SortStyle – Ascending, Descending and None
  * RadDropDownStyle – 'DropDown' allows editing of the textbox;
'DropDownList' read only drop down textbox
  * AutoScroll – T/F – Scrollbars automatically appear when the list is
longer than the control is tall.
  * AutoCompleteMode – Suggest, Append, **SuggestAppend** (most often the
best option)
  * DefaultItemsCountInDropdown – default = 6
  * How do I enable MultiItem select?

==== * CheckedDropDownList – rcddl ====

https://docs.telerik.com/devtools/winforms/controls/dropdown-listcontrol-and
-checkeddropdownlist/checkeddropdownlist/checkeddropdownlist

{{tdn:image65.png?218x103|dropdown-and-listcontrol-checkedropdownlist-
overview 001}}

==== * RadListControl – rlc  ====

http://www.telerik.com/help/winforms/dropdown-and-listcontrol-listcontrol-ov
erview.html

http://www.telerik.com/community/forums/winforms/dropdownlist-and-listcontro
l.aspx

RadListControl is the an alternative to the Microsoft ListBox control. This
is a basic list of selectable items.

**This is the simple list you are probably looking for.** The RadListView
offers more advanced features

{{tdn:image66.png?154x154}}
```

```
**NOTE: If you need a listbox with checkboxes use the [[#radcheckedlistbox-
rclb|RadCheckedListBox]]**

(including columns).

//NOTE: This control is shown under ‘Dropdown & List’ in the Telerik ‘Demo
Application – UI for Winforms’ and the online help.//

Supports drag and drop for reordering, or dragging between 2 different
RadListControls. See the Telerik sample application.

  * SelectionMode – offers 3 options:
    * One – only one row at a time
    * MultiSimple – click to select/deselect multiple rows
    * MultiExtended - <Shift>/<Ctrl> click to select multiple/ranges of
values
  * SortStyle – Ascending, Descending and None
  * AutoScroll – T/F – Scrollbars automatically appear when the list is
longer than the control is tall.
  * Items (Collection)
  *

==== RadListView – rlv  ====

http://www.telerik.com/help/winforms/listview-overview.html

http://www.telerik.com/community/forums/winforms/listview.aspx

http://www.telerik.com/help/winforms/listview-working-with-design-time-addin
g-groups.html\\
http://www.telerik.com/help/winforms/listview-custom-items.html

Flat list of labeled items (can be heirachical, like show car makes with
model as children).

//NOTE: If you are looking for a basic list of selectable items see the
[[#checkeddropdownlist-rcddl|RadListControl]].//

{{tdn:image67.png?416x242}}

Supports multiple columns.

  * **HorizontalScrollState** – State of horiz scrollbar
    * AutoHide – hide unless some data is out of bounds
    * AlwaysShow –
    * AlwaysHide – Use to disable the scrollbar if you don’t want to see it
  *

==== RadCheckedListBox – rclb ====

http://docs.telerik.com/devtools/winforms/checkedlistbox/checkedlistbox
```

RadCheckedListBox is an enhanced alternative to the standard Windows Forms checked list box control. RadCheckedListBox uses RadListView as a foundation. All previous functionality is preserved (visual formatting and data binding) and is now extended.

{{tdn:image68.png?181x89}}

{{tdn:image69.png?352x284}}

==== RadMultiColumnComboBox – rmcbo ====

http://www.telerik.com/help/winforms/multicolumncombobox-overview.html\\
http://www.telerik.com/forums/winforms/multicolumncombo

The multi-column combobox is a special case of combobox (RadDropDownList) control with RadGridView integrated in its drop-down. The control combines the functionality and features of RadComboBox and RadGridView showing multiple columns in the dropdownlist.

{{tdn:image70.png?361x198}}

  * The RadGridView control hosted in the drop-down does not allow editing operations
  * The RadGridView control hosted in the drop-down does not support filtering, grouping and sorting operations.

  * AutoCompleteMode – Suggest, Append, SuggestAppend (most often the best option) based on the DisplayMember

=== Filtering ===

http://www.telerik.com/help/winforms/multicolumncombobox-filtering.html

Per my 04/28/2015 question about flitering on alternate columns, TS responded: “//If you need to display a list of suggestions, it is appropriate to use the [[http://www.telerik.com/help/winforms/multicolumncombobox-filtering.html|filtering]] functionality. Thus, you can specify another property for the filtering action.//”

{{tdn:image71.png?428x147}}

```
private void Form1_Load(object sender, EventArgs e)

{

this.customersTableAdapter.Fill(this.nwindDataSet.Customers);

this.radMultiColumnComboBox1.AutoFilter = true;

this.radMultiColumnComboBox1.DisplayMember = "City";
```

```
FilterDescriptor filter = new FilterDescriptor();

filter.PropertyName = this.radMultiColumnComboBox1.DisplayMember;

filter.Operator = FilterOperator.StartsWith;

this.radMultiColumnComboBox1.EditorControl.MasterTemplate.FilterDescriptors.
Add(filter);

this.radMultiColumnComboBox1.DataSource = this.customersBindingSource;

this.radMultiColumnComboBox1.DisplayMember = "ContactName";

this.radMultiColumnComboBox1.ValueMember = "CustomerID";

}

==== * RadTreeView – rtv  ====

http://www.telerik.com/help/winforms/treeview-overview.html

http://www.telerik.com/community/forums/winforms/treeview.aspx\\
http://www.telerik.com/support/code-library/winforms/treeview

Telerik RadTreeView is the supercharged tree view component for Windows
Forms. It facilitates display, management, and navigation of hierarchical
data structures. The product offers many advanced features like drag-and-
drop, load on demand, context menus and data binding.

{{tdn:image72.png?180x181}}

{{tdn:image73.gif?330x289}}

**Select Multiple Items**

{{tdn:image74.png?196x225}}

**Tri-State Capable Checkboxes**

{{tdn:image75.png?195x172}}

**Mixed Checkboxes& Radio Buttons**

{{tdn:image76.png?206x232}}

Use the
[[http://www.telerik.com/help/winforms/treeview-design-time-property-builder
.html|Property Builder]] to configure the TreeView. It is very easy to set
properties, plus you can create elements (nodes) there as well.

**RadTreeView Properties**
```

```
  * ‘ShowLines’ - true
  * ‘ShowRootLines’ - allows expand/collapse of the entire tree from the
very top/root.
    * If False you cannot collapse any root elements that contain children
(so children are always visible).
    * If this is True and ‘ShowLines’ is False you will be able to
expand/collapse root parent nodes but there will be no lines (only [+]/[-]).
    * If False root nodes are not indented from left. So if the tree only
contains root nodes you can set ‘ShowRootLines’ = false and root node indent
will disappear.
  * ‘LineStyle’ = Solid, Dash, **Dot**, DashDot, DashDotDot
  * TreeIndent – **20** how far to indent nodes. Always applies to child
nodes. Not to root nodes if ‘ShowRootLines’ is False.
  * ‘ShowExpandCollapse’ - true
  * ‘FullRowSelect’ – highlight spans the entire width of the tree
  * ‘MultiSelect’ – false
  * ‘CheckBoxes’ – add checkboxes to every node
  *
‘[[http://www.telerik.com/help/winforms/treeview-working-with-nodes-using-ch
eckboxes.html|TriStateMode]]’ – for checkboxes where a parent node indicates
not all children are checked
  * ExpandAnimation – None, Opacity
  * KeyboardSearchEnabled – **True** = can user navigate to an item by
typing
  * ShowNodeToolTips - ### can’t make this work
  * ToggleMode – None, Doubleclick, SingleClick
  * AllowDragDrop **– False**
  * AllowAdd **– False**
  * AllowEdit **– False**
  * AllowRemove **– False**
  *

**RadTreeView Node Properties**

  * ‘Checked’
  * ‘CheckState’
  * ‘CheckType’ – None, CheckBox, RadioButton
  * ‘ToolTipText’ - ### can’t make this work – maybe the issue is
‘ShowNodeToolTips’??
  * ‘ShowNodeToolTips’ – **True** if you want tooltips

**Context Menu**

http://docs.telerik.com/devtools/winforms/treeview/context-menus/default-con
text-menu

[[http://docs.telerik.com/devtools/winforms/treeview/context-menus/modifying
-the-default-context-menu|Modifying the Default Context Menu]]

[[http://docs.telerik.com/devtools/winforms/treeview/context-menus/custom-co
ntext-menus|Custom Context Menus]]
```

```
RadTreeView displays a default context menu which appears when you right-
click on a node. This menu contains 7 items and they are:

  * New: A new sibling node is created.
  * Edit: An editor appears at the position of the node from which the
context menu is invoked. This editor allows you to change a property of the
node.
  * Delete: The node from which the menu is invoked is deleted.
  * Expand/Collapse: Depending on the expanded state of the node, this menu
item may display itself as "Expand" or "Collapse". When click the
corresponding operations occurs.
  * Cut: The node from which the context menu is invoked together with its
child nodes (in short - the whole branch) is cut.
  * Copy: The node from which the context menu is invoked together with its
child nodes (in short - the whole branch) is copied.
  * Paste: If there a cut or copied node (branch), this node is added as a
sibling node to the node from which the context menu is invoked.

The default context menu is disabled by default. Enable it in the IDE or via
code:

radTreeView1.AllowDefaultContextMenu = true; %%//%% Enable the default
context menu

These 3 menu items are diabled by default. Enable them individually in the
IDE or via code:

radTreeView1.AllowAdd = true; %%//%% Enable the ‘New’ context menu item

radTreeView1.AllowEdit = true; %%//%% Enable the ‘Edit’ context menu item

radTreeView1.AllowRemove = true; %%//%% Enable the ‘Delete’ context menu
item

==== RadBindingNavigator – rbn ====

http://docs.telerik.com/devtools/winforms/bindingnavigator/bindingnavigator\
\
http://www.telerik.com/forums/winforms/bindingnavigator

RadBindingNavigator’s main purpose is to provide a basic UI for navigation
through a collection of business objects. RadBindingNavigator can be paired
with a
[[https://msdn.microsoft.com/en-us/library/system.windows.forms.bindingsourc
e(v=vs.110).aspx|BindingSource component]].

{{tdn:image77.png?302x30}}

==== RadMap – rmap  ====

http://docs.telerik.com/devtools/winforms/map/overview\\
```

```
http://www.telerik.com/forums/winforms/map

RadMap can visualize tile data from the Bing Maps and the OpenStreetMaps
REST services as well as from the local file system.

{{tdn:image78.png?385x310}}

==== RadCarousel – rcar ====

http://docs.telerik.com/devtools/winforms/carousel/carousel\\
http://www.telerik.com/forums/winforms/carousel\\
VIDEO:
http://tv.telerik.com/watch/winforms/radcarousel/overview-radcarousel-winfor
ms

RadCarousel is a navigation control that animates a series of elements
either by the user clicking a particular element or by clicking the forward
and back arrows. Based on top of the Telerik Presentation Framework (TPF),
RadCarousel supports data binding, smooth animations and transitions,
automatic generation of image reflections and dynamic addition and removal
of items.

{{tdn:image79.png?461x220}}

====== Telerik – Gauges ======

==== RadBulletGraph - rgaugebullet ====

http://docs.telerik.com/devtools/winforms/gauges/bulletgraph/bulletgraph

The RadBulletGraph control is a variation of linear gauge. It combines a
number of indicators into one control making it light-weight, customizable,
and straightforward to setup and use. The control is a great tool for
dashboards as it is the optimal way to present a lot of information in
relatively small size.

{{tdn:image80.png?313x58}}

==== RadLinearGauge ====

http://docs.telerik.com/devtools/winforms/gauges/lineargauge/lineargauge

RadLinearGauge displays simple value within a specific range. The range is
displayed in a rectangle, this rectangle can contain scale with or without
ticks, labels and a scale bar. This control can be very useful when you need
to build business dashboards or you just need graphical indicators.

{{tdn:image81.gif?415x162}}

==== RadRadialGauge ====
```

http://docs.telerik.com/devtools/winforms/gauges/radialgauge/radialgauge

The RadRadialGauge control is designed to display a simple value within a definite range. This range is represented in a circular format similar to car speed gauge. The circular container contains a scale in it which controls the overall layout of ticks, tick labels, needles and ranges and renders an optional scale bar.

{{tdn:image82.gif?441x121}}

====== Telerik – Editors ======

==== * RadAutoCompleteBox – racb ====

http://docs.telerik.com/devtools/winforms/editors/autocompletebox/autocompletebox\\
http://www.telerik.com/community/forums/winforms/editors.aspx\\
Video:
http://www.telerik.com/videos/winforms/getting-started-with-radautocompletebox-for-winforms

RadAutoCompleteBox allows the end-user to easily fill-in text thanks to auto-complete functionality and tokens of text. This behavior is similar to the "To" field of Outlook and Facebook where you are filling-in the recipients to which you are going to send a message.

{{tdn:image83.png?335x158}}

==== * RadBrowseEditor – rbe ====

https://docs.telerik.com/devtools/winforms/controls/editors/browseeditor/overview\\
http://www.telerik.com/community/forums/winforms/editors.aspx

Browse to specify a file or folder on disk.

Take a look at this control set instead. [[https://docs.telerik.com/devtools/winforms/controls/file-dialogs/overview|File Dialogs]], It looks like this maybe replaces the RadBrowseEditor?

//It opens a FileOpen, File Save or Folder Selection dialog apparently without calling the Windows API.//

RadBrowseEditor is a themable control which allows users to select a file or a directory from the file system or directly type the full path to it in the editor.

{{tdn:image84.png?320x29}}

  * ‘DialogType’ – one of a [[http://docs.telerik.com/devtools/winforms/editors/browseeditor/dialog-type

```
s|predefined set of dialog types]] determined by the BrowseEditorDialogType
enumeration.
    * FolderBrowseDialog\\
radBrowseEditor1.DialogType = BrowseEditorDialogType.FolderBrowseDialog;
    * FontDialog\\
radBrowseEditor1.DialogType = BrowseEditorDialogType.FontDialog;
    * OpenFileDialog\\
radBrowseEditor1.DialogType = BrowseEditorDialogType.OpenFileDialog;
    * SaveFileDialog\\
radBrowseEditor1.DialogType = BrowseEditorDialogType.SaveFileDialog;

### Add the
[[https://docs.telerik.com/devtools/winforms/controls/file-dialogs/dialog-co
ntrols/radopenfolderdialog|RadOpenFolderDialog]] to this list. I like it
much better than the old FolderBrowseDialog. Also see it here, under the
RadBrowseEditor. I’m not sure which of those 2 controls are the one to use
now.

{{tdn:image85.png?278x365}}

{{tdn:image86.png?522x299}}

{{tdn:image87.png?524x300}}

==== RadCalculatorDropDown – rcdd  ====

http://docs.telerik.com/devtools/winforms/editors/calculatordropdown/calcula
tordropdown

RadCalculatorDropDown has a simple easy-to-use interface enabling the end
user to perform all basic calculations such as addition, subtraction,
multiplication, division as well as some more complicated ones – reciprocal,
square root, negate.

{{tdn:image88.png?205x229}}

==== RadColorBox – rcb  ====

http://www.telerik.com/help/winforms/editors-color-box-overview.html\\
http://www.telerik.com/community/forums/winforms/editors.aspx

The main purpose of the control is to allow the user to select a color from
a color dialog with preset colors or to type the color directly into the
text field. The control then displays the color name if it is a named color
or the RGB values of the selected color. A small rectangle filled with the
selected color is displayed as well.

{{tdn:image89.png?177x37}}

==== * RadDateTimePicker – rdtp  ====
```

```
http://www.telerik.com/help/winforms/editors-datetimepicker-overview.html

http://www.telerik.com/community/forums/winforms/calendar-and-datetimepicker
.aspx\\
VIDEO:
[[http://tv.telerik.com/watch/winforms/getting-started-with-radtimepicker-fo
r-winforms|Getting Started Video]]

RadDateTimePicker allows interactive selection of dates using a drop down
calendar.

{{tdn:image90.png?194x199}}

Choose a date (optionally a time of day?)

  * 'Format' = Short (9/24/2012), Long (Monday September 24, 2012), Time
(1:24:38 AM), Custom (9/24/2012 1:24 AM)
  * 'Value' -
 date selected from the picker set in code or via the drop down calendar in
the Properties window.

> this.rdtpStartDate.Value = DateTime.Today;  %%//%% Set Start date to today
>
>
this.radDateTimePicker1.Value = DateTime.Today.AddDays(1);  %%//%% Set the d
ate to tomorrow
>
>
this.radDateTimePicker1.Value = DateTime.Today.AddDays(-1);  %%//%% Set the
date to yesterday

  * By default a single arrow click moves one month and a double arrow click
moves three months (called Fast Navigation).\\
rrdtpNameOfControl.DateTimePickerElement.Calendar.FastNavigationStep = 12;
%%//%% override fast navigation to 12 months so double arrow click jumps 1
year
  * 'MinDate'/'MaxDate' - first and last dates that can be selected in the
picker. Attempts to select outside these bounds are ignored.

> this.radDateTimePicker1.MinDate = DateTime.Today.AddDays(-
DateTime.Today.Day);  %%//%% set the
>
> MinDate to be the first day of the current month.

  * 'NullText’ - text displayed when the NullableValue property is set to
null and RadDateTimePicker is not in focus. By default, NullText is an empty
string.

> this.radDateTimePicker1.NullText = "No date selected";

  * 'Format' - The DateTimePickerFormat enumeration values are Long, Short,
```

```
Time and Custom. Set Format to Custom to enable the CustomFormat property.
  * 'CustomFormat' - format string determines the display of the date in the
picker edit. See the Internationalization and Date Formats topic for more
information.
  * 'Culture determines the language that the drop down calendar and edit
will display in. See the Internationalization and Date Formats topic for
more information.
  * 'CalendarSize' - gets/sets the size of the RadCalendar in the
RadDateTimePicker drop-down.
  * 'NullableValue' - same as the Value property, but the NullableValue
property is of type Nullable DateTime. It can be null – in this case if
RadDateTimePicker is not selected it will show its NullText. In case
RadDateTimePicker is selected, it will show the last entered date – this
allows the end-user to enter and edit the date.

> this.radDateTimePicker1.NullableValue = null;

  *

==== * RadMaskedEditBox – rmeb  ====

http://www.telerik.com/help/winforms/editors-maskededitbox-overview.html\\
http://www.telerik.com/community/forums/winforms/editors.aspx

RadMaskedEditBox is a themeable text box that formats and constrains text to
a predefined pattern or a pattern you define. RadMaskedEditBox also handles
globalization for date and time edits. Date and Time masks allow the user to
navigate using the up and down arrow keys.

{{tdn:image91.png?416x262}}

The MaskType property defines what type of mask would be used in the masked
box:

<HTML><ul></HTML>
<HTML><li></HTML>> <HTML><p></HTML>**None**<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>>
<HTML><p></HTML>**[[http://www.telerik.com/help/winforms/editors-maskededitb
ox-date-and-time-masks.html|__DateTime__]] -** set the **MaskType** property
to **DateTime** and the **Mask** property to "d". //See
[[http://www.telerik.com/help/winforms/editors-maskededitbox-date-and-time-m
asks.html|__Date and Time Masks__]] for more information on how this mask is
constructed.//<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>>
<HTML><p></HTML>**[[http://www.telerik.com/help/winforms/editors-maskededitb
ox-numeric-masks.html|__Numeric__]] -**
[[http://www.telerik.com/help/winforms/editors-maskededitbox-numeric-masks.h
tml|The numeric mask]] is used when you need to display these
([[https://msdn.microsoft.com/en-us/library/dwhawy9k.aspx|defined on MS
```

```
site]]):<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>Percent
values.<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>Decimal
values.<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>Currency (culture-
aware).<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>Fixed point
values.<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML>>
<HTML><p></HTML>**[[http://www.telerik.com/help/winforms/editors-maskededitb
ox-standart-masks.html|__Standard__]] -** set the **MaskType** property to
**Standard** and the **Mask** property to "(###) ###-###". //See//
[[http://www.telerik.com/help/winforms/editors-maskededitbox-standart-masks.
html|__Standard Masks__]] //for more information on how this mask is
constructed.//<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>**Regex** - You can define
[[http://msdn.microsoft.com/en-us/library/2k3te2cs(v=vs.80).aspx|__Regex
expressions__]] as masks. For example mask **[A-z]** will check for at least
one symbol in this range (A-z) in MaskEditBox’s Text.<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>**IP** - allow user to input only **3**
digits in **0-255** range in **four** groups<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>**Email** - validate user input for the
valid mail. If this email is not valid will notify user with validation
icon:<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>‘TextMaskFormat ‘ - you may want to get a
RadMaskedEditBox.Value that differs from what the end-user sees in the
editor, or, in other words, from what the Text property provides. In such
cases the you will find the TextMaskFormat property useful. It allows you to
get variations of of the Text property value taking into account whether you
want to get the prompt characters, and the literals used in the mask. The
type of the TextMaskFormat property is the MaskFormat enum which provides
the following values:<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>‘IncludePromptAndLiterals’ (default value)
- The Value property returns Return the text input by the user as well as
any literal characters defined in the mask and any instances of the prompt
character.<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>‘ExcludePromptAndLiterals’ - The Value
property returns only the text input by the
user.<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>‘IncludePrompt’ - The Value property
returns the text input by the user as well as any instances of the prompt
character.<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>‘IncludeLiterals’ - The Value property
returns the text input by the user as well as any literal characters defined
```

```
in the mask.<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML></li></HTML><HTML></ul></HTML>

==== * RadPropertyGrid – rpg  ====

http://www.telerik.com/help/winforms/propertygrid-overview.html

http://www.telerik.com/community/forums/winforms/property-grid.aspx\\
Video:
http://www.telerik.com/videos/winforms/getting-started-with-telerik-radprope
rtygrid

Search/Filter PropertyGrid rows:
http://docs.telerik.com/devtools/winforms/propertygrid/features/filtering.ht
ml

Use the Tasks button to access these properties:

  * Enable Filtering
  * Enable Grouping
  * Enable Sorting
  * Show Helprich
  * Show SearchBox

Attach a RadPropertyGrid to an object and get a Visual Stuidio style
property editor for the object.\\
radPropertyGrid1.SelectedObject = new PropertyGridElement(); %%//%% point to
itself showing the PropertyGrid's own properties\\
using Telerik.WinControls.UI; %%//%% required for 'PropertyGridElement'

rpgControlProperties.SelectedObject = commandBarButton1; %%//%% Show the
properties of a RadCommandBar button

  * 'SelectedObject' - points to the object whose properties should be
displayed
  * 'HelpVisible' - show/hide the help section at the bottom which displays
the Description attribute of the selected property
  * 'HelpBarHeight' – in pixels of the help section at the bottom
  * 'ToolBarVisible' shows the toolbar where you would add RadButtonElements
for sort by category vs. alphabetical for example
  * ''Category' attribute of the SelectedObject property allows you to
create groups in the PropertyGrid (for sort by category vs. alphabetical)
  * 'PropertySort' sets default sorting to one of the enumerated values:
    * 'Alphabetical' – sorts the properties according to the SortOrder
setting (Ascending or Descending)
    * 'Categorized' – groups the properties according to their Category
attribute
    * 'CategorizedAlphabetical' – groups the properties according to their
Category attiribute and sorts them Alphabetically, according to the
SortOrder setting (Ascending or Descending)
    * 'None' – removes sorting and grouping
```

```
  * ‘SortOrder’ - Ascending, Descending and None
  * ‘ItemHeight’ - sets the visible items height.
  * ‘ItemSpacing’ - sets the spacing between the items.
  * ‘ItemIndent’ - sets the indent space in the beginning of the items.

The elements of the SelectedObject

Here’s some articles on using the RadPropertyGrid:

  *
[[http://www.telerik.com/help/winforms/propertygrid-editors-using-custom-edi
tor.html|Using Custom Editors]] (Replace a spin editor with a track bar
editor)

=== Force a RadPropertyGrid to update ===

>
radPropertyGrid.PropertyGridElement.PropertyTableElement.Update(PropertyGrid
TableElement.UpdateActions.Reset);

=== ToolTips on Value Column ===

Use the RadPropertyGrid.ToolTipTextNeeded event to specify the ToolTipText
for the value column and set the ToolTip.AutoPopDelay.

> private void radPropertyGrid1_ToolTipTextNeeded(object sender,
Telerik.WinControls.ToolTipTextNeededEventArgs e)
>
> {
>
> PropertyGridItemElement itemElement = sender as PropertyGridItemElement;
>
> if (itemElement!=null)
>
> {
>
> e.ToolTipText = itemElement.ValueElement.Text;
>
> e.ToolTip.AutoPopDelay = 1000;
>
> }
>
> }

==== RadRichTextBox – (deprecated) ====

http://www.telerik.com/help/winforms/richtextbox-overview.html

http://www.telerik.com/community/forums/winforms/richtextbox.aspx\\
Video:
http://www.telerik.com/videos/winforms/getting-started-with-telerik-radricht
```

```
extbox

Use the [[#radrichtexteditor---rrte|RadRichTextEditor]] instead.

==== RadRichTextEditor - rrte ====

http://docs.telerik.com/devtools/winforms/richtexteditor/overview

RadRichTextEditor is a control that is able to display and edit rich-text
content including formatted text arranged in pages, paragraphs, spans
(runs), tables, etc.

{{tdn:image92.png?525x404}}

RadRichTextEditor allows you to edit text and apply rich formatting options,
like:

  * Bold, Italic
  * Underline, Strike Through, Superscript and Subscript
  * Usage of all available system fonts
  * Text color and background
  * Bullet and numbered lists
  * Paragraph alignment and indentation
  * Show/Hide formatting symbols
  * Clear Formatting
  * Inserting pictures/symbols/hyperlinks
  * Table support – including nested tables, merging cells, setting
different RowSpan and ColumnSpan, tables with auto width and more.
  * ContextMenu - a standard one is provided out of the box, but it can be
easily customized or substituted for a user-defined one.
  * SelectionMiniToolbar - very similar to the one in MS Word, but can also
be replaced with another.
  * SpellChecker - extensible spell checking for different languages, using
built-in or custom dictionaries.
  * Printing - RadRichTextEditor supports printing out of the box.
  * Multi-level Undo/Redo Support - Telerik RadRichTextEditor provides
desktop-like usability and control thanks to its multilevel Undo/Redo
feature. It allows past actions to be reviewed and reversed, just like in
Microsoft Word.
  * Three types of layout – Paged, Flow (as text in an HTML page viewed in a
browser) and FlowNoWrap.
  * Multi-region selection - the built-in multi-region support enables you
to perform various operations for more than one selection simultaneously.
  * Rich-text clipboard - allows rich text copy/paste from Word, OpenOffice
etc.
  * Import/export - with Telerik RadRichTextEditor control you can load
XAML, HTML, RTF, DocX (rich-text) or TXT (plain text) into the control, you
can format and edit it and then export it back to any of the above formats
or PDF.
  * Headers and Footers - RadRichTextEditor supports headers and footers in
its document.
```

```
  * Track changes - adding and removing of text, images, hyperlinks, tables, etc. and formatting changes of these elements can be tracked by different users.
  * Bookmarks - provide easy retrieval of previously marked parts of the document.
  * Comments - offer the ability to add text to the document which is not inserted directly in the page.
  * Document protection - allows editing restrictions in the document for a set of users and groups.
  * Mail merge - makes populating a template document with data easy.
  * Custom annotations - enables you to extend the document model in order to serve specific needs.
  * Footnotes and Endnotes - usually used to show the source of a material or add an explanation.
  * Bibliographic references - can be used for referencing a source in the form of text or pointing the source of a citation.
  * Cross-references - references to a Heading, Caption or Bookmark can now change along with their source.
  * Captions for tables and figures - images, tables, etc. can now be easily labeled.
  * Document variables - a mechanism used for storing information in the document.
  * Code blocks - provide the ability to add source code fragments in a document.
  * IME support - RadRichTextEditor supports IME out the box.

=== Properties ===

  * .IsReadOnly = True to block editing
  * .SelectionEnabled = True to allow text selection
  * .IsSelectionMiniToolbarEnabled = False blocks this font toolbar popup on selected text
  * https://gyazo.com/5a11a797c43a8fe7f91f432de282efb7
    * This ContextMenu is still available: https://gyazo.com/d2f9184194167f7a833f7cbf507fc757

==== RadSpellChecker – rsc  ====

http://www.telerik.com/help/winforms/spellchecker-overview.html

RadSpellChecker enables developers to add multilingual spell checking capabilities to their Win Forms applications. The component is completely customizable and can be attached to any text-editing RadControl.

{{tdn:image93.png?420x315}}

==== * RadSpinEditor – rse  ====

http://www.telerik.com/help/winforms/editors-spineditor-overview.html\\
http://www.telerik.com/community/forums/winforms/editors.aspx
```

```
RadSpinEditor is a themable alternative to the standard Windows Numeric Up
Down control.

{{tdn:image94.png?110x32}}

==== * RadTextBox – rtxt  ====

http://www.telerik.com/help/winforms/editors-textbox-overview.html\\
http://www.telerik.com/community/forums/winforms/editors.aspx

[[http://docs.telerik.com/devtools/winforms/editors/textboxcontrol/radtextbo
xcontrol-vs-radtextbox|**RadTextBoxControl vs RadTextBox**]] - **RadTextBox
is a wrapper around the standard .NET TextBox control**, while
RadTextBoxControl is built entirely on top of Telerik Presentation
Framework.

Textbox that observes Telerik Theme settings

{{tdn:image95.png?278x159}}

  * 'Multiline' set to True for multiple line textbox. Then resize manually,
or Dock.
  * 'Enabled' - False greys out the control and restricts user interaction
  * IsReadOnly – True then the text is readonly
  * 'NullText' property specifies text to be displayed when the control has
no Text value. When the control receives the focus, this text is
automatically removed. Use this for things like ‘Search/Filter’ in a field
where the user will type to search.
  * 'PasswordChar' property specifies the character displayed in the text
box. For example, if you want asterisks displayed in the password box,
specify * for the PasswordChar property in the Properties window. Then,
regardless of what character a user types in the text box, an asterisk is
displayed. (Optional) Set the MaxLength property to limit password length.
  * CharacterCasing – forces entered characters to a specific case
  * See: http://msdn.microsoft.com/en-us/library/d3223ht2.aspx
  * See:
http://stackoverflow.com/questions/2555984/quick-question-how-to-set-a-text-
box-for-inputing-password-in-winforms

==== RadTextBoxControl – rtxtbc  ====

http://www.telerik.com/help/winforms/editors-textboxcontrol-overview.html\\
http://www.telerik.com/community/forums/winforms/editors.aspx

[[http://docs.telerik.com/devtools/winforms/editors/textboxcontrol/radtextbo
xcontrol-vs-radtextbox|**RadTextBoxControl vs RadTextBox**]] - **RadTextBox
is a wrapper around the standard .NET TextBox control**, while
RadTextBoxControl is built entirely on top of Telerik Presentation
Framework.

RadTextBoxControl provides similar to the standard .NET Framework text box
```

features. In addition it allows easy customizations and enhancements. It is generally used for editing text, although it can be used to display text, when in read-only mode. The control can display multiple lines, wrap text to the size of the control and perform text block formatting and replacement.

The control can display multiple lines, wrap text to the size of the control, add basic formatting, use themes, and //the main differences with RadTextBox - can be transparent and use gradients//.

==== * RadTimePicker – rtp ====

http://www.telerik.com/help/winforms/editors-timepicker-overview.html

http://tv.telerik.com/watch/winforms/getting-started-with-radtimepicker-for-winforms

http://www.telerik.com/community/forums/winforms/calendar-and-datetimepicker.aspx\\
Video:
http://www.telerik.com/videos/winforms/getting-started-with-radtimepicker-for-winforms

RadTimePicker control allows the end user to enter a time value to the editable area of the control, or pick the time from the drop down by selecting both the hours and the minutes. The drop down by default contains RadClock on the left side and two tables on the right side – for picking hours and minutes respectively.

{{tdn:image96.png?219x21}}

{{tdn:image97.png?249x195}}

Choose a time of day (not Date)

  * **‘Value’** – used to get or set the control value

> this.rtpStartTimeStd.Value = DateTime.Now; %%//%% Set Start time to now

  * **‘Culture‘** – Control the hour format (Standard [AM/PM], Military, or various International)

> timepicker1.Culture = new CultureInfo("en-gb"); %%//%% set to Military time
>
> timepicker1.Culture = new CultureInfo("en-us"); %%//%% set to Standard (AM/PM) time

  * ‘Step’ - gets or sets the desired minute’s interval
  * ‘ClockPosition’ – clock location in relation to the set time <Clock> dropbox (options: ClockBeforeTables, ClockAboveTables and HideClock)
  * ‘CloseButtonText’ – change the button text on the set time <Clock>

```
dropbox
  * ‘NullText’ – the text displayed in RadTimePickerElement when the Value
is null
  * ‘RowHeight’ – gets or sets the height of the rows in the hour/minutes
tables in the drop down
  * ‘ColumnsCount’ – gets or sets the number of the columns in the
hour/minutes tables in the drop down
  * ‘HeadersHeight’ – gets or sets the size of the headers (clock, hours and
minutes headers) in the drop down
  * ‘ButtonPanelHeight’ – sets or sets the height of the buttons panel at
the bottom of the drop down
  * ‘TableWidth’ – gets or sets the width of the hours and minutes tables in
the drop down
  * ‘TimeTables’ – allows you to choose between one table or two tables to
display both hours and minutes
  * ‘ReadOnly’ – makes the control read only – can’t type in it, or change
its value from the UI. Also the drop down cannot be opened

==== RadTrackBar – rtb  ====

http://www.telerik.com/help/winforms/track-and-status-controls-trackbar-over
view.html

http://www.telerik.com/community/forums/winforms/track-and-status.aspx

RadTrackBar, sometimes called a slider control, can be used for navigating a
large amount of information or for visually adjusting a numeric setting.

{{tdn:image98.png?369x270}}

There’s a bug that causes the bar to truncate if it’s physically shorter
than a % of the total.\\
rtbPlaybackSpeed.TrackBarElement.MinSize = new System.Drawing.Size(0, 0);
%%//%% Resolve by setting the MinSize of the TrackBarElement

==== # RadPopupEditor – rpe ====

http://docs.telerik.com/devtools/winforms/editors/popupeditor/popupeditor

Drop down a form where you can display/edit various controls/values
configurable at design time.

RadPopupEditor allows you to show any predefined or custom controls in its
popup. By default the control should be associated with a RadPopupContainer.
RadPoupContainer on the other hand allows you to build your layout at design
time. The controls added to the container will be shown in the popup window
when the popup is opened.

{{tdn:image99.png?233x185}}

From the Demo app:
```

```
{{tdn:image52.png?720x399}}

===== RadControls – Buttons =====

==== * RadButton – rbtn  ====

http://www.telerik.com/help/winforms/buttons-button-overview.html

http://www.telerik.com/community/forums/winforms/buttons.aspx

  * 'Text' \Behavior
  * 'DrawText' = False and no button text is displayed
  * 'Name' \Design
  * 'TextWrap' \Appearance (also must set 'AutoSize' = False, 'TextWrap' =
False, adjust 'Size' as required)
  * 'DialogResult' - Handles the close event of a form. Set to "Cancel" for
simple close. Set to "OK" for close but you can add saving code.

%%//%% Toggle the ‘Image’ on button click.

> %%//%% First add the 2 images as Project Resources (under Project
Properties)
>
> %%//%% Next create a property flag in the class to show the state
>
> namespace LS.Common.Log.Presentation
>
> {
>
> public partial class ucLogViewer : UserControl
>
> {
>
> public bool LogPaused { get; set; } %%//%% Play/Pause button state flag
property
>
> public ucLogViewer()
>
> {
>
> InitializeComponent();
>
> %%//%% Code continues…
>
> %%//%% Now create a click event handler to swap the image and update the
property flag
>
> private void rcbbtnPlayPauseLog_Click(object sender, EventArgs e) {
>
> if (LogPaused == true) {
>
```

```
> rcbbtnPlayPauseLog.Image =
LS.Common.Log.Presentation.Properties.Resources.Play_24x24;
>
> LogPaused = false;
>
> }
>
> else {
>
> rcbbtnPlayPauseLog.Image =
LS.Common.Log.Presentation.Properties.Resources.Pause_24x24;
>
> LogPaused = true;
>
> }
>
> }
>
> %%//%% NOTE: You could have used a bool variable (rather than a property)
and called some method(s) from the click event.

Show only an icon w/o button frame, like this 3<sup>rd</sup> button:\\
{{tdn:image100.png?256x48}}

> radButton3.ButtonElement.ShowBorder = false;
>
> radButton3.BackColor = System.Drawing.Color.Transparent;
>
> radButton3.DisplayStyle = Telerik.WinControls.DisplayStyle.Image;

Toggle between 2 image (icon) values

if (rbtnPlayPauseToggle.Text == "play") %%//%% use the .Text property as a
flag

{

rbtnPlayPauseToggle.Image = Properties.Resources.MediaPause_40; %%//%%
.MediaPause_40 is ‘MediaPause_40.png’ imported to the app repository

rbtnPlayPauseToggle.Text = "pause";

}

else

{

rbtnPlayPauseToggle.Image = Properties.Resources.MediaPlay_40; %%//%% toggle
to ‘MediaPlay_40.png’ image
```

```
rbtnPlayPauseToggle.Text = "play";

}

rbtnPlayPauseToggle.Refresh();

==== * RadCheckBox – rchk  ====

http://www.telerik.com/help/winforms/buttons-checkbox-overview.html

http://www.telerik.com/community/forums/winforms/buttons.aspx\\
http://docs.telerik.com/devtools/winforms/buttons/checkbox/handling-radcheck
box-states\\
http://docs.telerik.com/devtools/winforms/buttons/checkbox/databinding-radch
eckbox

RadCheckBox is designed to provide an interface element that can represent
an On or Off state using a check mark.

RadCheckBox supports three states. This is controlled by the IsThreeState
property. If it is set to false, the ToggleState property, which determines
the current state of the check mark, can be On or Off . Otherwise, it can be
set to On , Off , or Indeterminate .

CheckBox that observes Telerik Theme settings

  * 'text' text label
  * 'CheckAlignment' checkbox location in relation to the label
  * 'Checked' True/False (get & set) (somehow tied to 'ToggleState' as well)
  * ‘RightToLeft’ = Yes puts the text label to the left of the checkbox.

==== * RadDropDownButton – rddbtn  ====

http://www.telerik.com/help/winforms/buttons-dropdownbutton-overview.html

Each of the items of RadDropDownButton can be set to perform an action when
it is clicked. In addition, the items can contain other items, allowing you
to create any hierarchy that fits your needs of sub-items.

{{tdn:image101.png?190x114}}

==== RadSplitButton – rsbtn  ====

http://www.telerik.com/help/winforms/buttons-splitbutton-overview.html

http://www.telerik.com/community/forums/winforms/buttons.aspx\\
http://docs.telerik.com/devtools/winforms/buttons/splitbutton/working-with-r
adsplitbutton-items

RadSplitButton provides a menu-like interface contained within a button that
can be placed anywhere on a form.
```

```
{{tdn:image102.png?157x119}}

//RadSplitButton looks the same, and works nearly the same, as
RadDropDownButton, but RadSplitButton has a DefaultItem property to specify
the item whose Click event should be triggered when the button is clicked.
The functionality of RadSplitButton is "split" between the button itself and
the dropdown menu//.

  * ‘Items’ – the possible button contents: RadMenuItem,
RadMenuSeparatorItem, RadMenucomboItem, RadMenuButtonItem, RadMenuHeaderItem
  * ‘Image’ - property of the RadMenuItem enables you to include an icon
next to the Item
  * ‘DefaultItem’ sets the default by Item name Ex:
radSplitButton1.DefaultItem = mainItem;

==== * RadRadioButton – rrdo  ====

http://www.telerik.com/help/winforms/buttons-radiobutton-overview.html

http://www.telerik.com/community/forums/winforms/buttons.aspx

{{tdn:image103.png?128x81}}

Put multiple in a RadGroupBox

**Properties**

‘IsChecked’ = True, button is filled in

==== RadRepeatButton – rrbtn  ====

http://www.telerik.com/help/winforms/buttons-repeatbutton-overview.html

http://www.telerik.com/community/forums/winforms/buttons.aspx\\
http://docs.telerik.com/devtools/winforms/buttons/repeatbutton/working-with-
radrepeatbutton

RadRepeatButton provides press-and-hold functionality. While the mouse
button is pressed down, the ButtonClick event fires at a pre-determined
interval. RadRepeatButton is an ideal UI element for allowing users to
control an increasing or decreasing value, such as volume or brightness.

{{tdn:image104.gif?322x157}}

==== RadToggleButton – rtbtn  ====

http://www.telerik.com/help/winforms/buttons-togglebutton-overview.html

http:%%//%%docs.telerik.com/devtools/winforms/buttons/togglebutton/handling-
radtogglebutton-states
```

```
http://www.telerik.com/community/forums/winforms/buttons.aspx

RadToggleButton is designed to manage states on your form. It shares many
features with the RadCheckBox, but provides a different visual effect than
the standard check mark.

{{tdn:image105.png?180x117}}

  * 'ToggleState' On/Off
  * 'Text' \Behavior

To wrap the text label, 'AutoSize' = False, 'TextWrap' = True, adjust 'Size'
as required.

  * 'Name' \Design

==== RadToggleSwitch – rts ====

http://docs.telerik.com/devtools/winforms/buttons/toggleswitch/toggleswitch

RadToggleSwitch is a control designed to represent two states: e.g.
true/false, On/Off, etc.

{{tdn:image106.gif?130x92}}

===== RadControls – Menus & Toolbars =====

http://www.telerik.com/help/winforms/menus-menu-overview.html

==== RadApplicationMenu – ram ====

http://www.telerik.com/community/forums/winforms/menus.aspx

RadApplicationMenu is the Telerik counterpart of the application menu that
displays controls used to perform actions on entire documents and forms,
such as Save and Print. It also provides a list of recent documents, access
to form options, and the ability to exit the form or application.

{{tdn:image107.png?271x192}} {{tdn:image108.png?290x203}}

See [[#radribbonbar-rrb|RadRibbonBar]] for more property info.

  * ‘Text’ – shows up on the button itself
  * ‘ButtonItems’ collection defines buttons on the bottom of the dropdown.
For example the RadRibbonBar comes with an <Options> button here by default.

==== RadContextMenu – rcm ====

http://www.telerik.com/community/forums/winforms/menus.aspx

To implement context menus use RadContextMenu in your application.
```

```
RadContextMenu is a non-visual component that sits in the component tray
located below the form design surface. RadContextMenu, like RadMenu, can be
themed and has an items collection that accepts RadMenuItem,
RadMenuComboBoxItem, RadMenuSeparatorItem and RadMenuContextItem.

{{tdn:image109.png?272x247}}

**Create a right click (context) menu.**

Add to a form then link to a control via the control’s ‘RadContextMenu’
property.

Create a [[#telerik-systray-app|Systray app]] menu by binding this to a
Windows NotifyIcon (niMain).

RadContextMenuItem – rcmi

> These are the items that appear in the right click menu.

  * ‘Visibiility’ – hide a menu item
  * ‘Enabled’ – set the text to grey and disable user interaction

> Use an item as an information display only. You don’t want the highlight
on mousehover or the click event to fire, but you don’t want the greyed out
text of a disabled control.
>
> rmiLastSync.Enabled = false; %%//%% disable highlight on mousehover and
click event
>
> rmiLastSync.UseDefaultDisabledPaint = false; %%//%% control is disabled
but don't show as grey text

==== RadContextMenuManager – rcmm ====

[[http://www.telerik.com/help/winforms/menus-context-menu-assign-radcontextm
enu-to-telerik-and-non-telerik-
controls.html|__http:%%//%%www.telerik.com/help/winforms/menus-context-menu-
assign-radcontextmenu-to-telerik-and-non-telerik-controls.html__]]

You can assign RadContextMenu to Telerik and non-Telerik controls. This can
be achieved by making use of RadContextMenuManager component. This component
will add a RadContextMenu property to all controls on the form. Then, you
should simply set a RadContextMenu instance to the RadContextMenu property.

==== * RadCommandBar – rcb  ====

http://www.telerik.com/help/winforms/commandbar-overview.html

http:%%//%%docs.telerik.com/devtools/winforms/commandbar/design-time

http://www.telerik.com/community/forums/winforms/commandbar.aspx
```

```
Video:
[[http://www.telerik.com/videos/winforms/working-with-radcommandbar-for-winf
orms|Working with RadCommandBar]]\\
[[http://docs.telerik.com/devtools/winforms/commandbar/save-and-load-layout|
Save and Load Layout to XML file]]

RadCommandBar is a fully themeable tool strip that provides unprecedented
flexibility. More than just a collection of buttons, RadCommandBar hosts any
RadControl, including combo boxes, text boxes, split buttons, drop-down
buttons, toggle buttons and more. CommandBar can be moved, rearranged and
resized at run time for easy end-user customization.

{{tdn:image110.png?566x65}}

Basically a Toolbar. Can add buttons, dropdowns, etc.

[[http://docs.telerik.com/devtools/winforms/commandbar/structure|The
structure]] works like this:

  * RadCommandBar
    * CommandBarRowElement - rcbre (1 or more rows)
      * CommandBarStripElement –rcbse (1 or more strips within a row)
        * ‘DisplayName’ is visible to users at runtime in the Overflow
button dialog
        * ‘StretchHorizontally’ to make a Strip fill the available width on
the Row.
        * RadCommandBarGripButton
        * Items (various)
          * CommandBarButton - rcbbtn
          * CommandBarDropDownButton - rcbddbtn
          * CommandBarDropDownList - rcbddl
          * CommandBarHostItem – can host other controls like date selector
or checkbox
          * CommandBarSeparator
          * CommandBarLabel - rcblbl
          * CommandBarTextBox - rcbtxt
          * CommandBarToggleButton
          * CommandBarSplitButton
        * RadCommandBarOverflowButton

//**Select a CommandBarRowElement then click the Task button to ‘Edit
Strips’. You can reorder the Strips within a row from this dialog. Choose a
Strip to edit, then use the ‘Items’ property to add/edit items. Or can
select a Strip directly from the Row and click the Tasks buton to edit only
the Items for that Strip. Set the Strip ‘DisplayName’ so it’s visible from
the Overflow button dialog at runtime. Set ‘StretchHorizontally’ to make a
Strip fill the available width on the Row.**//

RadCommandBar allows the user to show/hide, reorder items and rearrange
strip elements on different rows at runtime. ### We need to build a common
lib class that saves/restores those customizations. The Save/Load layout
```

```
functionality gives your applications the opportunity to preserve user
settings concerning position, visibility and orientation.

http://www.telerik.com/help/winforms/commandbar-save-and-load-layout.html

Roll up empty space left behind from visible/collapsed ToolStrips\\
https://www.telerik.com/forums/problem-in-stripelement-visible-collapsed

https://www.telerik.com/forums/order-of-strips-to-display

[[#add-resources-to-a-solution-icons-graphics-files-etc.|Add Resources to a
Solution (Icons, Graphics, Files, etc.)]]

[[file:///\\192.168.93.1\KB_MiM\Telerik\20160204)%20Add%20Icon%20Graphics%20
to%20a%20VS%20Solution%20(Project%20Resource%20File).mp4|Here’s my updated
video on how to add Icon Graphics to a
Solution]]{{tdn:image111.png?395x267}}

The default icons are all small (16x16) but if you use a bigger icon (32x32
or 64x64) the RadCommandBar will expand as required. You can also force it
in code. For example show 32x32 image in CommandBarButton like this:\\
1. Use image with size 32x32.\\
2. Set the MinSize if you want to have space between image and borders of
the button.

> this.commandBarButton3.Image =
global::CommandBarButtonDefaultSize.Properties.Resources.Button_Next_icon32x
32;\\
> this.commandBarButton3.MinSize = new System.Drawing.Size(40, 40);

You can add the items by clicking the down arrow on the RCB, but it’s easier
to add and organize if you open the RCB ‘Collection Editor’. You can select
multiple items to set common properties.

**Set the width to match the container**

???

**Show an icon and text on a control (button, combo, etc) including set
button size for text wrap**

//NOTE: You can <Ctrl> click all the buttons to set these properties at
once.//

  * **'DrawText' = True**
  * **'TextImageRelation' = ImageAboveText** or ImageBeforeText (for text
beside the icon)
  * **‘TextWrap’ – True** if long text. Also set:
    * **‘AutoSize’** = False
    * **‘Size’ = 55,65** for 32x32 image. Adjust as required.
  * <del>Not necessary -- Set 'AutoSize' = False</del>
```

```
  * <del>Not necessary -- 'ImageAlignment' = MiddleLeft & TextAlignment' =
MiddleRight</del>
  * **‘DisplayName’** – //Be sure to set this one equal to the text property
or it won’t have a real name in the toolbar configuration control (from
RadCommandBarOverflowButton).//
  * ‘Image’ – clear to remove the icon, or set to a desired icon image
  * ‘visible’ = false to hide the commandbar collapsing the space it
occupies
  * CommandBarRowElement (Rows Collection) – a horiz or vert row where a
StripElement (toolbar) is shown
  * CommandBarStripElement (Strips Collection) – basically a single,
draggable toolbar on the CommandBar (which can hold multiple toolbars)
  * DisplayName – The name shown to the user for a StripElement or
subcontrol the user sees when customizing the toolbar at runtime.
  * RadCommandBarGripButton
  * Orientation - the orientation in RadCommandBar can be set via the
Orientation property:

radCommandBar1.Rows[0].Strips[0].Orientation = Orientation.Vertical;

  * ‘VisibleInStrip’ Item property to hide a control from the Strip.
  * [[http://docs.telerik.com/devtools/winforms/commandbar/item-types|Item
Types]] (arranged in ItemsLayout)
    * **CommandBarButton** – rcbbtn
      * ‘DrawText’ = True to display the text of the button. Use the
‘TextImageRelation’ property to control the layout of image and text:
Overlay, **ImageAboveText**, TextAboveImage, **ImageBeforeText**,
TextBeforeImage.
    * **CommandBarDropDownButton** – rcbddbtn (Drop down items are rcbddbtni
)
      * ‘Text’ – the text on the button
      * ‘Items’ – rcbddbtni (the buttons that appear in the dropdown)

>
https://www.telerik.com/forums/remove-left-side-extra-space-from-dropdownbut
tons
>
> Here’s how to have the main button display text, then have that text
change based on the selected button (rcbddbtni - ‘Items’) from the dropdown.

<HTML><ul></HTML>
<HTML><li></HTML>> <HTML><p></HTML>Set the DropDownButton ‘Text‘ to the
default value you want to display.<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>Add all the DropDownButton ‘Items’ and
set their ‘Text’ properties to the text you want to display if they are
selected. Make note of each time control name.<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>You have to add a Click Event to each
DropDownButton Item (not the parent DropDownButton itself). Use the VS
Property Grid dropdown to find each Item (that’s why we remembered them
```

```
before). Then add a Click event. Here’s a couple of examples of click events
that change the Text property on the CommandBarDropDownButton parent
control.<HTML></p></HTML>
<HTML></li></HTML><HTML></ul></HTML>

> %%//%% 'Show entire event' Entire vs. Entry Only drop down button
>
> private void rcbddbtniEntireEvent_Click(object sender, EventArgs e) {
>
> rcbddbtnShowEntryVEvent.Text = rcbddbtniEntireEvent.Text;
>
> }
>
> private void rcbddbtniEntryOnly_Click(object sender, EventArgs e) {
>
> rcbddbtnShowEntryVEvent.Text = rcbddbtniEntryOnly.Text;
>
> }

  * **CommandBarDropDownList** – rcbddl (Drop down items are rcbddli )
  * **CommandBarHostItem** – rcbhi (Item that can host any RadElement. Use
the HostedElement property to get/set the element that you are hosting.
[[http://www.telerik.com/community/forums/winforms/commandbar/add-a-datetime
picker-to-a-commandbar.aspx|For example a RadDateTimePicker]] and here’s how
to
[[http://www.telerik.com/community/forums/winforms/commandbar/read-value-fro
m-datetimepicker-in-commandbar.aspx|‘Read Value from DateTimePicker in
CommandBar’]])

> Note: You must use a **CommandBarLabel** with a **CommandBarHostItem**.
The built in ‘Text’ value won’t display.
>
> %%//%% RadDateTimePicker and RadTimePicker 'CommandBarHostItem's on a
RadCommandBar
>
> %%//%% Stick this right after “InitializeComponent();” on the form or UC.
>
> SetupHostedItems(); %%//%% setup the RadDateTimePicker and RadTimePicker
'CommandBarHostItem's on the RadCommandBar
>
> Follow up (what’s a better way to say these follow after the function that
include the Inititalize()?) with these:
>
> private void SetupHostedItems() %%//%% setup the 'CommandBarHostItem's on
the RadCommandBar
>
> {
>
> rcbhiStartDate.HostedItem = GenerateDateTimePicker("rdtpStartDate");
%%//%% Start Date
>
```

```
> rcbhiStartTime.HostedItem = GenerateTimePicker("rtpStartTime"); %%//%%
Start Time
>
> rcbhiEndDate.HostedItem = GenerateDateTimePicker("rdtpEndDate"); %%//%%
End Date
>
> rcbhiEndTime.HostedItem = GenerateTimePicker("rtpEndTime"); %%//%% End
Time
>
> }
>
> private RadDateTimePickerElement GenerateDateTimePicker(string Name)
%%//%% 'CommandBarHostItem' on the RadCommandBar
>
> {
>
> RadDateTimePickerElement rdtp = new RadDateTimePickerElement();
>
> rdtp.AutoSizeMode =
Telerik.WinControls.RadAutoSizeMode.FitToAvailableSize;
>
> rdtp.Name = Name;
>
> rdtp.MinSize = new System.Drawing.Size(100, 20);
>
> rdtp.Format = DateTimePickerFormat.Short;
>
> return rdtp;
>
> }
>
> private RadTimePickerElement GenerateTimePicker(string Name) %%//%%
'CommandBarHostItem' on the RadCommandBar
>
> {
>
> RadTimePickerElement rtp = new RadTimePickerElement();
>
> rtp.AutoSizeMode = Telerik.WinControls.RadAutoSizeMode.FitToAvailableSize;
>
> rtp.Name = Name;
>
> rtp.MinSize = new System.Drawing.Size(100, 20);
>
> return rtp;
>
> }
>
> **Checkbox as CommandBarHostItem**
>
> https://www.telerik.com/forums/checkbox-in-radcommandbar
```

```
>
> Create a RadCommandBar HostedItem in the IDE, then create a RadCheckBox in
code and link it to the HostedItem.
>
> %%//%% Create a RadCheckBox ‘rchkOnlyFavoriteColumns’ and link it to the
CommandBar HostedItem ‘rcbhiOnlyFavoriteColumns’
>
> RadCheckBoxElement rchkOnlyFavoriteColumns = new RadCheckBoxElement();
>
> rchkOnlyFavoriteColumns.MinSize = new System.Drawing.Size(19, 0); %%//%%
Set the size here to match the HostedItem from the IDE
>
> rcbhiOnlyFavoriteColumns.HostedItem = rchkOnlyFavoriteColumns;

  * **CommandBarSeparator**
  * **CommandBarLabel** - rcblbl
  * **CommandBarTextBox**
  * **CommandBarToggleButton** – rcbrtb\\
https://www.telerik.com/forums/commandbartogglebutton-appearance-when-disabl
ed
  * **CommandBarSplitButton** - rcbsb

> **RadCommandBarOverflowButton** - The overflow button automatically
displays items that don't have the real estate to display by default. The
end user can also customize the toolstrip by adding and removing buttons.

<HTML><ul></HTML>
<HTML><li></HTML>>
<HTML><p></HTML>[[http://www.telerik.com/help/winforms/commandbar-howto-cust
omize-the-overflow-button.html|Customize the contents of the
OverflowButton]]<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>Hide the Overflow button\\
> radCommandBarStripElement1.OverflowButton.Visibility =
Telerik.WinControls.ElementVisibility.Collapsed; %%//%% Or should you use
the ‘Hidden’ property?<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>Show the Overflow button only when
necessary as you resize the form<HTML></p></HTML>
<HTML></li></HTML><HTML></ul></HTML>

> Use the ItemOverflowed and ItemOutOfOverflow events of
CommandBarStripElement to show the overflow button only when needed and hide
it if not needed. See:
http://www.telerik.com/forums/hide-remove-add-or-remove-buttons\\
> protected override void OnLoad(EventArgs e)
>
> {
>
> base.OnLoad(e);
>
```

```
> AddCommandBar();
>
> radCommandBar1.Rows[0].Strips[0].ItemOverflowed += Form1_ItemOverflowed;
>
> radCommandBar1.Rows[0].Strips[0].ItemOutOfOverflow +=
Form1_ItemOutOfOverflow;
>
> }
>
> void Form1_ItemOutOfOverflow(object sender, EventArgs e)
>
> {
>
> if (!radCommandBar1.Rows[0].Strips[0].HasOverflowedItems)
>
> {
>
> radCommandBar1.Rows[0].Strips[0].OverflowButton.Visibility =
ElementVisibility.Collapsed;
>
> }
>
> }
>
> void Form1_ItemOverflowed(object sender, EventArgs e)
>
> {
>
> radCommandBar1.Rows[0].Strips[0].OverflowButton.Visibility =
ElementVisibility.Visible;
>
> }

<HTML><ul></HTML>
<HTML><li></HTML>> <HTML><p></HTML>Programmatically hide a button on the
CommandBar\\
> %%//%% Hide the Documents button on the CommandBar\\
> rcbbtnDocuments.VisibleInStrip = false; %%//%% removes it from
toolbar<HTML></p></HTML>
<HTML></li></HTML><HTML></ul></HTML>

> rcbbtnDocuments.VisibleInOverflowMenu = false; %%//%% removes it from
overflow too (user can't see at all)
>
> %%//%% We don’t know how to remove it from the Customize dialog

If you Dock a panel (or TableLayoutPanel) control to the form and it appears
behind the RadCommandBar, collapse the panel in the Document Outline, then
drag it above the RadCommandBar like this:\\
{{tdn:image112.png?232x124|C:\Users\rgray\AppData\Local\Temp\SNAGHTML4d8bb2c
2.PNG}}
```

```
==== * RadMenu – rm ====

http://www.telerik.com/help/winforms/menus-menu-overview.html\\
[[http://www.telerik.com/community/forums/winforms/menus.aspx|__http:%%//%%w
ww.telerik.com/community/forums/winforms/menus.aspx__]]

RadMenu enables you to integrate attractive and flexible menus on Forms
within your Windows applications.

{{tdn:image113.png?300x115}} {{tdn:image114.png?91x116}}

{{tdn:image115.png?306x238}} {{tdn:image116.png?387x225}}

{{tdn:image117.png?509x51}}{{tdn:image118.png?163x211}}

Other controls, like the top of a RadPanel, can appear behind a RadMenu. I'm
not sure how to force them both to the same layer, but you can use the
Margin – Top to push the hidden control down. Look for a better solution.
**The best thing to do is to add the panel after the RadMenu. //Also look to
my instructions on RadCommandBar to deal with controls behind this one. It's
about setting the order in the IDE Document Outline.//**

RadMenuItem – rmi

>
http://www.telerik.com/help/winforms/menus-menu-working-with-radmenu-items-r
admenuitem.html

RadMenuButton – rmb

Menu child items are under the RadMenu.Items (Collection). Then select a
child and look to its RadMenu.Items (Collection) for the grandchildren
items.

  * 'Visibility' – hide a menu item
  * 'Enabled' – set the text to grey and disable user interaction

If you Dock a panel (or TableLayoutPanel) control to the form and it appears
behind the RadMenu, collapse the panel in the Document Outline, then drag it
above the RadMenu like this:\\
{{tdn:image26.png?247x147|C:\Users\rgray\AppData\Local\Temp\SNAGHTML221ed26e
.PNG}}

=== Standard Legality Software Help Menu Setup ===

  - Add a radMenu control to rfrmMain and name it 'radMenu'
  - <del>Open rfrmMain.Designer.cs and add these lines directly under
'private Telerik.WinControls.UI.RadMenu radMenu;'</del>
  - Create these RadMenuItems **(Add to 'ActiveItems' under top level menu
items, and 'Items' for sub level menu items)**
```

```
> private Telerik.WinControls.UI.RadMenuItem rmiHelp;
>
> private Telerik.WinControls.UI.RadMenuItem rmiHelpOnline;
>
> private Telerik.WinControls.UI.RadMenuItem rmiLegalitySoftwareWebsite;
>
> private Telerik.WinControls.UI.RadMenuItem rmiLicenseRegistration;
>
> private Telerik.WinControls.UI.RadMenuItem rmiAbout;

//Note you might also want to include these items://

> private Telerik.WinControls.UI.RadMenuItem rmiFile;
>
> private Telerik.WinControls.UI.RadMenuItem rmiNew;
>
> private Telerik.WinControls.UI.RadMenuItem rmiOpenSettings;
>
> private Telerik.WinControls.UI.RadMenuItem rmiSaveSettings;
>
> private Telerik.WinControls.UI.RadMenuItem rmiSaveSettingsAs;
>
> %%//%% Add a separator then a Recent Items here – See File Renamer example
code example
>
> private Telerik.WinControls.UI.RadMenuItem rmiExit;

<HTML><ol start="4" style="list-style-type: decimal;"></HTML>
<HTML><li></HTML><HTML><p></HTML>Click “TELERIK | RadControls for WinForms |
Add RadAboutBox”. Name the form “rfrmRadAboutBox” and click
<Add>.<HTML></p></HTML><HTML></li></HTML>
<HTML><li></HTML><HTML><p></HTML>Add this to the top of
rfrmMain.cs<HTML></p></HTML><HTML></li></HTML><HTML></ol></HTML>

using System.Diagnostics; %%//%% required by ‘Process’ to launch external
apps (like the browser)

<HTML><ol start="6" style="list-style-type: decimal;"></HTML>
<HTML><li></HTML><HTML><p></HTML>Use this code to rfrmMain.cs to handle the
Help Menu clicks<HTML></p></HTML><HTML></li></HTML><HTML></ol></HTML>

> %%//%% Help Menu
==========
>
> private void rmiHelpOnline_Click(object sender, EventArgs e)
>
> {
>
> %%//%%TODO: FutureFeature - Implement online help system (wiki)
>
> Process.Start("http:%%//%%www.legalitysoftware.com"); %%//%% launch the
```

```
Legality Software site
>
> }
>
> private void rmiLegalitySoftwareWebsite_Click(object sender, EventArgs e)
>
> {
>
> Process.Start("http:%%//%%www.legalitysoftware.com"); %%//%% launch the
Legality Software site
>
> }
>
> private void rmiLicenseRegistration_Click(object sender, EventArgs e)
>
> {
>
> %%//%%TODO: FutureFeature - Implement licensing
>
> }

private void rmiCheckForUpdates_Click(object sender, EventArgs e)

{

%%//%%TODO: Larry - Wire up the Update feature

> }
>
> private void rmiAbout_Click(object sender, EventArgs e)
>
> {
>
> using (var rfrmAboutBox = new rfrmRadAboutBox()) %%//%% create an instance
of the form
>
> {
>
> rfrmAboutBox.ShowDialog(this); %%//%% then show/opens it
>
> }
>
> }

==== * RadRibbonBar – rrb  ====

http://www.telerik.com/help/winforms/ribbonbar-overview.html

http://www.telerik.com/community/forums/winforms/ribbonbar.aspx\\
Video:
[[http://tv.telerik.com/winforms/radribbonbar/radribbonbar-overview|Overview
```

```
Video]]

With the RadRibbonBar control you can build user interfaces similar to those
used in Microsoft Office.

{{tdn:image119.png?383x309}}

//**Consider adding a [[#radribbonform|RadRibbonForm]] to the Solution
rather than adding a RadRibbonBar to a regular RadForm.**//

  * The StartButtonImage property specifies an image to use for the <Start>
button in the upper left corner of the control.
  * **‘CommandTabs’ collection defines the ‘RibbonTab’ tabs across the top**
  * ‘contextualTabGroup’ collection defines areas across the top of the
form. I assume to control tab visibility by context.
  * ‘QuickAccessToolbarItems’ collection defines QAT items
  * ‘StartMenuItems’ collection defines the items under the round <Start>
button.
  * ‘StartMenuRightColumnItems’ collection defines the items in the right
column under the round <Start> button.

> RadRibbonTab – rrbt

  * ‘Items’ collection defines ‘RadRibbonBarGroup’ items - button container
groups on the current RadRibbonTab

> RadRibbonBarGroup – rrbg

<HTML><ul></HTML>
<HTML><li></HTML>> <HTML><p></HTML>‘Items’ collection defines ‘RadMenuItems’
(controls - buttons, etc) or ‘RadRibbonBarButtonGroup’ containers which
include a border and allow vertical orientation.<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>Set Buttons for AutoSize = False,
TextWrap = True and set Size/Width = narrower so the text will
wrap<HTML></p></HTML>
<HTML></li></HTML>
<HTML><li></HTML>> <HTML><p></HTML>I’m not sure how to center buttons in a
group when the group name is wider than the list of
buttons.<HTML></p></HTML>
<HTML></li></HTML><HTML></ul></HTML>

> RadRibbonBarButtonGroup – rrbbg

  * ‘Orientation’
    * ‘Horizontal’ default. Controls side by side.
    * ‘Vertical’ Allows you to stack controls vertically
  * ‘Items’ collection defines ‘RadMenuItems’ (controls - buttons, etc)

**RadButtonElement - rbe - Position text to BottomCenter, enable TextWrap
and config an icon**
```

```
  * Select one or more buttons and set ‘TextAlignment’ = ‘BottomCenter’
  * By default the text sets the button width, but you probably should set
size manually. These are good defaults:
    * AutoSize = False
    * TextWrap = True
    * Size/Width = small enough to wrap the text to 2 lines (assuming you
also want a small icon). <del>Good values are 64, 74 for a button with 2
lines of text and 61, 60 for buttons with one line of text. These will align
the top of the top lines.</del>
    * Size/Height = 66 is a good default. 68 starts to push down the
ButtonGroup label but is an acceptable max. <del>Set the ‘Height’ to force
single text to line up with the top of the text on 2 line text buttons. Good
options are 74 high for buttons with 2 lines of text and 60 for buttons with
one line. All this assumes you use in icon on the buttons as well.</del>
    * TextImageRelation – ImageAboveText
    * TextAlignment = BottomCenter.
      * Dealing with buttons with both 1 and 2 lines of text. Maybe better
than above, use bottom padding to push single line text up from the bottom.
Turn on TextWrap. Set button size to X,66, where x is the width that forces
any required text wrap. Set Padding.Top to 2-8 to push the icon down and
Padding.Bottom to 0 for buttons with 2 lines of text and 15 for buttons with
1 line of text.
  * ‘Image’ = ??? to set the icon graphic image. Choose “Local resource” |
<Import> | browse for graphic and finish. RibbonBar icons should be 32x32
pixels.
    * ‘ImageAlignment’ = TopCenter
    * Padding/Top = 8 (or so) to push the graphic down to the center of the
button (assuming the above sizes)

**Hide elements of the Ribbon Bar**

Add the following code right under ‘InitializeComponent();’ in frmMain() (or
whatever the form is called).

%%//%% Stick this right after “InitializeComponent();” on the form

%%//%% Hide the AppButton (Office Start button or 'File' tab) on the
RadRibbonBar

radRibbonBar1.RibbonBarElement.ApplicationButtonElement.Visibility =
Telerik.WinControls.ElementVisibility.Hidden;

%%//%% Hide the QAT (Quick Access Toolbar) on the RadRibbonBar

radRibbonBar1.RibbonBarElement.QuickAccessToolBar.Visibility =
Telerik.WinControls.ElementVisibility.Hidden;

%%//%%Hide Title Bar - collapse all 3 elements to hide this bar

richTextEditorRibbonBar.RibbonBarElement.QuickAccessToolBar.Visibility = Ele
mentVisibility.Collapsed;
```

```
richTextEditorRibbonBar.RibbonBarElement.ApplicationButtonElement.Visibility
 = ElementVisibility.Collapsed;

richTextEditorRibbonBar.RibbonBarElement.RibbonCaption.Visibility = ElementV
isibility.Collapsed;

%%//%% Hide any RibbonBar tab with 'Name' = "Hidden" or where 'Text'
contains "Hidden" (just the tab, not content)

foreach (Telerik.WinControls.UI.RibbonTab item in radRibbonBar1.CommandTabs)
{

if (item.Text.ToLower().Contains("hidden") == true ||
item.Name.ToLower().Contains("hidden") == true)

item.Visibility = Telerik.WinControls.ElementVisibility.Hidden; %%//%%
sometimes ‘Collapsed’

}

%%//%% Shift Tab Groups to the left when the AppButton (Office Start button
or 'File' tab) is collapsed

radRibbonBar1.RibbonBarElement.ApplicationButtonElement.Visibility = Telerik
.WinControls.ElementVisibility.Collapsed;

Padding p = radRibbonBar1.RibbonBarElement.TabStripElement.ItemContainer.Pad
ding;

radRibbonBar1.RibbonBarElement.TabStripElement.ItemContainer.Padding = new P
adding(0, p.Top, p.Right, p.Bottom);

%%//%% Hide the ‘ribbonTab1’ tab on the RadRibbonBar\\
ribbonTab1.Visibility = Telerik.WinControls.ElementVisibility.Hidden; %%//%%
sometimes ‘Collapsed’

%%//%% Hide the Caret button in the upper right that collapses the RibbonBar

this.radRibbonBar1.RibbonBarElement.ExpandButton.Visibility =
Telerik.WinControls.ElementVisibility.Collapsed;

%%//%% Select the 1st tab [0] on the RadRibbonBar by default

((Telerik.WinControls.UI.RibbonTab)radRibbonBar1.CommandTabs[0]).IsSelected
= true;

Use the ‘Document Outline’ view (<Ctrl>+W, U) to [[#radribbonform|manage a
RadRibbonForm]].

**Managing & Reordering RibbonBar Groups**
```

```
  * Select the RadRibbonBar, and click the <...> on the CommandTabs Property
to open RadRibbonBarCommandTab Collection Editor.
  * Select a RadRibbonBarTab on the left, then click the <...> on the Items
property to open the RadItem Collection Editor. <Add> and reorder Groups.
  * Select a RadRibbonBarGroup on the left, then click the <...> on the
Items property to open the RadItem Collection Editor. <Add> and reorder
buttons, etc.

> NOTE: Reordering CommandTabs in the wizard doesn’t seem to work. You can
use this code instead:
>
>
%%//%%The RibbonGroups 'rrbctArchives' and 'rrbctNotes' should be in the 1st
 and 2nd positions from the left. Telerik support ticket: 1453535
>
> this.richTextEditorRibbonBar.CommandTabs.Remove(this.rrbctArchives);
>
> this.richTextEditorRibbonBar.CommandTabs.Insert(0, this.rrbctArchives);
>
> richTextEditorRibbonBar.CommandTabs.Remove(this.rrbctNotes);
>
> richTextEditorRibbonBar.CommandTabs.Insert(1, this.rrbctNotes);

**Set default selected CommandTab**

> %%//%%Set Default Startup Tab
>
>
SetSpecifiedPageViewByText(radPageView, ((richTextEditorRibbonBar.SelectedCo
mmandTab.Text == "Archives") ? "Archives" : "Notes"));

=== Add RadDateTimePicker to RadRibbonBar ===

This control is not available in the RadRibbonBar Items collection, but you
can add it in code.

We wanted 2 date controls followed by a dropdown list of predefined date
ranges. Add the dropdown list in the designer, then insert the 2 date
controls above it (top down) in code.

{{tdn:image120.png?171x134}} {{tdn:image121.png?96x110}}

> public frmMain()
>
> {
>
>     InitializeComponent();
>
>     SetupRibbonFilters();  %%//%% add non-
standard controls to RadRibbonBar
>
```

```
> }
>
> void SetupRibbonFilters()
>
> {
>
>     AddDateTimeControl("rdtpStartDate");
>
>     AddDateTimeControl("rdtpEndDate");
>
> }
>
> void AddDateTimeControl(string controlName)
>
> {
>
>     RadDateTimePicker dtPicker = new RadDateTimePicker();
>
>     dtPicker.Name = controlName;
>
>
    %%//%%dtPicker.ThemeName = "Office2007Blue";  %%//%% inherits the form t
heme if not specified
>
>     RadHostItem hostItem = new RadHostItem(dtPicker); %%//%% placeholder
>
>     hostItem.MinSize = new Size(135, 25);
>
>     hostItem.MaxSize = new Size(150, 25);
>
>
    %%//%% rrbbgDateFilter.Items.Add(hostItem);  %%//%% add new control to t
he specified RibbonGroup
>
>     rrbbgDateFilter.Items.Insert(rrbbgDateFilter.Items.Count -
 1, hostItem);  %%//%% insert new control from top down
>
> }
```

=== Add CheckedDropDownList to RadRibbonBar ===

This control is not available in the RadRibbonBar Items collection, but you can add it in code.

{{tdn:image122.png?226x124}}

```
> public frmMain()
>
> {
>
>     InitializeComponent();
```

```
>
>     SetupRibbonFilters();  %%//%% add non-
standard controls to RadRibbonBar
>
> }
>
> void SetupRibbonFilters()
>
> {
>
>     AddOtherPartyDropDown();
>
> }
>
> void AddOtherPartyDropDown()
>
> {
>
>
    Dictionary<int, string> other_parties = new Dictionary<int, string>();
%%//%% create sample data
>
>     other_parties.Add(0, "Joe Bob");
>
>     other_parties.Add(1, "Jim Bob");
>
>     other_parties.Add(2, "Bob Bob");
>
>     other_parties.Add(3, "Leroy Bob");
>
>
    AddCheckedDropDownList("rcddOtherParty", "key", "value", other_parties);
>
> }
>
>
void AddCheckedDropDownList(string controlName, string dataMember, string va
lueMember, object dataSource)
>
> {
>
>     RadCheckedDropDownList rcddList = new RadCheckedDropDownList();
>
>     rcddList.Name = controlName;
>
>     rcddList.DataMember = dataMember;
>
>     rcddList.ValueMember = valueMember;
>
>     rcddList.DataSource = dataSource;
>
```

```
>     RadHostItem hostItem = new RadHostItem(rcddList);  %%//%% placeholder
>
>     hostItem.MinSize = new Size(150, 25);
>
>     hostItem.MaxSize = new Size(150, 25);
>
>
    rbgOtherParty.Items.Add(hostItem);  %%//%% add new control to the specif
ied RibbonGroup
>
> }
```

==== RadPanorama ====

http://docs.telerik.com/devtools/winforms/panorama/overview

RadPanorama is a control that displays elements of type RadTileElement in a mosaic manner. This control is inspired by the Metro Start Menu screen of Windows 8.

{{tdn:image123.gif?342x280}}

==== RadShortcut (Hotkeys) ====

http://docs.telerik.com/devtools/winforms/shortcuts/shortcuts-overview

Almost each application uses the so-called “Shortcuts” – a keyboard combination that triggers a specific action. For a valid shortcut is considered any keyboard combination where a Modifier Key (Ctrl, Alt and Shift or a combination of these) is down and other key(s) is pressed. This semantic is available out-of-the-box in our framework and allows you to seamlessly plug any valid keys combination as an action accelerator. Supported are also the so-called multiple keys shortcuts where the Keys member of each shortcut may be more than one key – e.g. Ctrl + A, S is recognized by the framework. Shortcuts without modifier keys are also supported, but they should be used with caution, since they may be in conflict with other controls which intercept keyboard input.

===== RadControls Misc =====

==== RadBreadCrumb – rbc ====

==== RadCarousel ====

http://www.telerik.com/help/winforms/carousel-overview.html\\
http:%%//%%www.telerik.com/support/code-library/winforms/carousel\\
Video:
http://www.telerik.com/videos/winforms/overview-of-radcarousel-for-winforms

==== RadClock – rclock ====

http://www.telerik.com/help/winforms/clock-overview.html

http://www.telerik.com/community/forums/winforms/calendar-and-datetimepicker.aspx\\
VIDEO:
http://tv.telerik.com/watch/winforms/getting-started-with-radtimepicker-for-winforms

//NOTE: The control cannot be resized since it uses a bitmap for its background.//

{{tdn:image124.gif?102x97}}

==== RadDesktopAlert – rda ====

http://www.telerik.com/help/winforms/desktopalert-overview.html

http://www.telerik.com/community/forums/winforms/desktop-alert.aspx\\
http:%%//%%www.telerik.com/help/winforms/desktopalert-getting-started.html\\
http://tv.telerik.com/watch/winforms/getting-started-with-raddesktopalert -Video- how to get started with RadDesktopAlert.\\
http://www.telerik.com/help/winforms/allmembers_t_telerik_wincontrols_ui_desktopalertmanager.html\\
VIDEO:
http://www.telerik.com/videos/winforms/getting-started-with-raddesktopalert

RadDesktopAlert component displays a small pop-up window on the screen to notify the user that a specific event has occurred in the application.

{{tdn:image125.png?294x96}}

My sample project:
[[file:///\\192.168.93.1\KB_MiM\Telerik\LS.TelerikSystrayApp|\\192.168.93.1\KB_MiM\Telerik\LS.TelerikSystrayApp]]

Popup notification similar to Outlook’s you have a new email Notify (Notification) popup. You can drag it around the screen. There are three buttons in the upper right to close, pin it to the screen and display a list of options.

For multi-monitor systems use the SetActiveScreen() method of the DesktopAlertManager class to set the screen where alerts display at ‘ScreenPosition’.

You can display multiple alert instances on the screen and the DesktopAlertManager will calculate the alerts’ location so that they do not overlap. Alerts are stacked on the screen in the order of their appearance. When an alert is closed, all related visible alerts are automatically relocated to utilize the screen real estate. To show multiple desktop alerts you must create a new instance for every alert. To replace an alert dispose the instance and create another new one.

```
int i ; %%//%% create multiple instances of the

for (i = 0; i <= 2; i++)

{

RadDesktopAlert a = new RadDesktopAlert();

a.CaptionText = i.ToString();

a.Show();

}

RadDesktopAllert does not support auto-size mode. You must set the size of
the control so your contents will fit.

Marcel tried this sample code in eTMsync but it didn’t work for him so he
ended up writing his own resize code.

http://www.telerik.com/forums/desktop-alert-not-resizing-accordingly-to-the-
contenttext-set

It offers some “HTML-like” formatting but does not support " ".
[[http://www.telerik.com/help/winforms/tpf-html-like-text-formatting.html|He
re is a list]] of supported tags and expressions.

[[http://www.telerik.com/forums/hyperlink-click-9782f6494b85|Here’s a code
example]] to make a clickable hyperlink in the Alert Content.

You can add a button (or other element) to any control, including an Alert.
Here’s how to add a button:\\
RadButtonElement element = new RadButtonElement();

element.Text = "Button";

element.MaxSize = new Size(20, 20);

this.radDesktopAlert1.Popup.AlertElement.ContentElement.Children.Add(element
);

[[http://www.telerik.com/forums/changing-desktopalert-color|Here’s Color
Change examples]] by choosing an alternate thems or using a customized theme
from Visual Style Builder.

//You can Preview the Alert in the IDE from the control Smart Tag in the
Component Bin.//

  * ‘ScreenPosition’ accepts six possible positions from the
AlertScreenPosition enumerator: TopLeft, TopCenter, TopRight, BottomLeft,
BottomCenter, BottomRight
```

```
  * ‘FixedSize’ defaults to 0,0 for a default size. Set this to override the
size, or do it in code:\\
this.radDesktopAlert1.FixedSize = new Size(200, 200);
  * ‘CanMove’ default True, enables the Alert to be dragged with the mouse
  * ‘AutoCloseDelay’ how long to stay on screen (defaults to 10 sec)
  * ‘AutoClose’ whether will disappear after the period of time defined by
the property. Set = false to leave open.
  * ‘AlertImage’ specifies an icon that will be shown on the left of the
alert panel
  * ‘ButtonItems’ collection adds custom buttons to the alert
  * ‘Opacity’ starting opacity when it appears (defaults to 0.8 – 80%).
Changes to 100% on mouse over.
  * ‘PopupAnimation’ = true to enable animations
    * ‘PopupAnimationFrames’ number frames in the popup animation. More
frames means longer animation.
    * ‘PopupAnimationDirection’ (default Down) direction of the animation.
Down slides down from the initial location. Can aldo have Up, Left or Right.
    * ‘PopupAnimationEasing’ different animation easing effects which are
used to start and finish the animation.
  * ‘FadeAnimationType‘ (default ‘FadeIn,FadeOut ‘)accepts values from
FadeAnimationType enum which also allows for combinations of its values.
  * You can also use the FadeAnimationFrames property to define the amount
of frames used to animate the opacity of the alert’s popup. By increasing
the amount of frames you will increase the duration of the fade animation.
  * Caption Buttons
    * ‘CloseButton’ closes the alert. Use ShowCloseButton to show/hide the
button.
    * ‘PinButton’ pin the alert on the screen and prevent it from being
closed automatically. ShowPinButton to show/hide the button.
    * ‘OptionsButton’ displays a drop-down menu with user-defined content
from the OptionItems collection. Use ShowOptionsButton to show/hide the
button.
  *

==== RadHScrollBar – rhsb ====

See: [[#radtoggleswitch-rts|RadVScrollBar]]

==== +-* RadLabel – rlbl ====

http://www.telerik.com/help/winforms/panels-and-labels-label-overview.html

http://www.telerik.com/community/forums/winforms/panels-and-labels.aspx

Telerik themeable label with
[[http://docs.telerik.com/devtools/winforms/telerik-presentation-framework/h
tml-like-text-formatting|HTML-like formatting]] capability. Also see
[[http://tv.telerik.com/watch/winforms/utility/enhanced-html-like-markup-sup
port|this video]].

{{tdn:image126.png?248x61}}
```

```
//NOTE: Our design policy it to only change the control name from default if
we will programmatically change the text.//

  * 'Text' \Behavior
  * 'Name' \Design
  * 'Location' \Layout
  * 'Size' \Layout
  * Multiline RadLabel - Set 'Auto Size' = False and ‘TextWrap’ = True. Then
go directly to the 'Size' and enter 50,50, or you can drag the box size.
Don't embed \n in the text. Or dock the label to a container to set the
size.

Custom Text Formatting via MarkupEditor

http://www.telerik.com/help/winforms/panels-and-labels-label-html-like-text-
formatting.html

  * For example making one word in a sentence bold. Click the down arrow on
the 'Text' value and click <Start MarkupEditor>.
  * Use <Shift><Enter> for new lines. It will generate HTML. The font
changes made here (color, etc) are not the same properties as those in the
Properties window.

=== Hyperlink to a function in the application ===

Set these RadLabel properties and double click the label to stub out a click
event.

  * ‘ForeColor’ = ‘HotTrack’ - a blue visible under the color dropdown
‘System’ tab
  * ‘Underline’ – ‘True’

=== Hyperlink to website in default browser ===

Instead of LinkLabel see the <Links> button on the ‘MarkupEditor’ behind the
‘[[#radlabel-rlbl|RadLabel]].Text’ property dropdown and open locations
without additional code.

  * Choose link ‘Type’ of ‘http:%%//%%’ or ‘https:%%//%%’ and enter a web
url there and the link works with no extra code. The .Text generated markup
looks like this:

> <html><a href="http:%%//%%www.legalitysoftware.com/">Legality
Software</a></html>
>
> <a href="http:%%//%%www.legalitysoftware.com/"><span style="color:
#d86040">Legality Software</span></a> - orange font in Wizards (NOTE: Set
‘BackColor’ = 225, 228, 240 to match wizard form)
>
> <html><a href="http:%%//%%support.lawtopia.net/">View log files</a></html>
```

```
  * Choose link ‘Type’ of ‘mailto:’ and enter an email address to open an
email in the default client.
  * Choose link ‘Type’ of ‘(other)’ and enter a local file path to pop the
Windows Explorer in the location.. The .Text generated markup looks like
this:

> <html><a href="file:%%//%%/C:/DevGit">View log files</a></html>

  * Make only part of the label a clickable link like this:

> <html>Click this <a href="file:%%//%%/C:/DevGit">View log files</a> to
open Windows Explorer</html>

Here’s the HTML for our company blurb for wizard pages, etc. (Set
‘TextAlignment’ = MiddleCenter):

<p><span style="font-size: 10pt; color: #00467f"><em>Legality Software
provides packaged software products and custom software solutions for law
firms. Our solutions typically enhance the capabilities of, or fix problems
with, the tools you use in your firm today. We also help firms automate
difficult, time consuming or repetitive tasks. Legality products and
solutions are designed to solve your problems so you can achieve your goals.
</em></span></p><p><span style="color: #00467f"><span style="font-size:
10pt"><em>If you have a specific problem we may already have a solution. <br
/></em></span><span style="font-size: 10pt"><em>If not we can build one for
you.</em></span></span></p><p><span style="font-size: 13px"><span
style="font-size: 10pt"><br /><span style="color: #00467f">Visit
</span></span><a href="http:%%//%%www.legalitysoftware.com/"><span
style="font-size: 10pt">Legality Software</span></a><span style="font-size:
10pt; color: #00467f"> to learn more.</span></span></p>

{{tdn:image127.png?478x164}}

### I DON'T KNOW HOW TO CENTER TEXT IN A MULTI-LINE LABEL

=== Dynamically Add Text to a Label ===

### grab code from TMFU

=== Dynamically Replace some Placeholder Text in a Label ===

### grab code from TMFU & ask Larry how he worked out the issue with the
Placeholder Text is not there for the 2<sup>nd</sup> replace

==== RadPrintDocument – rpd ====

==== RadProgressBar – rpb  ====

http://www.telerik.com/help/winforms/track-and-status-controls-progressbar-o
verview.html
```

```
http://www.telerik.com/community/forums/winforms/track-and-status.aspx

RadProgressBar is designed to display progress information to the user
during a long-running operation.

{{tdn:image128.gif?308x205}}

**### Need an example**

==== RadWaitingBar – rwb ====

http://www.telerik.com/help/winforms/track-and-status-controls-waitingbar-ov
erview.html

http://www.telerik.com/community/forums/winforms/track-and-status.aspx\\
http://docs.telerik.com/devtools/winforms/track-and-status-controls/waitingb
ar/waitingbar\\
http://docs.telerik.com/devtools/winforms/track-and-status-controls/waitingb
ar/properties-and-methods

Like a progress bar but with no start or end. It just shows activity.

^{{tdn:image129.gif?93x183}}^{{tdn:image130.gif?261x46}}^
|
|{{tdn:image131.gif?410x44}}|

Properties

  * 'WaitingStyle' – Indeterminate, **Throbber**, Dash
  * 'WaitingSpeed' – (90) animation speed
  * 'WaitingStep' – (1) number of pixels to move for each step
  * '.Text' – background text on the waiting bar – set 'ShowText' = true to
make it visible
  * 'StretchIndicatorsHorizontally' – inverses the color on the shaded
scrolling indicator for a nice effect
  *

> The 'WaitingBarIndicatorElement' is the sliding part, which has:

  * Its own '.Text' property making the text scroll with the colored
highlight.
  * 'ShouldPaint' which hides the slider (even though it still moves)
returning the waiting bar to a solid color – use this and the
StartWaiting()/StopWaiting() methods to indicate the process is finished.
//See note below about multiple elements…//
  *

> //NOTE: A RadWaitingBar may (does) have more than 1
'WaitingBarIndicatorElement', named 'WaitingBarIndicatorElement1' and
'WaitingBarIndicatorElement2' so if you use 'ShouldPaint', besure to change
both instances.//
```

```
Methods

  * StartWaiting() method starts the animation
  * StopWaiting() method stops the animation
  * ResetWaiting() method moves the waiting indicators to their initial
position.

**### Add an example here from HLS PdfParser**

==== RadRating ====

http://docs.telerik.com/devtools/winforms/track-and-status-controls/rating/r
ating

RadRating is a flexible UI component that allows users to place their rating
by selecting from a finite number of items (stars, diamonds and hearts). The
developers can fully customize the control by configuring its orientation,
rating precision, direction etc.

{{tdn:image132.png?277x119}}

==== RadRotator – rrot ====

http://docs.telerik.com/devtools/winforms/rotator/overview

RadRotator is a multipurpose component for content rotation and
personalization. Highly customizable, it delivers high interactivity and
user involvement. You can display a series of images, web URLs or any
collection of rad elements. Animation between frames can be opaque or
transparent and the movement can be in any direction. You can adjust the
level of granularity and interval between frames.

{{tdn:image133.png?340x287}}

==== RadTitleBar – rtb ====

http://www.telerik.com/help/winforms/forms-and-dialogs-radtitlebar-overview.
html

http://www.telerik.com/community/forums/winforms/forms-and-dialogs.aspx

RadTitleBar control is used in forms and provides functionality for
dragging, minimizing, maximizing and closing the form. This control is
internally used by RadForm.

To include a title bar with help, maximize, minimize and close button
functionality simply drag it from the toolbox on the desired form. This
control is great addition to the ShapedForm.

To customize the help, minimize, maximize and close buttons, use the
RadTitleBar.TitleBarElement's HelpButton, MinimizeButton, MaximizeButton and
```

CloseButton objects. Each button is a RadButtonElement type that include properties to control text, image, and layout.

{{tdn:image134.png?290x39}}

//Note: By default, the HelpButton is not shown. It is necessary to set its Visibility property to ElementVisibility.Visible in order to be displayed.//

==== RadVScrollBar – rvsb ====

http://www.telerik.com/help/winforms/track-and-status-controls-scrollbar-overview.html

http://www.telerik.com/community/forums/winforms/track-and-status.aspx

http://www.telerik.com/help/winforms/track-and-status-controls-scrollbar-getting-started.html

Using Telerik scrollbars is a bit more intricate compared to using the standard scrollbars because you have to handle scroll event manually. Add a RadPanel to your form then add a RadVScrollbar in the panel and dock it to the Right. Add another RadPanel inside the 1st RadPanel and set its height to the total height you want to be available upon scrolling (taller than the 1st RadPanel). This value can be statics e.g. 1000 pixels or dynamic determined by the scrollable content. Now add controls (the controls which are to be scrolled) to the second RadPanel.

{{tdn:image135.png?192x172}}

Then subscribe to the Scroll event of the vertical scrollbar and assign its negated value to the Top property of the second RadPanel:

```
void radVScrollBar1_Scroll(object sender, ScrollEventArgs e)

{

this.radPanel2.Top = -this.radVScrollBar1.Value;

}
```

Finally, set the Maximum property of the scrollbar to reflect the size of the scrollable height which is the total height of the scrollable content minus the visible height. For the example of this section in particular, that is the height of the second panel minus the height of the first panel.

```
this.radVScrollBar1.Maximum = this.radPanel2.Size.Height -
this.radPanel1.Size.Height;
```

==== * RadWizard – rw ====

http://www.telerik.com/help/winforms/wizard-overview.html

```
http://www.telerik.com/community/forums/winforms/wizard.aspx\\
Design Time Settings
http://www.telerik.com/help/winforms/wizard-design-time.html

Wizard dialog with multiple pages and <Back><Next> buttons.

{{tdn:image136.png?368x272}}

  * Could leave as default assigned name unless we will use more than one
Wizard in a project.
  * Drag this container to a regular form to create a Wizard.

Properties

  * ‘Mode’ - Wizard97 for the header panel we use, or Aero for a header with
nav buttons at the top

Quick Tasks Button

  * 'Edit Pages' - use this instead of 'Add Page'. It gives you more
control. You should add Welcome, Working (Wizard), and Completion pages.
    * This seems backwards but the 'Title' is the big text on top and the
'Header' is the smaller text under the Title.
  * Use the <Back>/<Next> buttons on the actual form to navigate in design
time.

> Another option is to click page name in 'Pages' Collection Editor to
jump/view.

### I’m not sure how to change the “Help” hyperlink in the lower left
corner.

Contains the following elements:

  * Pages collection – collection which contains Internal, Welcome and
Completion RadWizard pages.
  * Page header – element which is located above each page and contains
elements for title text, header text and page icon.
  * Command area – element which is located below each page and contains
command button – Back (Wizard 97), Next, Cancel, Finish, and Help.
  * Welcome image – element which contains the image of the Welcome page.
  * Completion image - element which contains the image of the Completion
page.
  * Top element – element which contains the Back button of Wizard Aero
view.

Events - help you to follow the state of the control at run time, implement
custom pages sequence and page processing validation.\\
http://www.telerik.com/help/winforms/wizard-events.html

  * ModeChanging - Fires before the mode of RadWizard is changed. It is
```

```
cancelable event. The arguments of the event provide the current and the
next WizardMode.
  * ModeChanged - Fires after the mode of RadWizard is changed. The
arguments of the event provide the previous and the current WizardMode.
  * Next - Fires when the Next command button is clicked. It is cancelable
event.

>
[[http://www.telerik.com/forums/how-to-add-custom-functionality-to-next-butt
on-in-a-wizard|Add custom/Validation functionality to Wizard Next Button]]
>
> Subscribe to the Next event
>
> this.radWizard1.Next += new WizardCancelEventHandler(radWizard1_Next);
>
> Handle the Next event
>
> private void radWizard1_Next(object sender, WizardCancelEventArgs e)
>
> {
>
> if (this.radWizard1.SelectedPage == this.radWizard1.Pages[1]) %%//%% Check
if this is the desired page
>
> {
>
> %%//%% Do some validation here
>
> e.Cancel = true; %%//%% If the validation does not pass
>
> this.radWizard1.SelectedPage = this.radWizard1.Pages[0]; %%//%% Could
return to 1<sup>st</sup> page
>
> %%//%% Or you could show some message to the user
>
> }
>
> }

  * Previous - Fires when the Back command button is clicked. It is
cancelable event.
  * SelectedPageChanging - Fires before the selected page of RadWizard is
changed. It is cancelable event. The arguments of the event provide the
selected page of the control and the page to be selected.
  * SelectedPageChanged - Fires after the selected page of RadWizard is
changed. The arguments of the event provide the previous selected page and
the selected page of the control.
  * Finish - Fires when the Finish command button is clicked.
  * Cancel - Fires when the Cancel command button is clicked.
  * Help - Fires when the Help command button is clicked.
```

```
[[http://www.telerik.com/forums/branching|Custom Page Branching Sample
Project]]

[[http://www.telerik.com/forums/navigation-buttons|Custom CommandArea
Controls like Navigation Buttons]]

Create multiple Welcome or Completion pages

  * ‘WelcomePage’ - Select one of multiple Welcome pages
  * ‘WelcomePage.Name’ – Provide a rwppagenameWelcomePage style name from
one of multiple Welcome pages
  * ‘CompletionPage’ - Select one of multiple Completionpages
  * ‘CompletionPage.Name’ – Provide a rwppagenameCompletionPage style name
from one of multiple Completionpages
  * Edit one of the alternate pages by selecting it from the Quick Task
‘Edit Pages’ list or ‘Pages’ collection, then click <Cancel> (believe it or
not). If you click <OK> it will jump to a different page.

RadWizardPage - rwp

> Only change from default if we will programatically change the text)

====== Telerik – Calendar, Time and Scheduling ======

==== Related Controls ====

  * [[#raddatetimepicker-rdtp|RadDateTimePicker]]
  * [[#radtimepicker-rtp|RadTimePicker]]
  * [[#radclock-rclock|RadClock]]

==== RadCalendar – rc  ====

http://docs.telerik.com/devtools/winforms/calendar/calendar\\
http://www.telerik.com/community/forums/winforms/calendar-and-datetimepicker
.aspx\\
VIDEO:
http://tv.telerik.com/watch/winforms/radcalendar/using-radcalendar-for-winfo
rms

RadCalendar supports all common features you would find in the standard
Month Calendar control, such as first day of week, show today, special days,
and show/hide week numbers, but adds several nifty features you will not
find in the Visual Studio toolbox, such as focused date, show/hide week
days, fish eye (date zoom functionality), read-only support, and show other
month days. We went even further and extended the calendar functionality
with footer support, controllable header/footer, fast navigation buttons,
and full control over cell formatting (paddings, borders, margins, spacings,
alignment, etc), making RadCalendar a feature-complete control with expected
behavior, which your users will truly appreciate.

{{tdn:image137.png?351x241}}
```

```
This is a big calendar like you would see in Outlook.

It’s not a dropdown to select a date. For that see [[#raddatetimepicker-
rdtp|RadDateTimePicker]].

Code snippet to set the calendar control double arrow navigation to a year
jump from the default 3 month jump. Put this code in the form load event.

> (radCalendarControlName.DateTimePickerElement.GetCurrentBehavior() as
RadDateTimePickerCalendar).Calendar.FastNavigationStep = 12;

==== RadScheduler – rs ====

http://www.telerik.com/help/winforms/scheduler-introduction.html

http://www.telerik.com/community/forums/winforms/scheduler.aspx\\
VIDEO:
http://www.telerik.com/videos/winforms/introduction-to-radscheduler-for-winf
orms-webinar\\
VIDEO: http://tv.telerik.com/watch/winforms/radscheduler/scheduler

RadScheduler is a highly-customizable appointment presentation component
that offers rich Outlook®-style functionality. When used with RadReminder
and RadDesktopAlert you can add rich scheduling UI to any WinForms
application and enjoy a lightweight yet highly customizable component.

{{tdn:image138.png?445x302}}

=== RadSchedulerNavigator – rsn  ===

http://www.telerik.com/help/winforms/scheduler-scheduler-navigator-overview.
html

RadSchedulerNavigator is a stand-alone control used for navigation in
RadScheduler control.

{{tdn:image139.png?483x73}}

=== SchedulerBinderDataSource – sbds  ===

http:%%//%%docs.telerik.com/devtools/winforms/scheduler/design-time/smart-ta
g

=== RadReminder – rr ===

http://docs.telerik.com/devtools/winforms/scheduler/reminders/radreminder
http://www.telerik.com/community/forums/winforms/scheduler.aspx\\
VIDEO:
http://tv.telerik.com/watch/winforms/getting-started-with-radschedulerremind
er
```

RadReminder is a component that reminds you of an object that you pass to it. This object should implement IRemindObject and depending on the values that you set in the implementation of the IRemindObject interface, RadReminder throws an event. When the event is fired, you can show an appropriate message to the user using RadDesktopAlert or another alert implementation of your choice.

{{tdn:image140.png?320x241}}

=== RadSchedulerReminder – rsr ===

http://www.telerik.com/help/winforms/scheduler-reminders-radschedulerreminder.html

RadSchedulerReminder represents a special reminder object for the appointments that are collected in RadScheduler. This component inherits from RadReminder.

{{tdn:image141.png?406x302}}

==== RadGanttView - rganttv ====

http://docs.telerik.com/devtools/winforms/ganttview/ganttview-\\
http://docs.telerik.com/devtools/winforms/ganttview/timeline/timeline-views

RadGanttView is a data-visualization and editing control for project planning data and different types of task and time scheduling. All tasks are represented as vertical bars aligned along a timeline with the beginning and end time of each task determining its location along the timeline. These elements and the dependencies between them comprise the work breakdown structure of a project.

{{tdn:image142.png?396x259}}

RadGanttView offers a number of built-in timeline views which allow you to show the timeline in different scales. It is reasonable to change the timeline view when the user zooms in or out. This will give the users a precise measurement of time in every zoom level.

====== Telerik - Document Processing ======

http://docs.telerik.com/devtools/document-processing/introduction\\
Telerik Document Processing is a bundle of UI-independent, cross-platform libraries enabling you to process the most commonly used flow, fixed and spreadsheet document formats. The components allow to create documents, import, modify and export them without external dependencies between the following file formats: CSV, DOCX, HTML, PDF, RTF, TXT, XLSX, ZIP

===== RadPdfProcessing =====

http://docs.telerik.com/devtools/document-processing/libraries/radpdfprocess

```
ing/overview

RadPdfProcessing is a processing library that allows to create, import and
export PDF documents.

Some of the features you can take advantage of are:

  * Import/export to PDF format.
  * Support for images and shapes.
  * Easy-to-use API for document generation.

===== RadSpreadProcessing =====

http://docs.telerik.com/devtools/document-processing/libraries/radspreadproc
essing/overview

RadSpreadProcessing is a document processing library that enables your
applications to easily import and export files to and from the most common
spreadsheet file formats.

{{tdn:image143.png?344x219}}

Some of the features which you can use are:

  * Import/export to different spreadsheet formats. Check the Formats and
Conversion section for more information on this topic.
  * More than 200 built-in formulas.
  * Powerful and easy-to-use API.
  * Data Validation
  * Filtering
  * Grouping
  * Sorting

===== RadSpreadStreamProcessing =====

http://docs.telerik.com/devtools/document-processing/libraries/radspreadstre
amprocessing/overview

Spread streaming is a document processing paradigm that allows you to create
big spreadsheet documents with great performance and minimal memory
footprint. The key for the memory efficiency is that the spread streaming
library writes the spreadsheet content directly to a stream without creating
and preserving the spreadsheet document model in memory. Each time an
exporter object is disposed, the set values are written into the stream.
This allows you to create large documents with an excellent performance.

Some of the features you can take advantage of are:

  * Export to XLSX or CSV files
  * Writing directly into a stream
  * Append new worksheets to existing ones
```

```
  * Grouping
  * Styling and formatting cells
  * Hidden rows and columns
  * Freezing panes

**RadSpreadStreamProcessing vs. RadSpreadProcessing**

There are two main differences between the libraries.

  * RadSpreadStreamProcessing can be used only to create documents and
append data to existing ones. On the other hand you can use the
RadSpreadProcessing also for reading and modifying the content of documents.
  * RadSpreadStreamProcessing writes directly into a stream, unlike
RadSpreadProcessing which creates models for the elements in the document.
This is why the memory used with the spread streaming library is
significantly lower than when using RadSpreadProcessing.

===== RadWordsProcessing =====

http://docs.telerik.com/devtools/document-processing/libraries/radwordsproce
ssing/overview

RadWordsProcessing is a processing library that allows to create, load,
modify and export documents to a variety of formats.

{{tdn:image144.png?339x315}}

Some of the features are:

  * Import/export to several file formats.
  * Rich document model support, including tables, images, hyperlinks,
headers and footers.
  * Comprehensive and easy-to-use API allowing for generation and
manipulation of documents.
  * Variety of built-in themes and styles.

===== RadZipLibrary =====

http://docs.telerik.com/devtools/document-processing/libraries/radziplibrary
/overview

With RadZipLibrary you can compress data like images, docx or pdf files.

{{tdn:image145.png?451x245}}

This is a list with short descriptions of the top-of-the-line features of
Telerik's Zip Library control:

  * Easy to use API: The Zip Library exposes flexible and easy API to
provide you with full control over the compressed data.
  * Load or create ZIP files: You can load data from existing ZIP files,
```

```
create new ones and edit ZIPs that can be used by other applications. You
can also create ZIP files in memory or add data to ZIP file from stream.
  * Store your data in the Isolated Storage: You can store the compressed
data in the Isolated Storage.
  * Support for large files: The Zip Library works seamlessly with large
files.
  * Support for encryption: You can protect your ZIP file with password for
more security.

===== RadPdfViewer – rpdfv =====

http://www.telerik.com/help/winforms/pdfviewer-overview.html\\
Video:
http://www.telerik.com/videos/winforms/getting-started-with-the-winforms-rad
pdfviewer

RadPdfViewer is a control that can natively visualize PDF documents straight
in your application. It comes with a predefined UI that is intuitive and
provides the means for utilizing the features of the control. The control
utilizes virtualization and supports load on-demand mode in order to
guarantee good performance with larger documents.

{{tdn:image146.png?418x274}}

Here is a list of the supported features:

  * Open PDF document from file or stream
  * Pan
  * Zoom
  * Selection
  * Annotations
  * Copy to clipboard
  * Find
  * Keyboard navigation
  * Navigation toolbar
  * Context menu
  * Printing

**Unsupported Features**
http://www.telerik.com/help/winforms/pdfviewer-unsupported-features.html

  * It can’t open encrypted or password protected PDFs
  * Annotations - of all annotations, only link is supported
  * Structured content - there are several ways to define structured content
in a PDF file. RadPdfViewer does not support document outline, article
threads and interactive form
  * Some other things as well

==== RadPdfViewerNavigator – rpdfvn  ====

http://docs.telerik.com/devtools/winforms/pdfviewer/pdfviewernavigator
```

```
Adds toolbar to RadPdfViewer that adds support for the typical commands you
want to use on a PDF file.

{{tdn:image147.png?720x36}}

This control can be used with RadPdfViewer. It provides predefined UI for
the most common operations used with PDF files. The following list contains
all features:

  * Open File: You can open files with this button.
  * Print: Use this button to print the content.
  * SaveAs: Use this option to save the current file in a specific location.
  * Rotate: Rotate all pages clockwise or counter clockwise using the
buttons.
  * Navigation: You can navigate trough the document by using the
previous/next page buttons or manually typing the page number.
  * Pan and Select: You can select text in the document or you can move the
page with the mouse.
  * Zoom: You can use the buttons to increase/decrease the zoom factor or
select it from the drop down.
  * FitWith and FithPage: These buttons allows you to fit the page according
to the current window size.
  * Search: You can search for a particular string and navigate among the
results.

===== RadDiagram =====

http://docs.telerik.com/devtools/winforms/diagram/diagram

{{tdn:image148.png?233x207}}RadDiagram offers flexible and interactive
diagramming layouts for your rich data-visualization applications.

{{tdn:image149.png?308x213}}

====== Telerik – Themes ======

http://www.telerik.com/help/winforms/themes-using-a-default-theme-for-the-en
tire-application.html Definitely this one\\
http://www.telerik.com/support/kb/winforms/details/creating-a-theme-componen
t Maybe this one??\\
http://www.telerik.com/community/forums/winforms/themes-and-visual-style-bui
lder.aspx\\
Video:
[[http://www.telerik.com/videos/winforms/theme-for-telerik-ui-for-winforms|T
heme for Telerik UI for Winforms]]\\
Video:
http://www.telerik.com/videos/winforms/changing-themes-at-run-time-with-radc
ontrols-for-winforms

Use the Telerik ThemeViewer Tool to examine all the controls in the
available themes. **This is also a good place to see all the controls in
```

```
action.**

### At some point I'd like to have a standard "Tools | Options" dialog in
our project templates where that's appropriate. This would contain a tree on
the left and a property grid on the right. I'd like a search field above
that tree so you can type word(s) to filter and only see branches of the
tree that contain that text. This would enable the user to easily find
settings.

==== ucTelerikThemeSelector in LS.Common.Presentation ====

See this video:
[[file:///\\192.168.93.1\KB_MiM\Telerik\20160110)%20ucTelerikThemeSelector.m
p4|\\192.168.93.1\KB_MiM\Telerik\20160110) ucTelerikThemeSelector.mp4]]

### I need to recreate this video (shorter) with the direct procedure

  - Add the ucTelerikThemeSelector to a solution by adding the DLL
'C:\DevGIT\LS.Common\LS.Common.Presentation\bin\x86\Debug\LS.Common.Presenta
tion.dll' to the VS toolbox.
  - Drop the ucTelerikThemeSelector on the (Options) form.
  - Search the VS Toolbox for 'theme' and drag these themes (any you want to
make available) to the rfrmMain:

  * <del>Aqua</del> – dark grey base with sliver control faces and bright
blue selection highlights. Has rounded control corners.
  * <del>Breeze</del> – grey base with bright blue control faces and white
selection highlights. Hard to read white text on grey in some controls.
  * 'Desert' (Desert) – gunmetal grey base with matching, slightly darker
control faces and white selection highlights (nice *)
  * <del>HighContrastBlack</del> – black base, black control faces, white
text with purple and green selection highlights. Not a fan.
  * 'Office 2007 Black' (Office2007Black) – dark grey base, fairly dark
foreground (nice *)
  * (Office2007Silver) – silver base
  * 'Office 2010 Black' (Office2010Black) – same dark grey base as 2007 with
lighter foreground (nice *)
  * 'Office 2010 Blue' (Office2010Blue) – light blue base, blue control
faces and light foreground
  * 'Office 2010 Silver' (Office2010Silver) – similar to 2007 with a
slightly lighter foreground
  * 'Office 2013 Light' (Office2013Light) – flat controls with nearly white
base and light blue selection highlights
  * 'Office 2013 Dark' (Office2013Dark) – flat controls with light grey base
and light blue selection highlights (not very dark)
  * 'Metro' (TelerikMetro) – flat controls with a green base
  * 'Metro Blue' (TelerikMetroBlue) – flat controls with a blue base (nice
*)
  * <del>'Metro Touch' (TelerikMetroTouch) – same as Metro (flat controls
with a green base) but with larger controls for finger navigation. NOTE:
requires app restart to leave.</del>
```

```
  * ‘Visual Studio 2012 Light’ (VisualStudio2012Light) – flat controls with
light grey base, med grey control faces, med and light blue selection
highlights
  * (VisualStudio2012Dark) – flat controls with dark grey base, med grey
control faces, darker grey foreground, med blue selection highlights
  * ‘Windows 7’ (Windows7) - light blue base, (classic) light grey control
faces and light foreground (nice *)
  * ‘Windows 8’ (Windows8) – flat controls with a blue base and light grey
control faces (nice *)

<HTML><ol start="4" style="list-style-type: decimal;"></HTML>
<HTML><li></HTML><HTML><p></HTML>Enter the desired app default theme (as
‘spelled’ above but with no spaces) in the ‘ThemeName’ property and the form
will change in the IDE and when the app is run. CONFIRM THIS: //Note you can
manually rename each dropped theme control but you can’t use spaces.//\\
NOTE: I currently have the default themes set in the ucTelerikThemeSelector
code behind to
‘office2013DarkTheme’.<HTML></p></HTML><HTML></li></HTML><HTML></ol></HTML>

### This works BUT the form does not use the specified theme on app launch.
The theme only changes when selecting a new entry while running. Need to fix
that.

<HTML><ol start="5" style="list-style-type: decimal;"></HTML>
<HTML><li></HTML><HTML><p></HTML>Add this code fragment after
InitializeComponent(); to set the app to the default theme on launch.\\
%%//%% Set the theme to the current ucTelerikThemeSelector default\\
ThemeName = ucTelerikThemeSelector1.ThemeName; %%//%% Set the theme for this
(existing) form\\
ThemeResolutionService.ApplicationThemeName =
ucTelerikThemeSelector1.ThemeName; %%//%% Set the theme for all other
(future created) forms in the
application<HTML></p></HTML><HTML></li></HTML><HTML></ol></HTML>

==== Code to Set the Current Theme %%**%% ====

ThemeResolutionService.ApplicationThemeName = "Office2013Dark"; %%//%% Set
the Current Theme Applicaton wide

%%//%% NOTE: You must also load the Office2013Dark theme in the solution and
add ‘’

ThemeName = ucTelerikThemeSelector1.ThemeName; %%//%% Set the theme for this
(existing) form. This is good for rfrmMain where it is already created
before the ApplicationThemeName was set.

Or load the Office2013Dark theme as the default/only theme for a simple
application (with no ThemeSelector)

Add this code fragment after InitializeComponent();
```

```
ThemeResolutionService.ApplicationThemeName = "Office2013Dark"; %%//%% Set
the Current Theme Applicaton wide

ThemeName = "Office2013Dark"; %%//%% Set the theme for this (existing) form.

%%//%% NOTE: You must also add the Office2013Dark theme control by dragging
it from the Toolbox to the form

//Note: You still must to drop the theme component on the form (e.g.
Office2007Black in the code samples above) or create an instance of the
desired theme programatically.//

==== Enable/Disable the globally set theme for a specific control ====

  * ‘EnableApplicationThemeName’ – Set a control’s
‘EnableApplicationThemeName’ property to a different theme than the assigned
application wide theme.
    * If true the control will use the globally set ApplicationThemeName
property.
    * If false this control’s theme is set by its own ThemeName property.

> ThemeResolutionService.ApplicationThemeName = "Office2010Blue"; %%//%%
Application wide theme
>
> radGridView1.ElementTree.EnableApplicationThemeName = false; %%//%% set
GridView to use separate theme
>
> radGridView1.ThemeName = "Office2010Silver"; %%//%% set GridView alternate
theme

### NOTES:

  - I want to adjust the default theme list (above) in the actual user
control.
  - I’d like to see if the ucTelerikThemeSelector coud be coded to search
the solution for included themes and populate it’s list from that so we
don’t have to manually match them up.
  - I’d also like to be able to select the default theme as a property
directly in the IDE, rather than typing in the correct theme name.

### We need to upgrade our Telerik to the latest version and see what new
themes we want to add to our supported list. Also which do we currently want
to use as our default?

====== Telerik CAB Enabling Kit ======

http://docs.telerik.com/devtools/winforms/telerik-cab-enabling-kit/telerik-c
ab-enabling-kit\\
VIDEO:
http://tv.telerik.com/watch/winforms/cab/intro-telerik-cab-enabling-kit\\
//NOTE: There are multiple other videos on the main page.//
```

The Telerik CAB Enabling Kit for WinForms is part of the Telerik UI for for WinForms suite since the R1 2008 release. It provides seamless integration of Telerik UI for for WinForms with the Composite UI Application Block (CAB). It is well suited for enterprise applications, and helps developers with using the best practices and patterns.

The Telerik CAB Enabling Kit (TCEK) for WinForms can be used to develop smart-client line-of-business applications such as OLTP (online transaction processing) front ends for data entry applications; rich client portals to back-end services like government or bank teller applications; UI-intensive stand-alone applications such as those used by call center staff, IT support desks or stock traders.

Composite UI Application Block Framework

The Composite UI Application Block (CAB) framework is intended for large WinForms-based applications. The three major goals of CAB design are:

  * Modularity - you can implement business logic, visual SmartParts, infrastructure components, presenter or controller components, and any other objects the application requires, in separate modules.
  * Productivity - Developer productivity increases because each developer (or team) is able to concentrate on a specific task. The use of consistent user interfaces increases end-user productivity in applications built using Composite UI Application Block, reducing the end-user training requirements when releasing a new application.
  * Extensibility - the design of the Composite UI Application Block promotes extensibility in many different ways

====== Telerik – Coded UI ======

http://docs.telerik.com/devtools/winforms/codedui/codedui\\
VIDEO:
http://tv.telerik.com/watch/radcontrols-for-winforms/getting-started-with-coded-ui-for-winforms\\
VIDEO:
http://www.telerik.com/videos/winforms/what-is-new-in-q3-2012-radcontrols-for-winforms

The CodedUI Extension for Telerik UI for for WinForms controls is an extension for Microsoft Visual Studio, which runs in the Visual Studio Coded UI Test process and captures information about the RadControls that it encounters during a test recording and then generates code to replay that test session.

CodedUI tests can test your application through the user interface (UI). Coded UI Tests are particularly useful where there is validation of the control properties or other logic in the user interface. If the control property value is invalid, the test fails.

Creating a coded UI test is easy. You simply perform the test manually while

```
the CUIT Test Builder runs in the background. You can also specify what
values should appear in specific control properties. The CUIT Test Builder
records your actions and generates code from them. You can add an assertion
during the recording. After the test is created, you can edit it in a Visual
Studio editor and modify the sequence of the actions.

====== Edit UI Elements ======

===== Hide Control Borders =====

Options to hide control Borders (using RadPanel as an example, but this
works on RadTextbox, etc):

  * Click SmartTag and set “Edit UI Elements | BorderPrimative |
ShouldPaint” to ‘False’, or use this code:

this.rpnlPanel.PanelElement.PanelBorder.ShouldPaint = false;

  * Or to hide some/selected borders using this code:

%%//%% Hide some/selected RadPanel borders\\
this.rpnlPanel.PanelElement.PanelBorder.BoxStyle =
BorderBoxStyle.FourBorders; %%//%% enable border segments

> this.rpnlPanel.PanelElement.PanelBorder.TopWidth = 0; %%//%% hide the top
border
>
> this.rpnlPanel.PanelElement.PanelBorder.BottomWidth = 0;
>
> this.rpnlPanel.PanelElement.PanelBorder.LeftWidth = 0;
>
> this.rpnlPanel.PanelElement.PanelBorder.RightWidth = 0;\\
> %%//%% From:
http://www.telerik.com/forums/can-radpanel-only-have-bottom-border-or-top-et
c

===== HTML-like Text Formatting =====

http://docs.telerik.com/devtools/winforms/telerik-presentation-framework/htm
l-like-text-formatting

VIDEO: Enhanced HTML-like Markup Support

Telerik UI for for WinForms provide an advanced text styling mechanism which
can be applied to all Telerik WinForms controls and their elements, because
it enhances one of the smallest element in Telerik Presentation Framework -
the text primitive. The new rich text formatting mechanism uses plain HTML
tags to display formatted text such as font style, font color, font size,
etc. Your text must start with the <html> tag so that HTML-like formatting
is activated. The list of supported markup tags is given below:
```

```
**Supported Tags**

^Tag          ^End Tag        ^Description
^
|<font>       |N/A            |Font Family. Please use the span tag since
the font tag is not supported by RadMarkupEditor.
|
|<color>      |N/A            |Text color. Please use the span tag since the
color tag is not supported by RadMarkupEditor.
|
|<size>       |N/A            |Font size. Please use the span tag since the
size tag is not supported by RadMarkupEditor.
|
|<b>, <strong>|</b>, </strong>|Bold
|
|<i>, <em>    |</i>, </em>    |Italic
|
|<u>          |</u>           |Underlined text
|
|<br>         |N/A            |Line break
|
|<p>          |</p>           |Paragraph
|
|<span>       |</span>        |Span. There is limited support of the style
attribute and the CSS properties: font-family, font-size, color, and
background-color. Refer to the example below. The Span tag is preferable to
font, color, and size tags.|
|<ol>         |</ol>          |Ordered list
|
|<ul>         |</ul>          |Unordered list
|
|<li>         |</li>          |List item. Defines a list item in an ordered
or unordered list.
|
|<strike>     |</strike>      |Striked text.
|
|<a>          |</a>           |Link
|
|<img>        |N/A            |Image
|

===== Next… =====

====== User Controls ======

Add a ‘New Project’ to the solution. Choose “Visual C# | Windows” in the
left panel. Then choose “Windows Forms Control Library” from the middle
panel.

Name it ‘ucXxxxXxxx’ and click <OK>
```

```
===== Setup Properties Configurable in the VS IDE =====

Here is an overview video:\\
[[file:///\\192.168.93.1\KB_MiM\Visual%20Studio%202012\20151227)%20uc%20Prop
erties%20set-able%20in%20the%20IDE.mp4|\\192.168.93.1\KB_MiM\Visual Studio
2012\20151227) uc Properties set-able in the IDE.mp4]]

Each settings name should be formatted as below then typically defaulted to
True:

  * ControlNameVisible
  * ControlNameEnabled

Here are code samples to setup properties for control from the IDE

[Category("Legality Software")] %%//%% Category creates a group in the VS
IDE Property Grid

public bool RefreshVisible

{

get { return rbtnRefresh.Visible; }

set { rbtnRefresh.Visible = value; } %%//%% Show or hide the button

}

[Category("Legality Software")]

public bool RefreshEnabled

{

get { return rbtnRefresh.Enabled; }

set { rbtnRefresh.Enabled = value; } %%//%% Enable or disable the button

}

[Category("Legality Software")]

public bool RecordCountVisible

{

get { return rlblRecordCount.Visible; }

set { rlblRecordCount.Visible = value; } %%//%% Show or hide the label

}
```

```
[Category("Legality Software")]

public bool FilterVisible

{

get { return tlpFilter.Visible; }

set { tlpFilter.Visible = value; } %%//%% Show or hide the TableLayoutPanel
(subpanel in a master tlp)

}

[Category("Legality Software")]

public bool SearchVisible

{

get { return tlpSearch.ColumnStyles[0].Width == 50; } %%//%%
TableLayoutPanel 1<sup>st</sup> Column default width is 50

> %%//%% ### I’m not sure why we don’t have to support getting the
ColumnStyles[1].Width of 27?

set

{

if (value == true)

{

tlpSearch.ColumnStyles[0].Width = 50; %%//%% Show column by restoring to
default width of 50

tlpSearch.ColumnStyles[1].Width = 27; %%//%% Show column setting width to 27
(actually a %)

}

else

{

tlpSearch.ColumnStyles[0].Width = 0; %%//%% hide column by setting width to
0

tlpSearch.ColumnStyles[1].Width = 0;

}
```

```
}

}

[Category("Legality Software")]

public bool QueryListTreeSearchRowVisible

{

get { return tlpQueryListTree.RowStyles[0].Height == 26; } %%//%%
TableLayoutPanel 1<sup>st</sup> Row default height is 26

set

{

if (value == true)

{

tlpQueryListTree.RowStyles[0].Height = 26; %%//%% Show row by restoring to
default height of 26

}

else

{

tlpQueryListTree.RowStyles[0].Height = 0; %%//%% hide row by setting height
to 0

}

}

}

[Category("Legality Software")]

public string OptionAllText

{

get { return rrdoAll.Text; }

set { rrdoAll.Text = value; } %%//%% Set the radio button text

}
```

```
[Category("Legality Software")]

public Color OptionAllForeColor

{

get { return rrdoAll.ForeColor; }

set { rrdoAll.ForeColor = value; } %%//%% Set the radio button font color

}

===== Legality User Controls =====

Video:
[[file:///\\192.168.93.1\KB_MiM\Telerik\20151221)%20Adding%20LS%20Common%20U
ser%20Controls%20to%20a%20project.mp4|\\192.168.93.1\KB_MiM\Telerik\20151221
) Adding LS Common User Controls to a project.mp4]]

Start watching at 3:20 for the ‘right click Toolbox| Choose Items’ option.
Look for the Project DLL for example:

//‘C:\DevGIT\ls.common\LS.Common.SqlEditor\bin\x86\Debug\LS.Common.SqlEditor
.dll’// contains all the versions of the SQL Editor User Control.

==== DLL Folders to add Items to ToolBox ====

...\DevGIT\LS.Common\LS.Common.Presentation\bin\Debug\LS.Common.Presentation
.dll\\
...\DevGIT\LS.Common\LS.Common.FTP\bin\Debug\LS.Common.FTP.dll\\
...\DevGIT\ls.common\LS.Common.Grid\bin\Debug\LS.Common.Grid.dll\\
...\DevGIT\ls.common\LS.Common.ItemList\bin\Debug\LS.Common.ItemList.dll\\
...\DevGIT\LS.Common\LS.Common.Log.Presentation\bin\Debug\LS.Common.Log.dll\
\
...\DevGIT\LS.Common\LS.Common.Scheduler\bin\Debug\LS.Common.Scheduler.dll\\
...\DevGIT\LS.Common\LS.Common.SqlEditor\bin\Debug\LS.Common.SqlEditor.dll\\
...\DevGIT\LS.Common\LS.Common.TableViewer\bin\Debug\LS.Common.TableViewer.d
ll\\
...\DevGIT\LS.Common\LS.Common.Tree\bin\Debug\LS.Common.Tree.dll\\
...\DevGIT\LS.Common\LS.Common.TMDBConnector\bin\Debug\LS.Common.TMDBConnect
or.dll

Current list of folders:

  * C:\DevGIT\LS.Common\LS.Common.Grid\ucGridStatus.Designer.cs
  * C:\DevGIT\LS.Common\LS.Common.Grid\ucGridStatus2.Designer.cs
  * C:\DevGIT\LS.Common\LS.Common.Log.Presentation\ucLogData.Designer.cs
  * C:\DevGIT\LS.Common\LS.Common.Log.Presentation\ucLogViewer.Designer.cs
  * C:\DevGIT\LS.Common\LS.Common.Presentation\ucEmailConnector.Designer.cs
  *
C:\DevGIT\LS.Common\LS.Common.Presentation\ucFolderPathUserSecurity.Designer
```

```
.cs
  * C:\DevGIT\LS.Common\LS.Common.Presentation\ucGridFilter.designer.cs
  * C:\DevGIT\LS.Common\LS.Common.Presentation\ucPassword.Designer.cs
  *
C:\DevGIT\LS.Common\LS.Common.Presentation\ucTelerikThemeSelector.Designer.c
s
  *
C:\DevGIT\LS.Common\LS.Common.Presentation\ucUsernamePassword.Designer.cs
  * C:\DevGIT\LS.Common\LS.Common.Scheduler\ucScheduler.designer.cs
  * C:\DevGIT\LS.Common\LS.Common.SqlEditor\ucSqlEditor.Designer.cs
  * C:\DevGIT\LS.Common\LS.Common.SqlEditor\ucSqlEditor2.Designer.cs
  * C:\DevGIT\LS.Common\LS.Common.TableViewer\ucTableViewer.Designer.cs
  *
C:\DevGIT\LS.Common\LS.Common.TMDBConnector\ucTMDatabaseConnector.Designer.c
s
  * C:\DevGIT\LS.Datatool\LS.Components\ucQueryConfig.Designer.cs
  * C:\DevGIT\LS.Datatool\LS.Components\ucSqlDatabaseConnector.Designer.cs
  * C:\DevGIT\LS.Datatool\LS.Components\ucTelerikThemeSelector.designer.cs
  * C:\DevGIT\LS.Datatool\LS.Components\ucTMDatabaseConnector.Designer.cs
  *
C:\DevGIT\LS.Datatool\LS.Components\ucTMDatabaseConnector_Old.Designer.cs
  *
C:\DevGIT\LS.Common\LS.Common.Presentation\Scheduler\ucScheduler.designer.cs
  *
C:\DevGIT\LS.Common\LS.Common.Presentation\TMDBConnector\ucTMDatabaseConnect
or.Designer.cs
  * C:\DevGIT\LS.Common\LS.Common.FTP\ucFtpConnector.Designer.cs
  * C:\DevGIT\LS.Common\LS.Common.Grid\ucGridSmart.Designer.cs

==== ucFilePath.cs ====

...\DevGIT\LS.Common\LS.Common.Presentation\bin\Debug\LS.Common.Presentation
.dll

### Make the button here and on ucFolderPath the same width. Also shouldn’t
the file control NullText say something about a ‘file’?

{{tdn:image150.png?338x40}}

==== ucFilter.cs ====

//Look for [[#ucgridfilter.cs|ucGridFilter.cs]]//

==== ucFolderPath.cs ====

...\DevGIT\LS.Common\LS.Common.Presentation\bin\Debug\LS.Common.Presentation
.dll

{{tdn:image151.png?336x39}}

==== ucFolderPathUserSecurity.cs ====
```

```
...\DevGIT\LS.Common\LS.Common.Presentation\bin\Debug\LS.Common.Presentation
.dll

{{tdn:image152.png?267x136}}

==== ucFtpConnector.cs ====

...\DevGIT\LS.Common\LS.Common.FTP\bin\Debug\LS.Common.FTP.dll

Everything you need to configure settings for a FTP/SFTP connection and
transfer files.

{{tdn:image153.png?326x154}}

==== ucGridFilter.cs ====

...\DevGIT\LS.Common\LS.Common.Presentation\bin\Debug\LS.Common.Presentation
.dll

### NEED TO MOVE TO GRID PROJECT

//See video above on how to add, but in short look for it in the Toolbox.//

Includes a ‘Type to Filter/Search” text box and a <X> button to clear it.
Filters the contents of a grid on any text column value. Filtering starts
after the 3<sup>rd</sup> character is typed, or on <Tab> when there are less
than 3 characters.

{{tdn:image154.png?275x32}}

==== ucDateRangeDays ====

...\DevGIT\LS.Common\LS.Common.Presentation\bin\Debug\LS.Common.Presentation
.dll

{{tdn:image155.png?451x37}}

==== ucDateRangeMonths ====

...\DevGIT\LS.Common\LS.Common.Presentation\bin\Debug\LS.Common.Presentation
.dll

{{tdn:image156.png?451x46}}

==== ucGridSmart.cs ====

...\DevGIT\ls.common\LS.Common.Grid\bin\Debug\LS.Common.Grid.dll

//See video above on how to add, but in short look for it in the Toolbox by
searching for ‘uc’.//
```

```
Our most basic GridView wrapper user control. Includes some code to autoset
the column widths

It does not include any extra controls. No filter, record count, <Refresh>
etc.

{{tdn:image157.png?252x213}}

==== ucGridSmart2.cs ====

...\DevGIT\ls.common\LS.Common.Grid\bin\Debug\LS.Common.Grid.dll

This is a ucGridSmart with a Filter, Record Count and <Refresh> button.

### Need to create this!

{{tdn:image158.png?393x260}}

==== ucGridStatus.cs ====

...\DevGIT\ls.common\LS.Common.Grid\bin\Debug\LS.Common.Grid.dll

Our grid with 4 colored radio button filters (originally used for the
eTMsync log status), a search/filter control, record count and <Refresh>.

### There are some properties here to control visibility of some uc elements
that DO NOT work here but do work in the ucGridStatus2 user control. We need
to move that code over.

{{tdn:image159.png?411x280}}

==== ucGridStatus2.cs ====

//See video above on how to add, but in short look for it in the Toolbox.//

...\DevGIT\ls.common\LS.Common.Grid\bin\Debug\LS.Common.Grid.dll

Used in dbEntreé, //this is ucGridStatus in a RadDock document window with a
tool window containing a tree view for a list of queries.//

### There are some properties here to control visibility of some uc elements
that work here but do not work in the basic ucGridStatus user control. I
probably should have put the original ucGridStatus in this new user control
but I actually copied what we had before. This enables us to change this
beyond the original but means common code changes must be maintained.

{{tdn:image160.png?499x253}}

=== Legality Software Properties ===

  * FormatString - [DefaultValue("{0}: {1}/{2}")]
```

```
  * RefreshVisible - Query Refresh button above GridView show/hide
  * RecordRename - "Records:"
  * QueryToolbarVisible - The tool-bar above the GridView
  * RefreshEnabled - Query Refresh button above GridView enable/disable
  * RecordCountVisible - "Visible count/total count" above GridView
  * ViewsToolBarVisible - Views tool-bar above the Query List tree
  * ViewsExpandCollapseAllVisible - The <+>/<-> buttons next to the filter
controls above the Query List tree
  * FilterVisible
  * NoGridDataMessage
  * GroupPanelText
  * LayoutSourceFile
  * GroupPanelHeight
  * SearchVisible - result set filter above the Query List tree
  * EnableTreeSearch
  * SelectedTreeNode
  * SearchRowVisible
  * OptionRowVisible
  * QueryListTreeSearchRowVisible
  * OptionAllText
  * OptionAllForeColor
  * OptionAllVisible
  * Option1Text
  * Option1ForeColor
  * Option1Visible
  * Option1WatchValue
  * Option2Text
  * Option2ForeColor
  * Option2Visible
  * Option2WatchValue
  * Option3Text
  * Option3ForeColor
  * Option3Visible
  * Option3WatchValue
  * WatchColumn
  * WatchColumnVisible
  * FormatGridRows
  * HighlightingEnabled
  * FilterType
  * AutoRowHeight
  * AutoColumnWidth
  * ShowTooltips
  * GridType
  * QueryListDropDownVisible - The main application config will enable the
user to show/hide either or both the QueryListDropDown and the
QueryListTree, so that should be a settable property of the UC. But the
application should also hide both query lists if there is only one query in
the *.SQLDT file so your code needs to deal with that too.
  * QueryListTreeVisible - This property should hide/show the
rdtwQueryListTree. It will be set based on the main application user
configurable options.
```

```
  * MultiSelectRows
  * OverrideFormatString
  * TreeViewDockState
  * TreeViewCaption
  * DataSource
  * QueryName

=== Legality Software Events ===

  * PositionChangedEventHandler SelectedQueryChanged
  * PositionChangingEventHandler SelectedQueryChanging
  * RadTreeView.RadTreeViewEventHandler SelectedTreeNodeChanged
  * RadTreeView.RadTreeViewCancelEventHandler SelectedTreeNodeChanging

**ucListControlWithFilter.cs**

...\DevGIT\LS.Common\LS.Common.Presentation\bin\Debug\LS.Common.Presentation
.dll

[[https://www.telerik.com/products/winforms/listcontrol.aspx|RadListControl]
] with our standard type to filter textbox and <X>.

Our [[#radlistcontrol-rlc|RadListControl help]] in this document

[[https://docs.telerik.com/devtools/winforms/controls/dropdown-listcontrol-a
nd-checkeddropdownlist/listcontrol/listcontrol|Telerik RadListControl Docs]]

==== ucItemList.cs ====

...\DevGIT\ls.common\LS.Common.ItemList\bin\Debug\LS.Common.ItemList.dll

{{tdn:image161.png?228x261}}

==== ucLogData.cs ====

...\DevGIT\LS.Common\LS.Common.Log.Presentation\bin\Debug\LS.Common.Log.dll

{{tdn:image157.png?252x213}}

==== ucLogViewer.cs ====

...\DevGIT\LS.Common\LS.Common.Log.Presentation\bin\Debug\LS.Common.Log.dll

{{tdn:image162.png?538x261|C:\Users\rober\AppData\Local\Temp\SNAGHTML2dbdb37
8.PNG}}

==== ucPassword.cs ====

...\DevGIT\LS.Common\LS.Common.Presentation\bin\Debug\LS.Common.Presentation
.dll
```

```
{{tdn:image163.png?201x32}}

==== ucUsernamePassword.cs ====

...\DevGIT\LS.Common\LS.Common.Presentation\bin\Debug\LS.Common.Presentation
.dll

{{tdn:image164.png?200x31}}

==== ucQueryConfig.cs ====

//Can’t find this in Common//

==== ucRichTextEditorHighlighter ====

...\DevGIT\LS.Common\LS.Common.Presentation\bin\Debug\LS.Common.Presentation
.dll

==== ucScheduler.cs ====

...\DevGIT\LS.Common\LS.Common.Scheduler\bin\Debug\LS.Common.Scheduler.dll

{{tdn:image165.png?394x174}}

==== ucSqlDbName.cs ====

...\DevGIT\LS.Common\LS.Common.Presentation\bin\Debug\LS.Common.Presentation
.dll

==== ucSqlEditor.cs ====

...\DevGIT\LS.Common\LS.Common.SqlEditor\bin\Debug\LS.Common.SqlEditor.dll

This is the basic editor with SQL syntax highlighting. No toolbars, extra
controls or functionality.

{{tdn:image166.png?415x296}}

==== ucSqlEditor2.cs ====

...\DevGIT\LS.Common\LS.Common.SqlEditor\bin\Debug\LS.Common.SqlEditor.dll

This is the SQL editor with toolbars for SQL editing, inserting
<SqlParameter> values, executing the query.

This does NOT include the DB tree ToolWindow showing tables and columns.

{{tdn:image167.png?484x345}}

==== ucSqlEditor3.cs ====
```

```
...\DevGIT\LS.Common\LS.Common.SqlEditor\bin\Debug\LS.Common.SqlEditor.dll

This is a RadDock, containing a SqlEditor2 (with toolbars for SQL editing,
inserting <SqlParameter> values, executing the query) in the Document
window, and a ucTableViewer in a ToolWindow showing tables and columns.

{{tdn:image168.png?720x333|C:\Users\rober\AppData\Local\Temp\SNAGHTML11178bc
5.PNG}}

==== ucSqlEditor4.cs ====

...\DevGIT
\ls.common\LS.Common.NewSqlEditor\bin\Debug\LS.Common.NewSqlEditor.dll

This control contains the new version of ActiproSoftware.

This is the SQL editor with toolbars for SQL editing, inserting
<SqlParameter> values, executing the query.

This does NOT include the DB tree ToolWindow showing tables and columns.

{{tdn:image169.png?544x313}}

==== ucSqlEditor5.cs ====

...\DevGIT
\ls.common\LS.Common.NewSqlEditor\bin\Debug\LS.Common.NewSqlEditor.dll

This control contains the new version of ActiproSoftware.

This is a RadDock, containing a SqlEditor4 (with toolbars for SQL editing,
inserting <SqlParameter> values, executing the query) in the Document
window, and a ucTableViewer in a ToolWindow showing tables and columns.

{{tdn:image170.png?725x335}}

==== ucTableViewer.cs ====

...\DevGIT\LS.Common\LS.Common.TableViewer\bin\Debug\LS.Common.TableViewer.d
ll

{{tdn:image171.png?248x367}}

==== ucTelerikThemeSelector.cs ====

...\DevGIT\LS.Common\LS.Common.Presentation\bin\Debug\LS.Common.Presentation
.dll

{{tdn:image172.png?252x43}}

==== ucTreeFilter.cs ====
```

```
...\DevGIT\LS.Common\LS.Common.Tree\bin\Debug\LS.Common.Tree.dll

{{tdn:image173.png?276x29}}

==== ucUpdaterSettings.cs ====

...\DevGIT\LS.Common\LS.Common.Presentation\bin\Debug\LS.Common.Presentation
.dll

==== ucDatabaseConnector.cs ====

...\DevGIT\LS.Common\LS.Common.TMDBConnector\bin\Debug\LS.Common.TMDBConnect
or.dll

{{tdn:image174.png?388x159}}

==== ucTMDatabaseConnector.cs ====

...\DevGIT\LS.Common\LS.Common.Presentation\bin\Debug\LS.Common.Presentation
.dll

//### Also here, so we need to clean up one or the other:\\
//...\DevGIT\LS.Common\LS.Common.TMDBConnector\bin\Debug\LS.Common.TMDBConne
ctor.dll

**NOTE: This one also contains the ucDatabaseConnector.cs**

{{tdn:image175.png?558x248}}

==== ucDbWikiOptions ====

X:\DevGIT\LS.Common\LS.Common.DbWiki\bin\Debug\LS.Common.DbWiki.dll

{{tdn:image176.png?332x122}}

==== ucDbWikiBrowser ====

X:\DevGIT\LS.Common\LS.Common.DbWiki\bin\Debug\LS.Common.DbWiki.dll

{{tdn:image177.png?377x209}}

==== ucEmailConnector.cs ====

...\DevGIT\LS.Common\LS.Common.Presentation\bin\Debug\LS.Common.Presentation
.dll

**To use this you must also add a reference to:\\
**...\DevGIT\ls.common\LS.Common.Communication\bin\Debug\LS.Common.Communica
tion.dll

{{tdn:image178.png?312x317}}
```

```
{{tdn:image179.png?310x314}}

{{tdn:image180.png?305x310}}

{{tdn:image181.png?299x304}}

===== Legality Extended (via inheritance) Controls =====

==== LSRadPropertyGrid - lsrpg ====

Extended with these settable properties under the ‘Legality Software’
branch:

  * AutoFitLabel – flag to autosize the label column width
  * ValueColumnWidthPercentage – (0.0-1.00) % of the total width is assigned
to the value column
  * RequiredFieldBackground – color for required fields

====== LS.Common Library ======

===== TMDBConnector & ucTMDatabaseConnector Control =====

To use the ucTMDatabaseConnector in a project:

  - Add an ‘Existing Project’ to the solution and choose:
“C:\DevGIT\LS.Common\LS.Common.TMDBConnector\LS.Common.TMDBConnector.csproj”
\\
### This is the old version. Need to update it to the one in eLaw ###
  - Build the project. The user control won’t appear in the toolbox until
the solution is built with it included. You will get an error if the DLLs
don’t exist locally for any required subprojects. For example TMDBConnector
uses “C:\TFS\LS.Common\Main\LS.Common.Utilities\LS.Common.Utilities.csproj”.
So add other required projects per any compile errors.
  - Add the ucTMDatabaseConnector to your form from the VS Toolbox.

//NOTE: This version
“C:\TFS\LS.Common\Main\LS.Common.Presentation\TMDBConnector “ is old. Don’t
use it. We need to archive this one out.//

===== Add an Existing Form from another project  =====

For example add the somewhat standardized Options form (from
LS.TMLinkedDocumentMonitor.Presentation ) that includes a wired up instance
of the TMDBConnector & ucTMDatabaseConnector Control.

Add rfrmOptions to your Presentation project:

  * First you must add any Common Lib project(s) to the solution that will
be used by the form you will add. The Options form requires these LS Common
Libraries:
    *
```

```
“C:\TFS\LS.Common\Main\LS.Common.Utilities\LS.Common.Utilities.csproj”.
    *
“C:\TFS\LS.Common\LS.Common.TMDBConnector\LS.Common.TMDBConnector.csproj”
    * “C:\TFS\LS.Common\Main\LS.Common.TimeMatters\
LS.Common.TimeMatters.csproj”
    * “C:\TFS\LS.Common\Main\LS.Common.Serialization\
LS.Common.Serialization.csproj”
    * C:\TFS\LS.Common\Main\LS.Common.DataLayer\ LS.Common.DataLayer.csproj”
  * Manually add a Reference for each Common Lib (ex:
LX.Common.TMDBConnector) to the Project\References. This is necessary
because you won't be manually adding the user control from the toolbox.
  * rfrmOptions also requires the Settings.cs abd DBSettings.cs files as
well so add them to the Presentation project.
  * Copy the 3 form files to the new project folder (ex:
'rfrmOptions.Designer.cs', 'rfrmOptions.resx', 'rfrmOptions.cs').
  * Right click the Presentation project and “Add | Existing Item” then
choose (just) the 'rfrmOptions.cs' file.
  * Find/Replace the ('rfrmOptions.cs') namespace to match new Project
namespace. You should only have to change it at the top of the
'rfrmOptions.cs' file (below the 'using's) and it will patch up the other
files are necessary.

//Not sure if this is worth the effort since there is so much manual fix up
required. We would maybe be better off setting up a template.//

===== ChilkatDotNet4.dll Library =====

http://www.chilkatsoft.com/

==== C# Documentation ====

http://www.chilkatsoft.com/refdoc/csharp.asp

==== C# Examples Online ====

http://www.example-code.com/csharp/default.asp

==== Email Features ====

=== Errors ===

See https://www.chilkatsoft.com/win32_Framework4.asp

**Error 1:** ‘Could not load file or assembly 'ChilkatDotNet4' or one of its
dependencies. An attempt was made to load a program with an incorrect
format.’

The Chilkat .NET assembly is a mixed-mode assembly. It provides a managed
API, but the inner core contains native code. A process must load the
correct assembly at runtime. A 32-bit process must load the 32-bit
ChilkatDotNet4.dll, and a 64-bit process must load the 64-bit
```

```
ChilkatDotNet4.dll. This error occurs when a process tries to load the
incorrect format.

**Error 2:** ‘Could not load file or assembly 'ChilkatDotNet4.dll' or one of
its dependencies. The specified module could not be found.’

The Chilkat .NET 4.0 assembly requires the VC++ 2010 runtime to be installed
on the computer. Both eTMsync installers install this runtime by default.

{{tdn:image182.png?526x385}}

Be sure to include the VC++ 2010 runtime as a prerequisite in any installer
that uses the Chilkat library. Setup the installer to download it from
here:\\
https://download.microsoft.com/download/5/B/C/5BC5DBB3-652D-4DCE-B14A-475AB8
5EEF6E/vcredist_x86.exe

===== ExceptionReporter =====

NOTE: First you must install and license the Namtuk Screen Capture Library
on your development PC.

  * Download:
http://www.namtuk.com/download.aspx?email=bizmgr@mattersinmotion.com&orderid
=460910105-1
  * License installation password: 45a963-zbnglob

  - Add references to LS.Common.Utilities and
LS.Common.ExceptionReporter.Presentation from the main project (usually the
one ending in .Presentation)
  - Change the program.cs file, Program.Main method to look like this:

private static void Main()

{

using (new SingleInstanceChecker(1000)) %%//%% Allow this application to
only run once (1 instance)

{

CrashReporter.Initialize(new CrashConfiguration

{

ToEmail = "exceptionreporter.list@legalitysoftware.com", %%//%% the email to
use when sending error reports

Folder = SystemSettings.GetLogsFolder(), %%//%% the folder where the report
is saved (HTML and XML)
```

```
InstallVersion = SystemSettings.GetInstallVersion(), %%//%% version number;
I'm normally reading it from the registry

});

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new rfrmMain());

}

}
```

The first line should only be there if you want a single instance of the application. That's the case normally but if you do want to be able to launch multiple copies, remove that line. The 1000 argument is the timeout in milliseconds - there's no reason to change it.

The application name will be used to create the settings and log folders - they'll be under C:\ProgramData\Legality Software\{app name}. It will default to "uninitialized" if you forget to set it - if you see that folder below Legality Software, check the Program.cs file.

The naming of the main form is not always the same between project, sometimes it's rfrmMain, sometimes MainForm - that line might have to be changed accordingly.

You may also need to also add a reference to _3rdpartylibraries\Namtuk\My Screen Capture\1.5\Namtuk.MyScreenCaptureNET.dll

==== Enable/Disable ExceptionReporter ====

CrashReporter may be enabled/disabled via the app.config file | appSettings section | CrashReporterEnabled key. A sample can be found into the app.config file from LS.Common.ExceptionReporter.Presentation.Tester. CrashReporterEnabled = false disables the ExceptionReporter. If the key is missing from the configuration file then the CrashReporter is enabled.

//NOTE: 07/17/2018 I asked Adi to change the CrashReporterEnabled key to ExceptionReporterEnabled.//

==== Manual Exception test menu option ====

===== LS.Common.EmailSupport =====

==== LS.Common.EmailSupport.Test ====

Test project demonstrating the implementation and active testing of the EmailSupport project.

```
Main tabs of the form LS.Common.EmailSupport.Test:

=== Email Support tab - On “Open EmailSupport” button click ===

  * Gets the data from the UI textboxes and maps it to the EmailSettings
object. All the textbox values are assigned to properties of the
EmailSettings object.
  * Passes the EmailSettings object to the rfrmEmailSupport form and
displays it

=== CC Validation tab – On “Validate” button click ===

  * Creates a new instance of the rfrmEmailSupport form but doesn’t displays
it
  * Sends the “CC” textbox value to the rfrmEmailSupport class ParseCCs
method
  * Returns the valid CC email addresses to the “Parsed CC” textbox

=== Zip Compress tab – on “Zip Files” button click ===

  * Creates a new instance of the Zip class
  * Reads the text inserted in the “File Paths” textbox, splits the text by
new line and stores the values as a list
  * Sends the list and the value of the “OutputFileName” textbox to the Zip
class Compress method. For the second parameter of the Compress method, it
sends the value of the “OutputFileName” textbox. If the value is empty, it
will generate a name using the provided rule ( {date}_{time}.zip ).
  * It displays the returned value of the Compress method, which is the zip
file path, in the “Output Directory” textbox. The textbox is read only.
  * Opens an Explorer window to the path of the zip file for a faster
examination during tests

===== SoftwareKey/SOLO Licensing =====

==== LS.Common.Licensing ====

==== LS.Common.Licensing.Test2 ====

A clean implementation example of adding our SK/SOLO license class to a
Telerik app.

====== Tools ======

===== ControlSpy =====

http://www.telerik.com/help/winforms/tools-controlspy-overview.html

http://www.telerik.com/community/forums/winforms/tools.aspx\\
Video:
http://www.telerik.com/videos/winforms/telerik-radcontrols-for-winforms---us
ing-the-radcontrols-spy
```

```
The Control Spy is a tool to let you examine the detailed internal structure
of any RadControl. It is not a stand-alone tool. Rather, the Control Spy is
supplied as part of a library that you compile into your application. After
adding this library, you can start the Control Spy and use it to examine and
alter properties of any RadControl which is currently running.

===== Visual Style Builder =====

Video:
http://www.telerik.com/videos/winforms/styling-basics-with-visual-style-buil
der-for-winforms\\
Video:
http://www.telerik.com/videos/winforms/introduction-to-the-new-visual-style-
builder-for-winforms

===== Theme Viewer =====

http://www.telerik.com/help/winforms/tools-themeviewer.html

ThemeViewer is a tool that gives you the ability to preview a custom or a
predifined theme for all themable controls of the suite.

**RadThemeManger\\
**http://docs.telerik.com/devtools/winforms/tools/theme-manager/theme-manage
r#radthememanager

RadThememanger is a component which can be dragged from the toolbox. It
purpose is to manage the XML files that contain the theme for the controls.

===== ShapeEditor =====

http://www.telerik.com/help/winforms/tools-shapeeditor-overview.html

http://www.telerik.com/community/forums/winforms/tools.aspx

Each RadElement / RadItem can be shaped by setting the Shape property. The
shape can be easily modified either in design mode or in Visual Style
Builder by using the Shape Designer. The Shape Designer provides an easy way
for creating all kinds of custom shapes only by adding, removing and
dragging the points which represent the shape. Each line can also be curved
only by right-clicking and making it a curved line.

===== Element Hierarchy Editor =====

http://www.telerik.com/help/winforms/tools-element-hierarchy-editor-overview
.html

When dealing with complex control types that have multiple nested elements,
such as a RadRibbonBar control, you want the ability to set properties at
multiple levels of the class hierarchy. For example, in the case of a
RadRibbonBar you can set properties of the RadRibbonBarElement or you can
```

```
set the properties of one of its RadTabStripElement elements or possibly one
of its RadDropDownButtonElement elements.

Since all controls inherit from one or more base classes it is also helpful
to be able to easily see the classes in the control hierarchy so you can set
the properties on the appropriate class. When designing a RadButton control
you can set base class properties or any properties of classes that
RadButton derives from. The RadButton Visibility property can be set at the
base class level (RootRadElement) while the RadButton Text property can be
set at the RadButtonElement class level.

===== JustCode =====

<Ctrl><Shift>
```

to see the shotcut list

ActiProSoftware Syntax Editor

Licensing

GIT Source Control on BitBucket

Watch: \\192.168.93.1\KB_MiM\GIT Source Control\20150211) GIT Add and Download w Marcel.mp4

https://xkcd.com/1597/ - the real answer

Install GIT for Windows

https://git-for-windows.github.io/

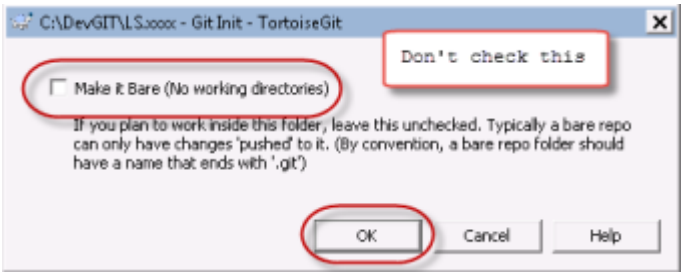
Install TortoiseGIT on your PC

Get it here: https://code.google.com/p/tortoisegit/wiki/Download

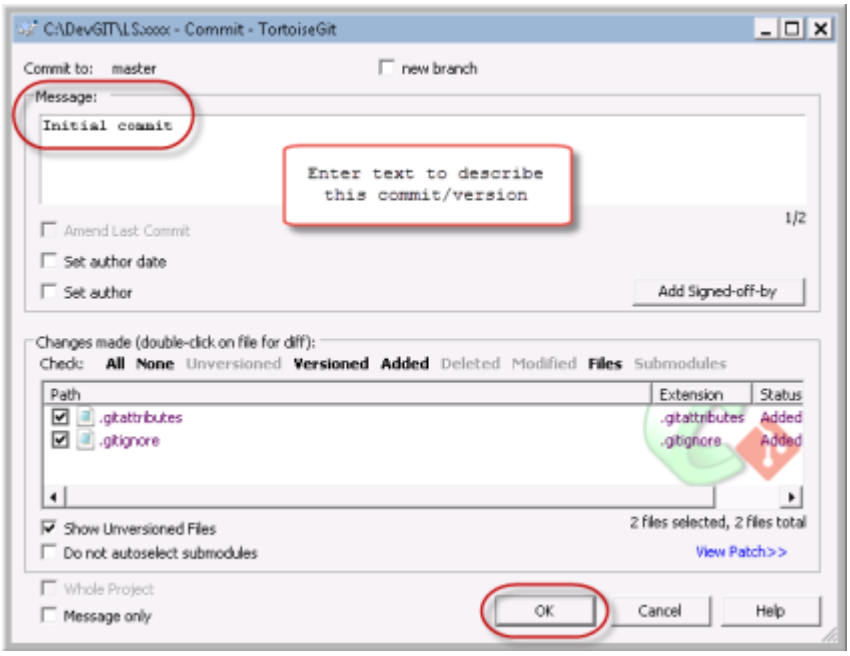
Create a root dev folder to hold all your projects here: “C:\DevGIT”

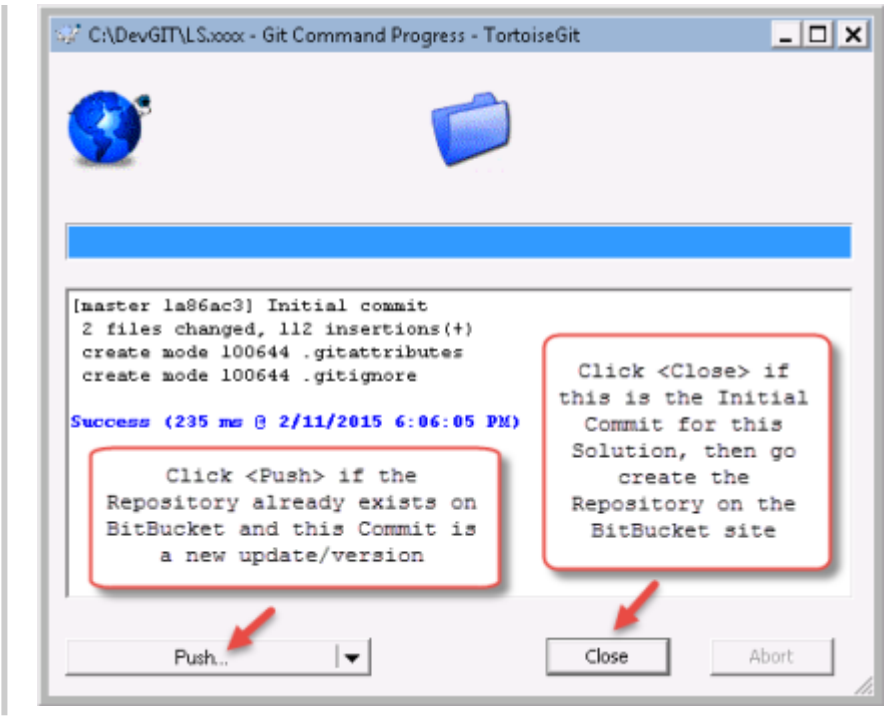
Create a Repository and Initial Upload of a New Solution

- Create the solution in a folder like “LS.xxxx” in your project root “C:\DevGIT”.
- Open the “C:\DevGIT\LS.xxxx” solution folder then right click a blank space and choose “Git Create repository here…”

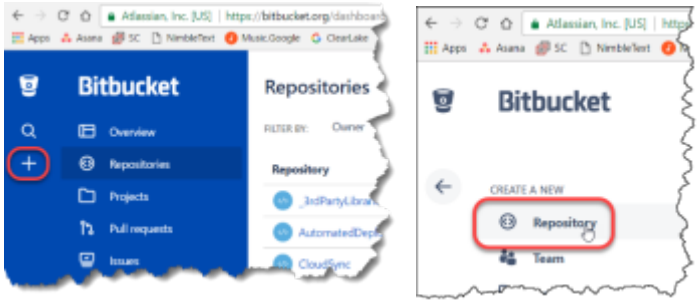


- Copy these files from an existing project folder (I suggest you keep a copy of them in “C:\DevGIT__GIT Default Settings”)
    - .gitignore –contains the list of file types GIT should ignore (exe, zip, dll, log, etc)
    - .gitattributes – default GIT behavior for specific file types
- From the “C:\DevGIT\LS.xxxx” solution folder (root of the solution) right click a blank space and choose “TortoiseGit | Add”
- Check all the files you want to push to GIT. Use “Select/deselect all” to check all. Use “Include ignored files” to show the files left out by the .gitignore” settings file. Click <OK>
- Click <Commit> on the “Add Finished!’ dialog. Your local Repository is now created and populated.

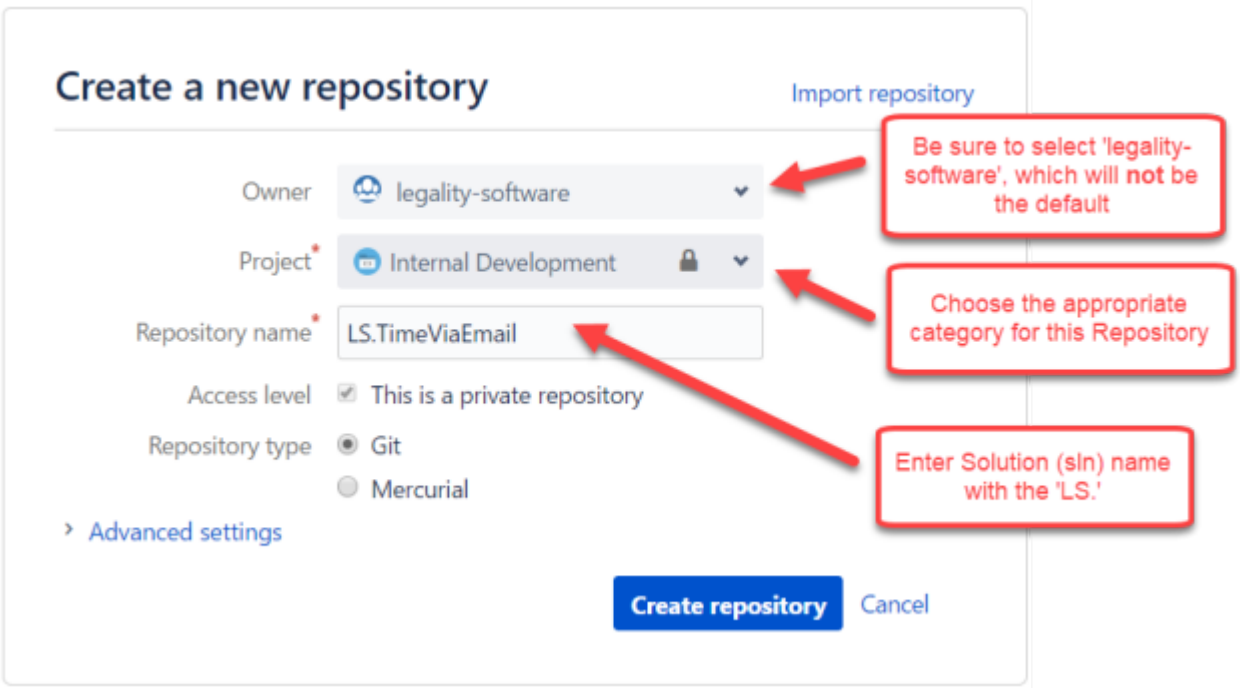


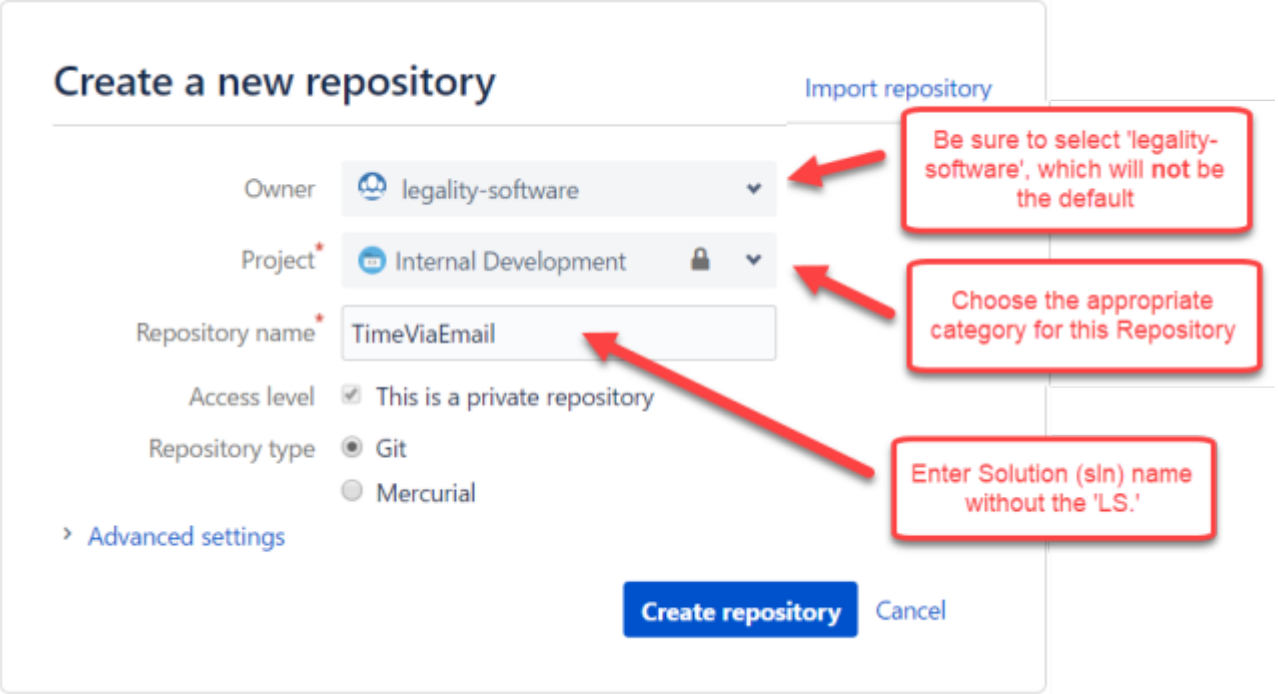


- Create a new Repository on the BitBucket site.



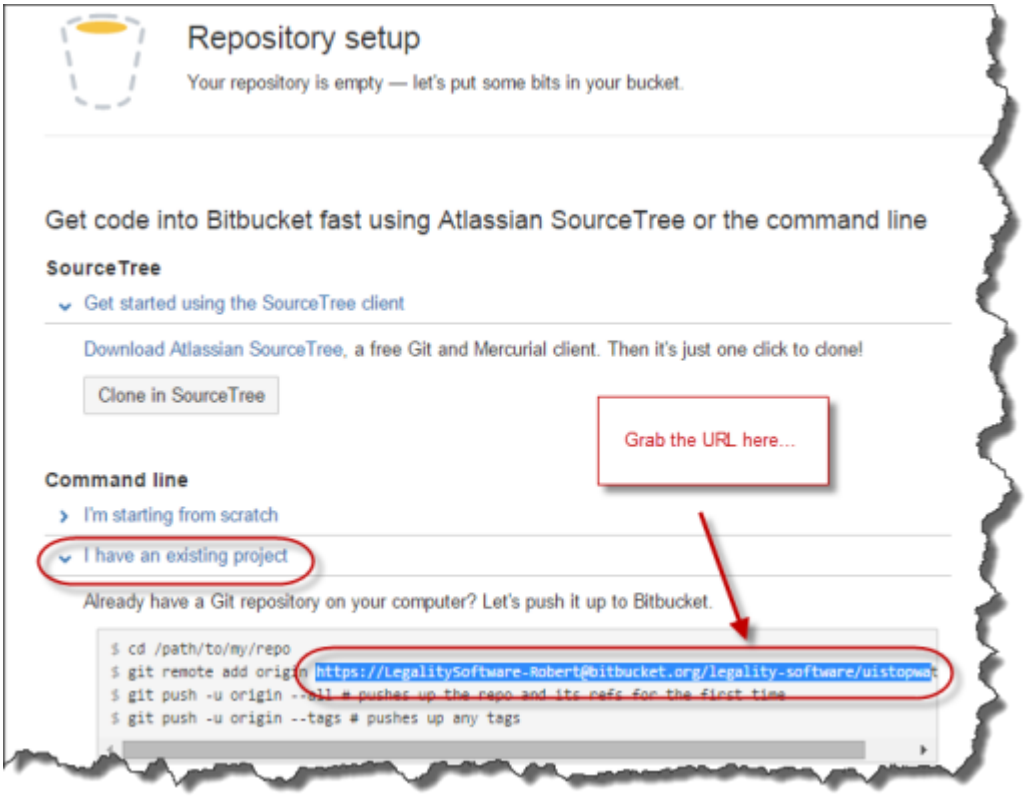
I can’t decide if the Repository should be named with or without the ‘LS. Here are both options…



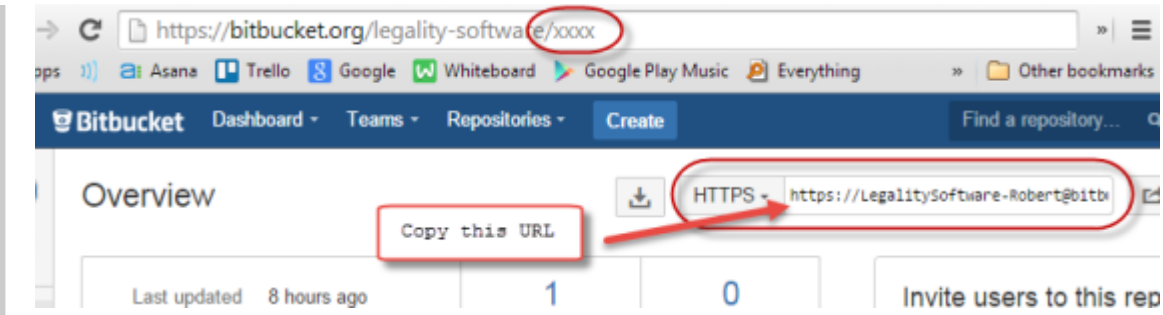


- Copy the HTTPS URL from the new Repository page

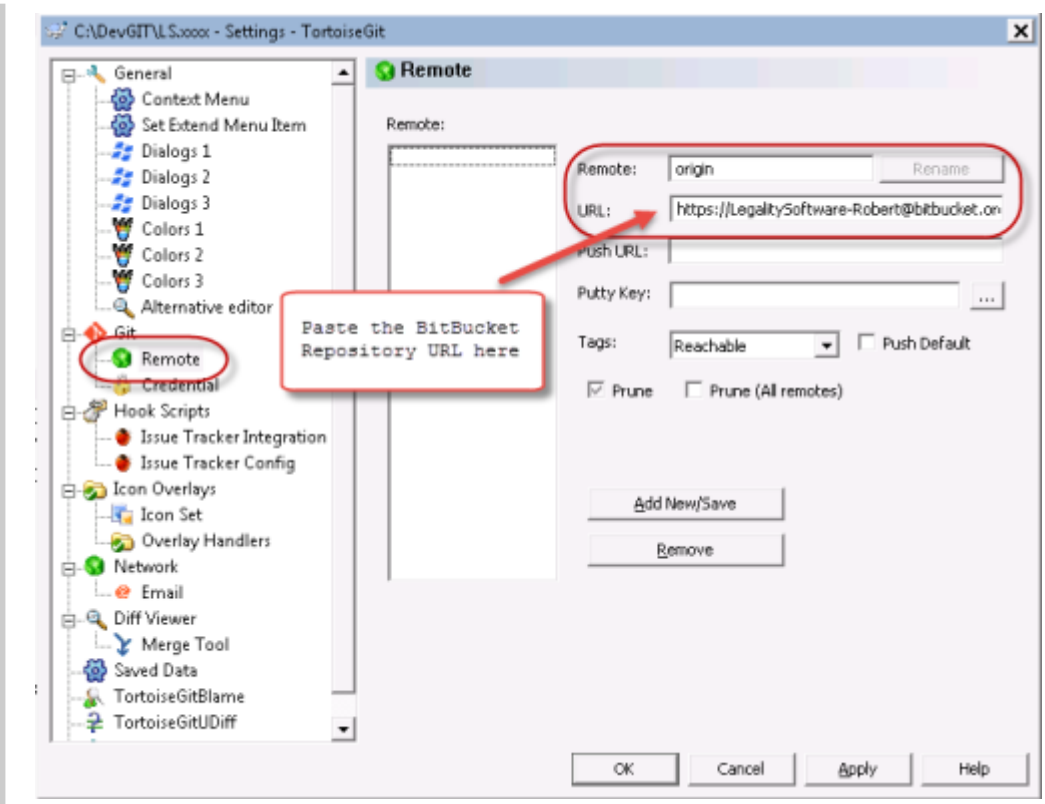
The first time you will need to pull the URL from “I have an existing project”



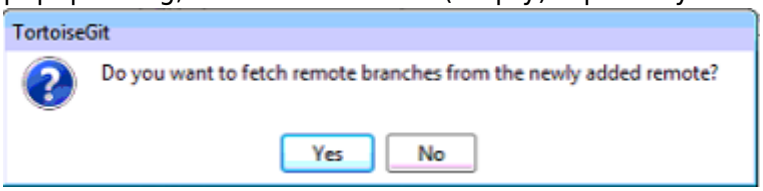
Once the Repository has been updated you can grab it from the upper right corner of the page like this.



- Back in the “C:\DevGIT\ LS.xxxx” solution folder right click a blank space and choose “TortoiseGit |Settings | Remote” and paste the URL into the “URL: field. The “Remote:” field will update to “origin”.



- Click <OK>.
- Click <No> on the “Do you want to fetch remote branches from the newly added remote?” popup dialog, since this is a new (empty) repository.

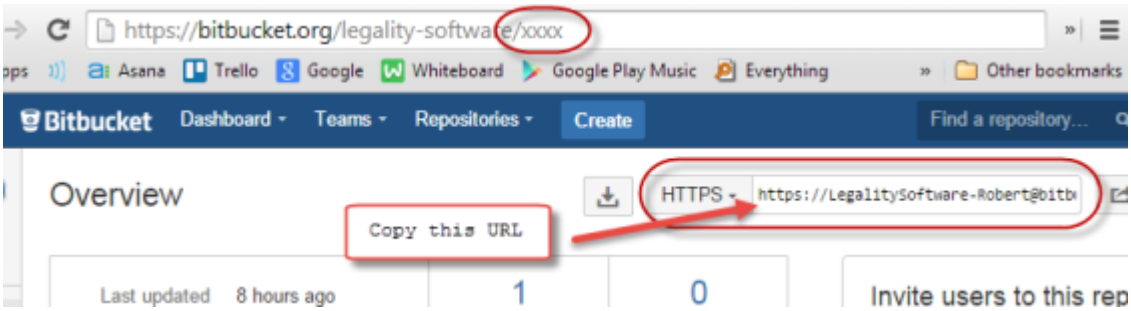


- Right click a blank space in the “C:\DevGIT\ LS.xxxx” solution folder and choose “Git Push…” then click <OK>
- Enter your BitBucket logon password.
- Click <Close> and you are done.

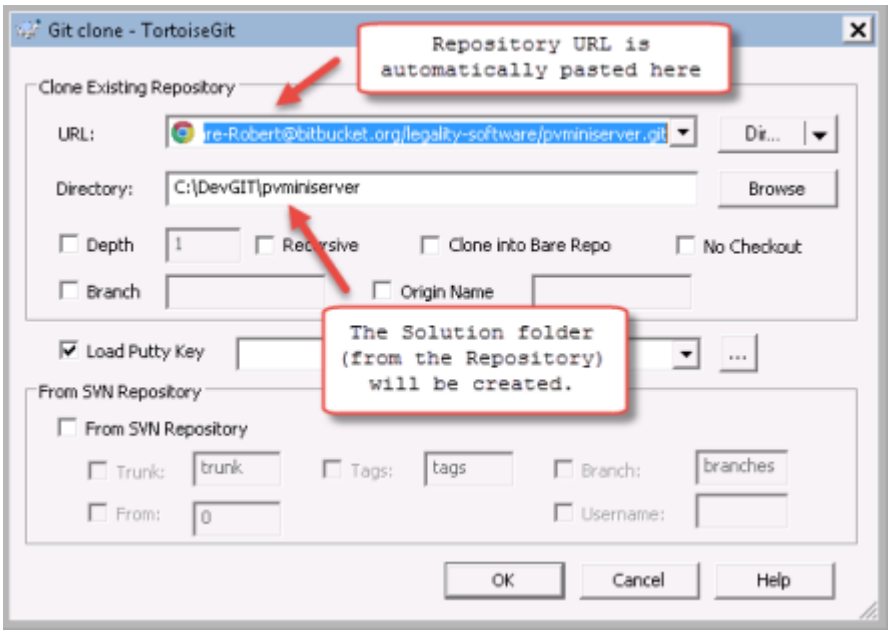
Initial Download of an Existing Solution from a Repository

- Log in to BitBucket and go to the Solution Repository page.
- Copy the HTTPS URL from the Repository page. The part before the @ is my username. After @

is the URL.



- Right click a blank space in “C:\DevGIT” (our solution root folder) and choose “Git Clone…”



- Click <OK>
- Enter your BitBucket logon password.
- Click <Close> and you are done.
- NOTE: GIT may create a folder without the LS. prefix. You can simply rename the local folder and it will be OK. Or you can edit the “Directory” field before starting the download.

Possible Issues

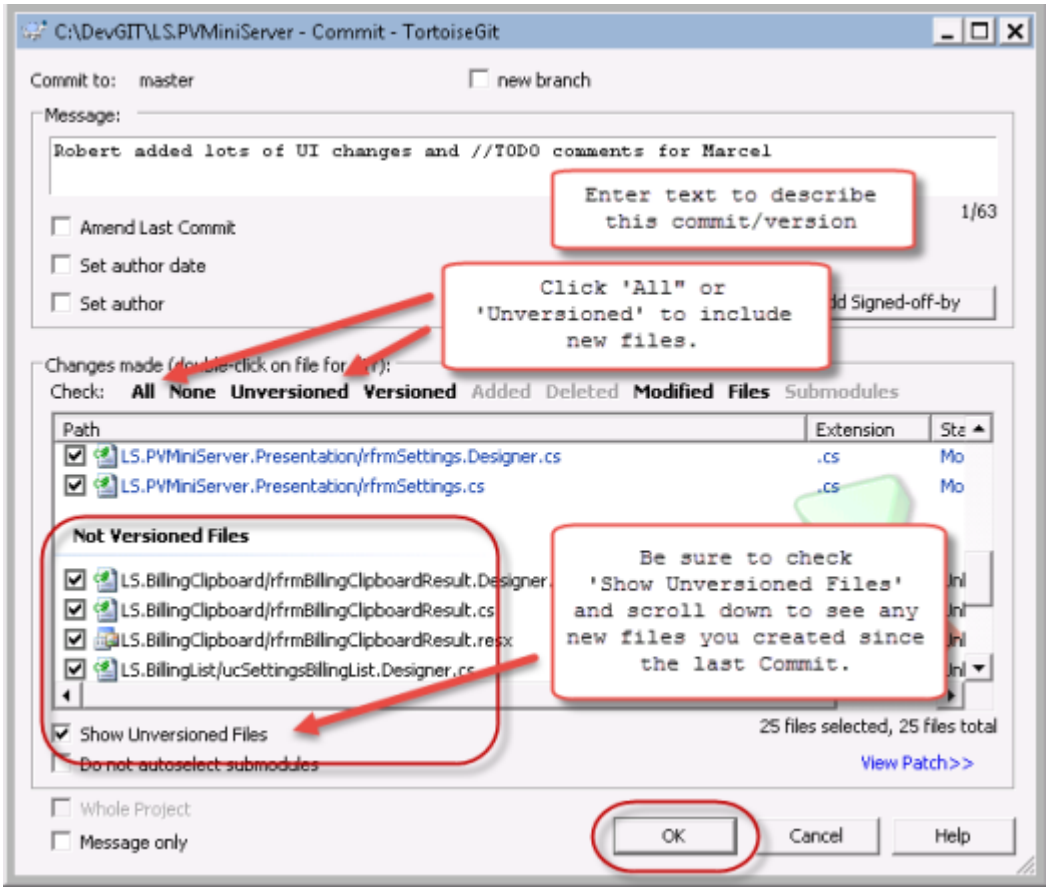
- Your Solution may require some NuGet Packages. These will not be restored by the GIT Clone operation. Do a Rebuild All and VS will download and restore any required packages.
- Be sure your Solution doesn’t require a new version of LS.Common or some other common library. If so refresh it and Rebuild All before you Rebuild All on this solution.
- There may be some new, but unfinished Project in the LS.Common library. If so it’s likely your Solution doesn’t even need this project, but it will still cause LS.Common to fail to compile. Follow the error(s) to the source Project, then “Remove” (<Del>) that Project from the Solution Explorer for now. Just don’t push this version of Common back up to GIT.

Commit a New Version/Update to an Existing Solution Repository

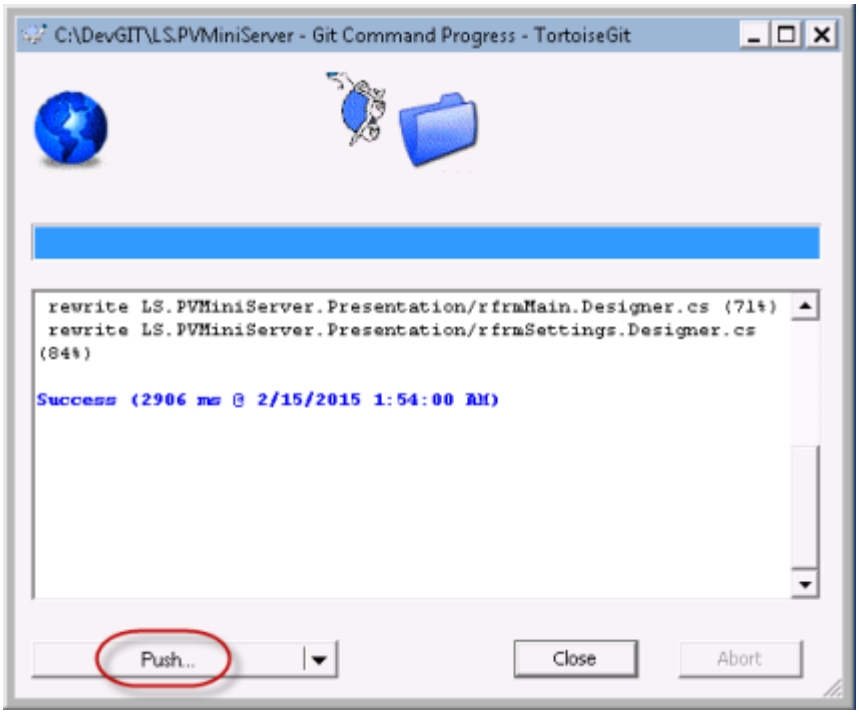
NOTE: Files in these folder of these types do not need to be included in the repo:

- /obj/*.json
- /obj/*.props
- /obj/*.targets
- /obj/*.cache
- /obj/Debug/net6.0-windows/*.cache
- /obj/Debug/net6.0-windows/*.json

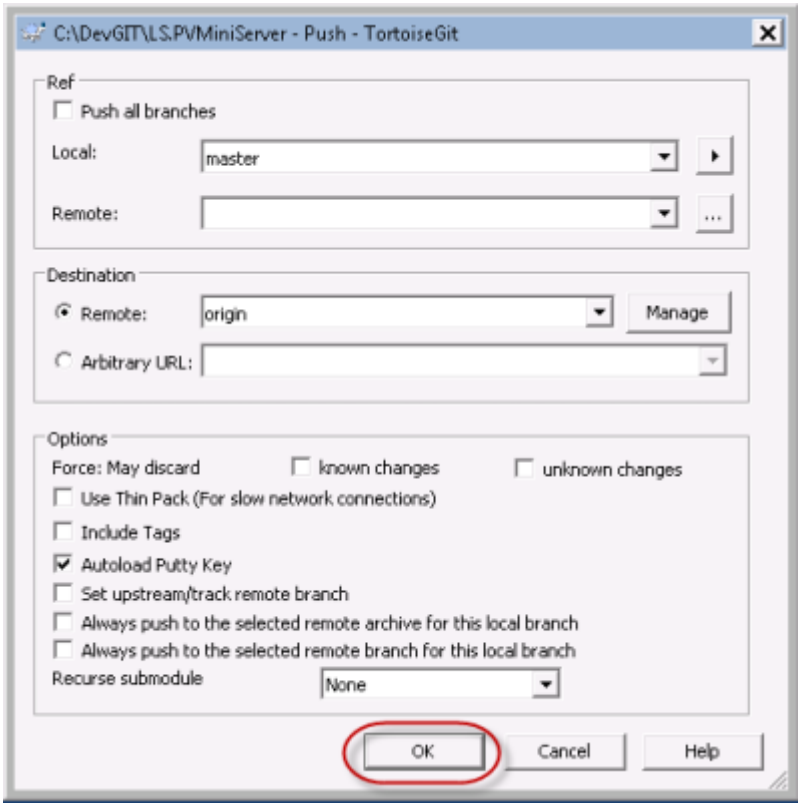
- Right click a blank space in the “C:\DevGIT\ LS.xxxx” solution folder and choose “Git Commit → master” then click <OK>



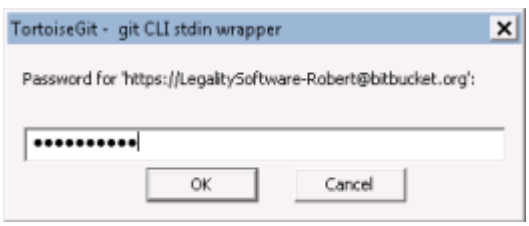
- No changes. Just click <Push>



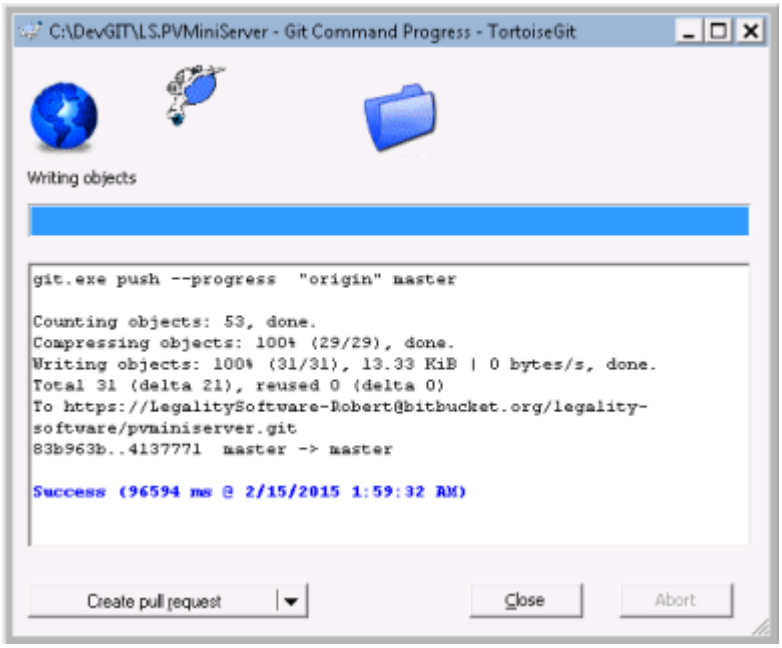
- No changes. Just click <OK>



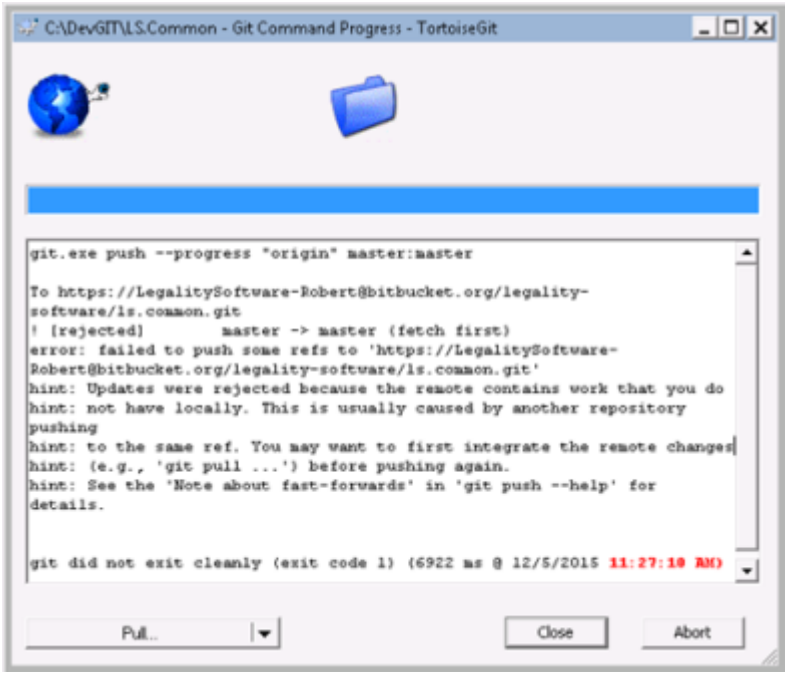
- Enter your BitBucket logon password and click <OK>



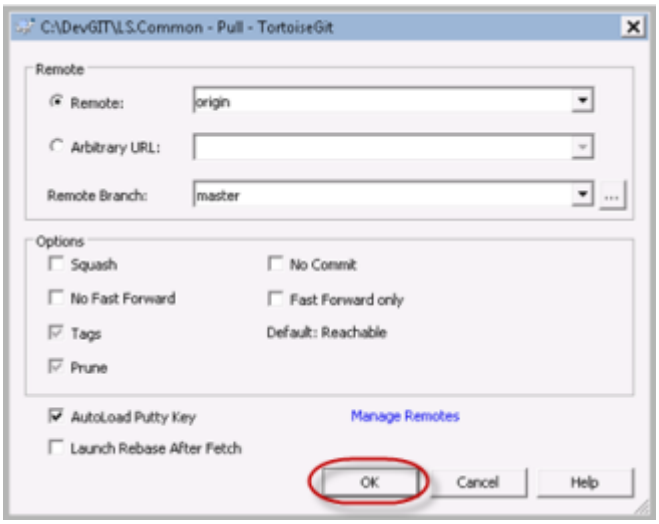
- Click <Close>



You will get this error if there are changes exist on GIT that aren't on your local PC. Click <Pull>



Click <OK>



Enter your GIT pw

If this goes well then try the Push again

Delete ('Revert') changes in the local Repository so you can Pull a fresh copy from the web

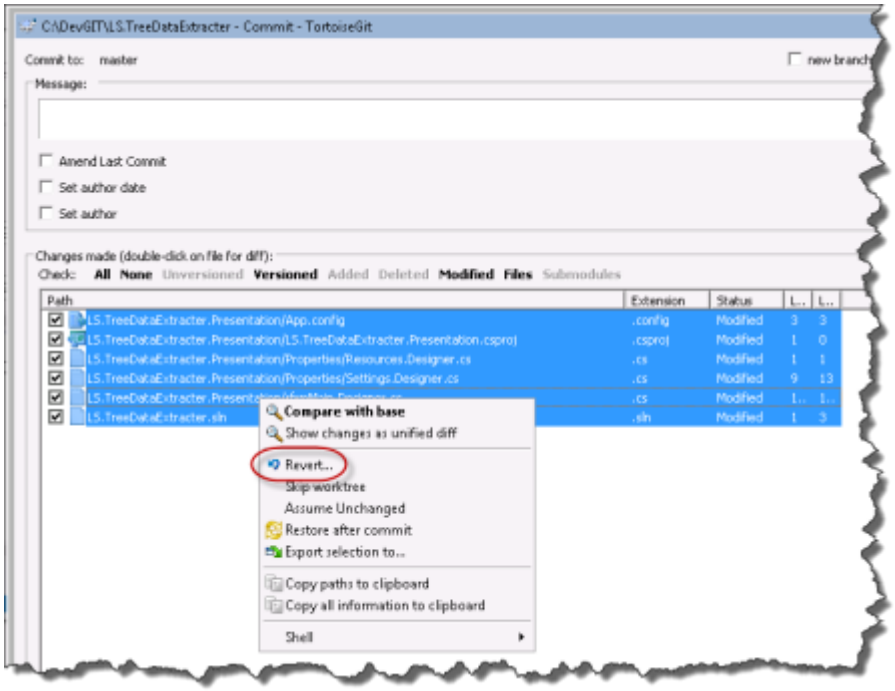
You might make local changes that you don't want to Commit to the web. Instead you want to do a Pull and have it overwrite these local changes.

You could delete the whole folder and do a new Clone, but here's an easier way.

Right click a blank space in the "C:\DevGIT\ LS.xxxx" solution folder and choose "Git Commit → master".

<Ctrl><A> to select all the locally changed files, or only some if you don't want to lose them all.

Right click one of the selected files and click 'Revert'.



Once the files are rolled back the Commit dialog <OK> button will be disabled. Just click the <X> to close.

Now you can pull the latest version from GIT.

Force a Local Repository Push to overwrite the GIT Repository version

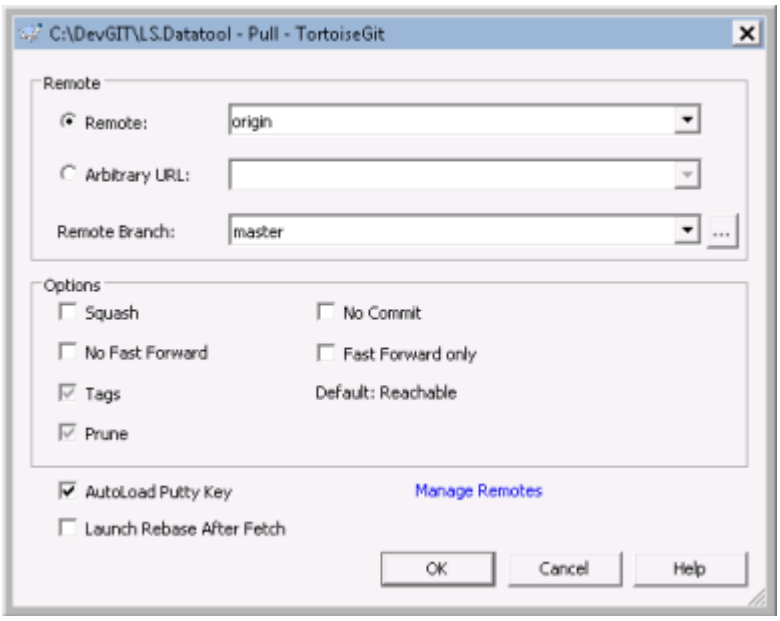
Need to document this...

Roll the current 'Master' version back to some previous log time

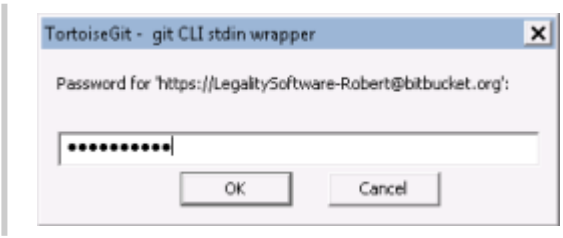
Pull the Current Version from an Existing Repository

NOTE: You do not have to close the Solution in VS before Pulling an update. Once the Pull is complete VS will ask if you want to reload and changed files.

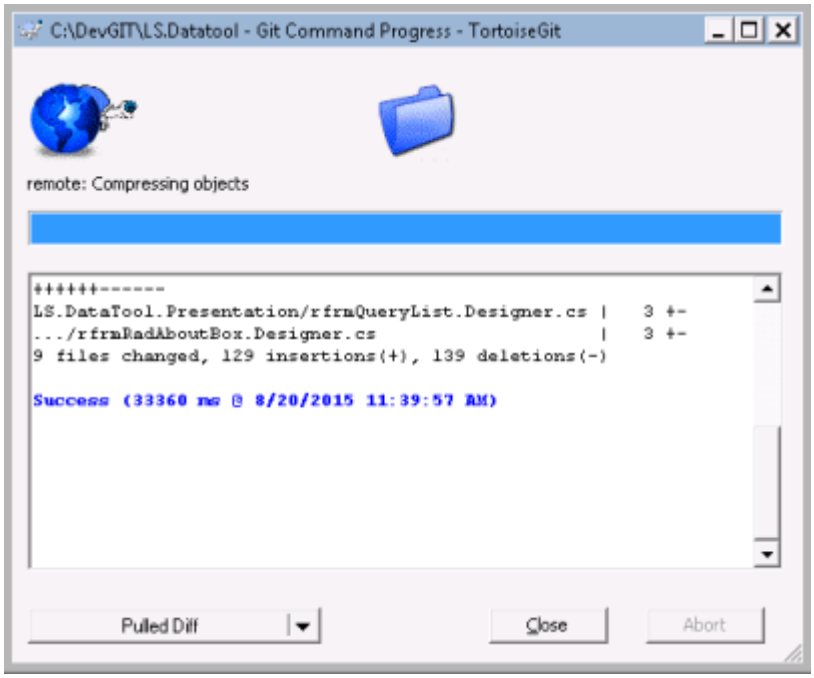
- Open the local solution folder and right click and choose "Git Pull".
- Click <OK>



- Enter your BitBucket logon password and click <OK>



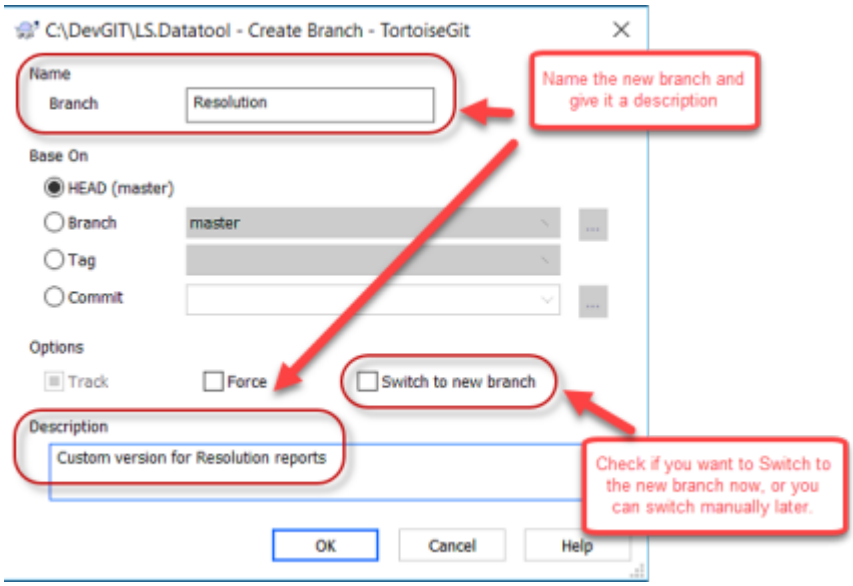
- The dialog should show "Success". Click <Close>



Branches - Create

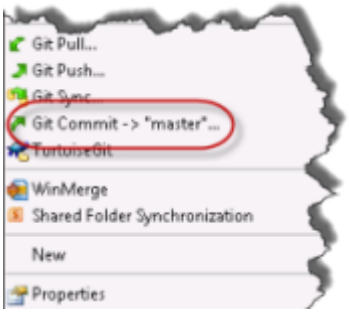
Be sure to Commit and Push your current data before creating a new Branch.

Right click inside the Solution folder | TortiseGIT | Create Branch...



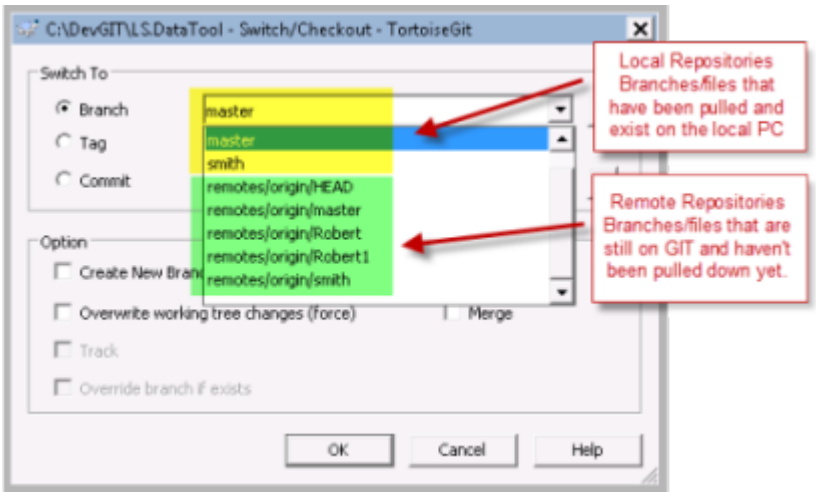
Branches – See the Current Branch

Right click inside the Solution folder and look here to see the current working Branch:



Branches – Switch

Right click inside the Solution folder | TortiseGIT | Switch/Checkout | choose the Branch you require



If you choose a remote branch (that hasn't been pulled yet) it will allow you to pull it.

The files in the folder will now appear just as they would in the branch.

You do not need to close Visual Studio when you change Branches. It will ask you to reload all changed files. **Let it do that!**

When you have multiple branches you should Switch and Push/Pull each individually as changes are made. There is no global Push/Pull.

Branches – Delete

Document this...

Delete Repository but not files

Delete the (hidden) .git folder in the root-folder of your repository to only delete the git-related information (branches, versions). Run the following commands from the root-folder of your repository:

List the hidden git folders in all subfolders:

DIR /AHD /S /B *.git

Delete your local git repository, but not the remote repo (if one exists):

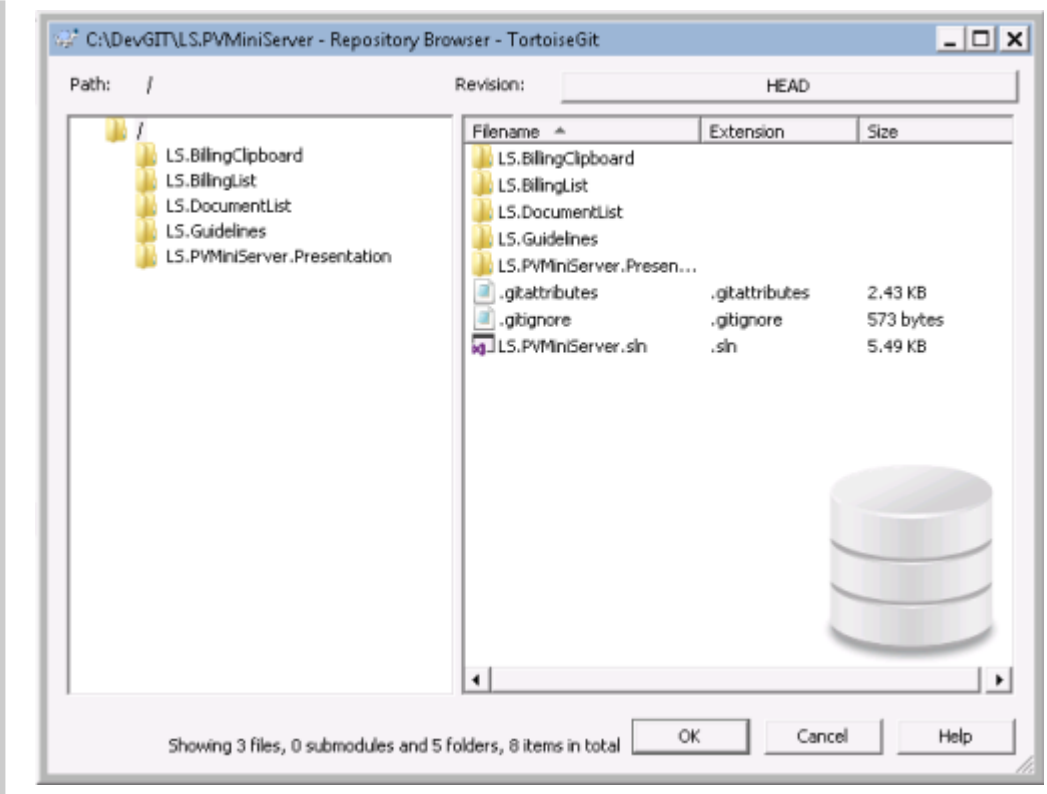
RMDIR /S .git (include /Q for no prompt)

Use this PowerShell script to delete the entire folder for a local repository

C:\> remove-item _Documentation -Force | Where { **$_**.PSIsContainer } *# use -Force to delete the locked '_Documentation' folder from the current folder*

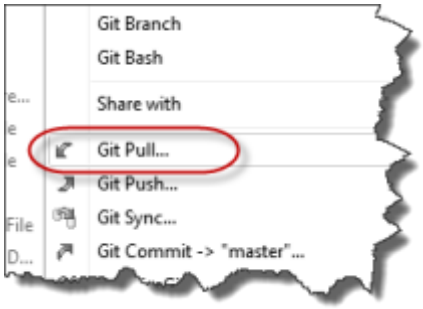
The Repository Browser

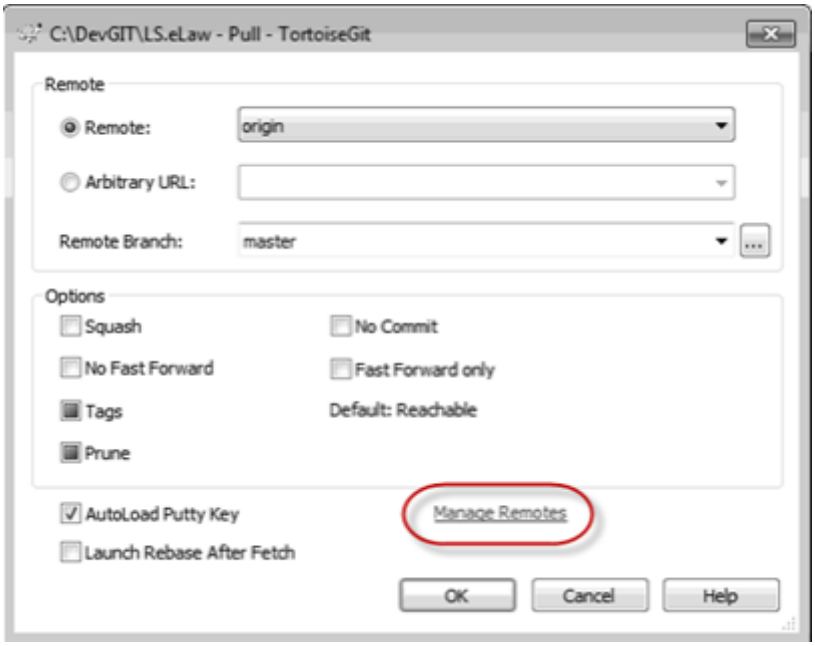
- Right click and choose “TortoiseGit | Repo-browser” to see the local Repository.

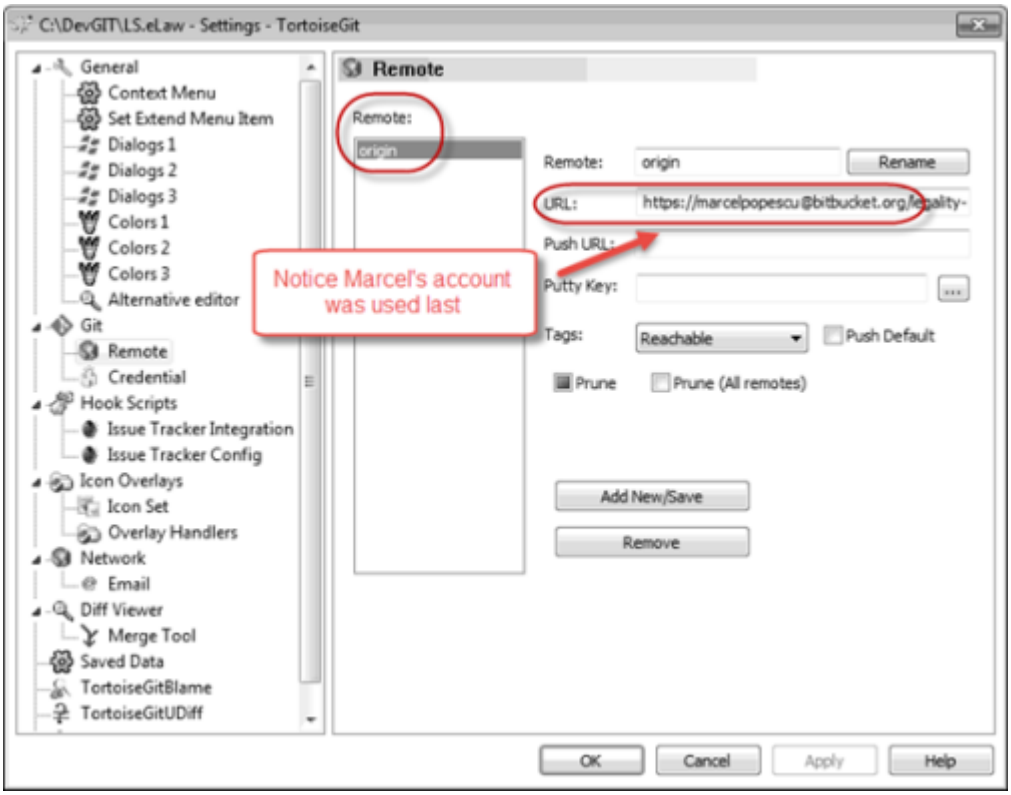


Change the Current GIT User in TortoiseGit

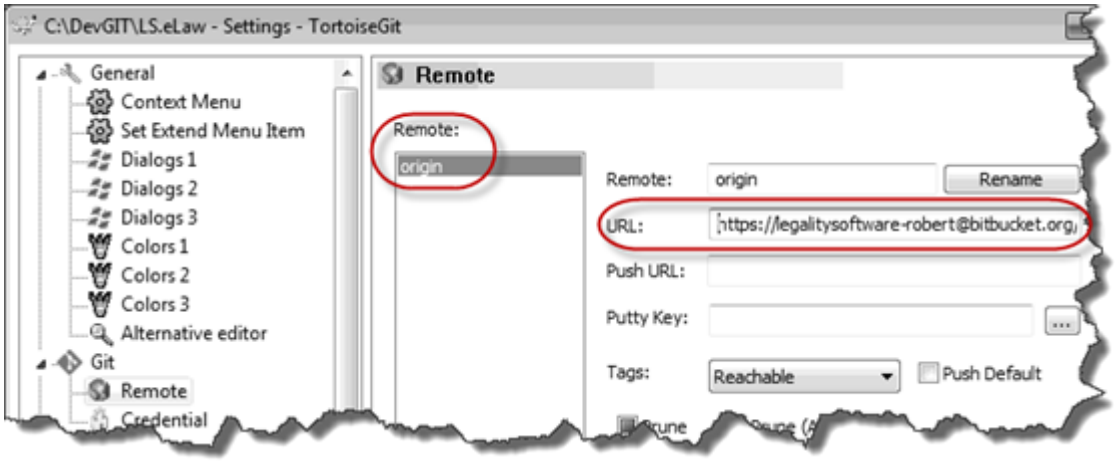
If some other user Pulled from GIT with teir account you will see the their account name on the password dialog. Here’s how you switch users.



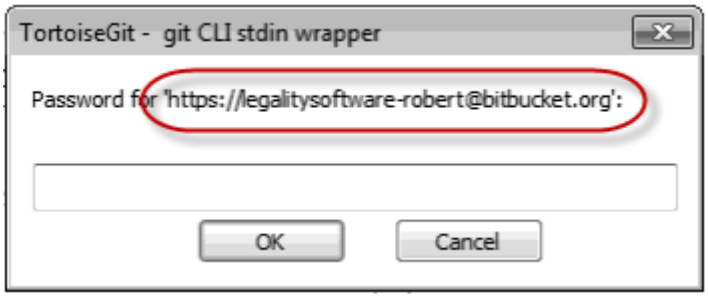




Change the user account



Now it will user Robert's account



Code How Tos

This site has some good quick tutorials: http://www.homeandlearn.co.uk/csharp/csharp_s4p9.html

Snippets

List all snippets by typing <Ctrl> <k>.<x>

<Alt><R><F> creates Properties for existing variables

Add the ones from this page: http://www.dotnetperls.com/snippet

prop
// create simple property

public int MyProperty { get; set; }

propfull
// create full property with private variable
private int myVar;

public int MyProperty

{

get { return myVar; }

set { myVar = value; }

}

propg<tab><tab>

// public get, private set

public int MyProperty { get; private set; }

cw
Console.WriteLine();

%%//%% Set Form to load on program Start - C#

// Look in “Program.cs” (in the 'Presentation' project)

static void Main()

{

> Application.EnableVisualStyles();
>
> Application.SetCompatibleTextRenderingDefault(false);
>
> Application.Run(new frmMain()); // This sets the form to load on program launch

}

NOTE: VB allows you to choose Form to load on program Start from a combobox in the Project Properties

%%//%% LinkLabel clicked launches http:%%//%%www.legalitysoftware.com in browser - C#

NOTE: Use RadLabel.Text MarkupEditor for a no code Hyperlink method

using System.Diagnostics; // required by Process.Start

private void linkLabel_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

Process.Start(“http://www.elaw.com/license.asp”);

}

%%//%% Open/Display a Form

private void rmiOptions_Click(object sender, EventArgs e) // menu click event

{

using (rfrmOptions rfrmOptionsInstance = new rfrmOptions()) // creates an instance of the form

{

rfrmOptionsInstance.ShowDialog(this); // then shows/opens it

}
}

In the example above the the using creates a new instance of the form each time is rmiOptions clicked. Here are 3 options to deal with this (typically you should pick the 3rd):

1. Create a new form instance each time then destroy the form instance when it is closed.
2. Instantiate the form at the top of the class (like this) then show/hide that one instance of the form.

> namespace LS.eLaw.Systray.Presentation
>
> {
>
> public partial class frmMain : Telerik.WinControls.UI.RadForm
>
> {
>
> rfrmOptions rfrmOptionsInstance = new rfrmOptions(); // creates an instance of Settings form
>
> public frmMain()
>
> {
>
> InitializeComponent();

```
<ol start="3" style="list-style-type: decimal;">
```

```
<li>
```

```
<p>
```

Check the currently open forms and only create if it doesn’t already exist (hide it on close) - **ONLY NON-MODAL!**

```
</p>
```

```
</li>
```

```
</ol>
```

private void rmiRules_Click(object sender, EventArgs e)

{

foreach (Form OpenForm in Application.OpenForms) // walk the collection of open forms

{

if (OpenForm.Name.ToString() == “rfrmRules”) // ...looking for this form by name

{

OpenForm.StartPosition = FormStartPosition.CenterScreen; // Open form in center screen

OpenForm.Show (this); // show/open it BUT ONLY FOR NON-MODAL WINDOWS

// OpenForm.ShowDialog(this); // DOES NOT WORK - ShowDialog creates modal which closes on Hide();

return;

}

}

using (rfrmRules rfrmRulesInstance = new rfrmRules()) // only create it if one doesn't already exist

{

rfrmRulesInstance.StartPosition = FormStartPosition.CenterScreen; // Open form in center screen

rfrmRulesInstance.Show (this); // show/open it BUT ONLY FOR NON-MODAL WINDOWS

// rfrmRulesInstance.ShowDialog(this); // DOES NOT WORK - ShowDialog creates modal which closes on Hide();

}

} // If you use this method to reuse the same form instance you should hide it when you close it (rather than destroy it)

%%//%% Open a form in the center of the screen, or parent window

form1.StartPosition = FormStartPosition.CenterScreen; // Open form in center screen

rfrmOptions.StartPosition = FormStartPosition.CenterScreen; // Open form in center screen

rfrmOptions.StartPosition = FormStartPosition.CenterParent; // Open centered on the parent form/window

using (rfrmAnalysisProgress rfrmAnalysisProgressInstance = new rfrmAnalysisProgress()) // creates an instance of the form

{

rfrmAnalysisProgressInstance.StartPosition = FormStartPosition.CenterParent; // center it on the parent

rfrmAnalysisProgressInstance.ShowDialog(this); // then shows/opens it

}

%%//%% Show, Hide and Close a form

NOTE: The form must exist before these will work so create it 1st per %%//%% Open/display a form

> Add this code to the click event of a main form button **(you MUST create it 1st!)**:

- RadFormChild.ShowDialog(); // Show the form as modal
- RadFormChild.Show(); // Show the form as not modal, which is typically a bad idea
- RadFormChild.Hide(); // Hide, but don't destroy this instance of the form. You can .Show() it again later.
- this.Hide(); // Hide current form without specifying its name

> You can add this code to the click event of a Child form button:

- Close(); // Closes 'this.' Form
- but the preferred method is to use the RadButton 'DialogResult' property, like this:

DialogResult = rbtnCancel.Text == "Cancel" ? DialogResult.Cancel : DialogResult.OK;

%%//%% Programmatically minimize a window

this.WindowState = FormWindowState.Minimized; // or 'Normal' or 'Maximized'

Or you can specify a form name:

frmMain.WindowState = FormWindowState.Maximized;

If you don't want to minimize an already minimized form

if (form_Name.WindowState != FormWindowState.Minimized) form_Name.WindowState = FormWindowState.Minimized;

%%//%% Access the Assembly for info like Product Name

~~The original/old way we accessed the Assembly~~

~~private string AssemblyProduct~~

~~{~~

~~get~~

~~{~~

~~// Get all Product attributes on this assembly~~

~~object[] attributes = Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyProductAttribute), false);~~

~~// If there aren't any Product attributes, return an empty string~~

~~if (attributes.Length == 0)~~

~~return “”;~~

~~// If there is a Product attribute, return its value~~

~~return [1]; SystemSettings.AppName = assemblyDataBuilder.GetAssemblyData().Product; Application.EnableVisualStyles(); Application.SetCompatibleTextRenderingDefault(false); Application.Run(new rfrmMain()); } // Grab the Product name from the Assembly so it is globally available as 'SystemSettings.AppName' // Stick this in Program.cs var assemblyDataBuilder = new AssemblyDataBuilder(Assembly.GetExecutingAssembly()); SystemSettings.AppName = assemblyDataBuilder.GetAssemblyData().Product; *[11:40:54 AM] Robert Gray: Should we use that? Or do you recommend something else? [11:41:02 AM] Marcel Popescu: If you want Product instead of Title then you do need to use that [11:41:11 AM] Marcel Popescu: The new code in common uses AppName = AssemblyData.Title [11:41:27 AM] Marcel Popescu: But if you're ok with using Title then you don't need to do anything in Program.cs anymore [11:42:20 AM] Robert Gray: I see. So I need to spend a few minutes with you working out exactly how to use that and what assembly properties end up where. Then I can document it and standardize our future projects. ====* // Close form/ Exit application ==== === Best Option === Use the Product Name from the Assembly NOTE: To access the Assembly info (for Product Name, etc), you must start by adding a Project Reference to: *‘X:\DevGIT\ls.common\LS.Common.Utilities\bin\Debug\LS.Common.Utilities.dll’* AND add using LS.Common; to any .cs file that requires access to the Assembly. using Telerik.WinControls; // required by RadMessageBox using System.Windows.Forms; // required by RadMessageBox button names using LS.Common; // required to access the Assembly **// Use the Product Name from the Assembly** private void rfrmMain_FormClosing(object sender, FormClosingEventArgs e) { // This is the place to check for dirty settings and offer to save before exit. // Ask the user if they want to exit. If not, cancel the FormClosing event. e.Cancel = RadMessageBox.Show(“Exit ” + SystemSettings.AssemblyData.Product + “?”, SystemSettings.AssemblyData.Title, MessageBoxButtons.YesNo, RadMessageIcon.Question) == DialogResult.No; } // NOTE: You must add: using Telerik.WinControls; // to use RadMessageBox() // Then close the main form to end the app… private void rmiExit_Click(object sender, EventArgs e) {
  Close(); // Close the current form (the app in this case)
  // FormClosing will pop confirmation dialog } <del>// If you do you'll need to get the Assembly info as the 1st thing inside 'public partial class rfrmMain : Telerik.WinControls.UI.RadForm {' and add ‘using System.Reflection; // To get the Assembly info’~~

> ~~// Grab Assembly info so the Product name is available to the 'rfrmMain_FormClosing()' event~~

~~private string AssemblyProduct~~

~~{~~

~~get~~

~~{~~

~~// Get all Product attributes on this assembly~~

~~object[] attributes = Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyProductAttribute), false);~~

~~// If there aren't any Product attributes, return an empty string~~

~~if (attributes.Length == 0)~~

~~return “”;~~

~~// If there is a Product attribute, return its value~~

~~return~~ [2] ~~{ rfrmRadAboutBox.ShowDialog(this); } } ==== // List all controls in the project to create spec doc ==== ??? Look at this page:~~ http://www.telerik.com/community/forums/winforms/forms-and-dialogs/getting-to-tooltips-from-rad control.aspx ~~==== // Use Tooltips with Telerik controls ==== * ‘ToolTipText’ – For a lot of controls you can just populate the ‘ToolTipText’ property and tooltips simply work. * Some controls like the RadButton don’t have a ‘ToolTipText’ property. In this case you have these options: * Get to the underlying ‘RadItem’ ToolTipText property via the ‘ButtonElement’ like this. See this page for more. radButton1.ButtonElement.ToolTipText = “hello”; > Or you can do the same in design time via the IDE. Just click the Telerik control “Tasks Button” | “Edit UI Elements” | and edit the ToolTipText there. Didn’t work * Drag the regular Windows (non-Telerik) toolTip control to any spot on the form (it will not be visible). The name doesn’t matter. It will add a ‘**ToolTip on toolTip**’ property to the Rad Controls that don’t already have a ‘ToolTip’ property. This control has some properties you can set for delays on appearance and disappeance. ToolTipManager ‘AutoCloseDuration’ – how long the tooltips remain onscreen ==== // TreeView - Expand All/Collapse All ==== This code will expand or collapse all nodes of a Treeview control. ??? ==== // Open a URL in the default browser ==== using System.Diagnostics; // Launch the web help page private void rmiHelp_Click(object sender, EventArgs e) { Process.Start(“http://~~www.legalitysoftware.com~~”); } Also use to open a local drive/folder. // Line Continuation for Long Multiline Strings (multiple lines) There are a few options to consider. Use the @ symbol in front of a string to form a~~ verbatim string literal ~~string query = @“SELECT foo, bar FROM table WHERE id = 42”; You~~ do not have to escape special characters ~~when you use this method, except for double quotes as shown here: string quote = @“Jon said, ““This will work,““ - and it did!”; The problem with using string literal is your code must be completely left aligned in order to avoid whitespace since all characters are included in the string itself. Note whitespace also means all \r\n newlines will be included in the final string. You can use String.Join like this to avoid the whitespace issue: var someString = String.Join(Environment.NewLine, “The”, “quick”, Environment.NewLine, // embedded new line “brown”, “fox…”); Another option is to use the + operator and break the string up into human-readable lines. The compiler will pick up that the strings are constant and concatenate them at compile time. Like this: const string myVeryLongString = “This is the opening paragraph of my long string. ” + “Which is split over multiple lines to improve code readability, ” + “but is in fact, just one long~~

~~string.”; Or this: var myString = String.Format(“hello ” + “world” + “ i am {0}” + “ and I like {1}.”, animalType, animalPreferenceType); // hello world i am a pony and I like other ponies. ==== // Enable/Disable Controls based on RadioButton/Checkbox settings ==== ### Add an entry here based on the code in the TreeDataExtracter project ==== // Multiple lines of text with new line (cr/lf) ==== > rtxtTextbox.Text = “Here’s a long string of text that will be…\r\n” + > > “…continued on the next line.\r\n” + > > “And wrap up here.”; ==== Create a Delay or Wait for some time ==== using System.Threading; Thread.Sleep(2000); // sleep for 2 seconds NOTE: completely blocks the current thread ==== Application Folder – Open Windows Explorer ==== // **Open the Messages subfolder in the app (exe) folder** System.Diagnostics.Process.Start(“explorer.exe”, Application.StartupPath + “\\Messages”); // **The @ ('string literal' character) escapes the ‘\’ so you don’t have to ‘\\’** System.Diagnostics.Process.Start(“explorer.exe”, Application.StartupPath + @“\\Messages”); // **Use Path.Combine(path1, path2) rather than 'adding' strings together** System.Diagnostics.Process.Start(“explorer.exe”, Path.Combine(Application.StartupPath, @“\Messages”)); // **Check if the \images folder exists, and delete or create it** string outputDir = System.IO.Path.GetDirectoryName(System.Windows.Forms.Application.ExecutablePath); DirectoryInfo df = new DirectoryInfo(outputDir + @“\images\”); if(df.Exists) // df.Delete(true); // directory exists, do you want to delete it? else DirectoryInfo di = Directory.CreateDirectory(outputDir + @“\images\”); // create new directory Not all of these methods will return the same value. They can return the same value, but be careful since their purposes are different: Application.StartupPath // return the StartupPath parameter (Can be set when run the app) System.IO.Directory.GetCurrentDirectory() // returns the current directory, which may or may not be the folder where the app is located, and is the same as Environment.CurrentDirectory. If using this in a dll, it will return the path of where the process is running (this is especially true in ASP.NET) ==== Control ‘Bring to Front’/’Send to back’ at runtime ==== // GetChildIndex method of the Controls collection gets the index of a particular child control. // index = 0 control is front most this.Controls.GetChildIndex(myControl); this.BringToFront(); // pull this control in front of other controls on the parent form ==== Create a Collapsible Region ==== Use #region to collapse a named section of code. Look for the <+>/↔ to the left of the #region line #region Any region name you want even with spaces private void InitializeMainGridViewDefaults() { this.rgvMainGridView.ShowColumnHeaders = true; #region Regions can also be inside a Class this.rgvMainGridView.AllowColumnReorder = true; #endregion this.rgvMainGridView.AllowColumnResize = true; this.rgvMainGridView.AllowColumnChooser = true; this.rgvMainGridView.AllowColumnHeaderContextMenu = true; } #endregion // Code continues outsdie the region… ==== Conditional Compilation with #define ==== **Define is in** // Conditional Compilation Flag declarations must be at the very top of the source file #define ReadOnlyWithGrouping // Since this is defined… using System; public class Class1 { public Class1() { #if ReadOnlyWithGrouping this.rgvMainGridView.ShowColumnHeaders = true; // these lines are included in compilation this.rgvMainGridView.AllowColumnReorder = true; #else this.rgvMainGridView.AllowColumnResize = true; // this line is grey and excluded from compilation #endif this.rgvMainGridView.AllowColumnChooser = true; this.rgvMainGridView.AllowColumnHeaderContextMenu = true; // Code continues outsdie the region… } } **Define is out** // Conditional Compilation Flag declarations must be at the very top of the source file // #define ReadOnlyWithGrouping // Now that this is NOT defined… using System; public class Class1 { public Class1() { #if ReadOnlyWithGrouping this.rgvMainGridView.ShowColumnHeaders = true; // these lines are grey and excluded from compilation this.rgvMainGridView.AllowColumnReorder = true; #else this.rgvMainGridView.AllowColumnResize = true; // this line is included in compilation #endif this.rgvMainGridView.AllowColumnChooser = true;~~

> ~~this.rgvMainGridView.AllowColumnHeaderContextMenu = true; // Code continues outside the region… } } ====== Base Template Projects ====== ===== LS.Base0 ===== Loads/saves a default settings file on application start/end. There is no option for the user to name the file or specify the path. The settings file is named '%applicationname%.dat' and is stored in the standard location: 'C:\ProgramData\Legality Software\%Application Name%'. Includes exception handling, licensing , setting storage. frmMain includes a RadDock, GridView in a Document pane and Detail PropertyGrid in a Tool Window. ===== LS.Base0.0 ===== Same as LS.Base0 but rfrmMain does not contain a RadDock or data controls. ===== LS.Base1 ===== Supports load/save of named settings files from/to a specified path. Includes exception handling, licensing , setting storage. frmMain includes a RadDock, GridView in a Document pane and Detail PropertyGrid in a Tool Window. ===== LS.Base1.0 ===== Same as LS.Base1 but rfrmMain does not contain a RadDock or data controls. ===== LS.Base3 ===== To be developed… Supports multiple queries like dbEntreé. Actually this will be a rewrite of dbEntreé but using controls that break up all the major functionality. Work with Marcel on where these breaks should be made. Supports load/save of named settings files from/to a specified path. Includes exception handling, licensing , setting storage. ==== Next thing… ====~~

1)

AssemblyProductAttribute)attributes[0]).Product;</del> } } **Now we can access the Assembly via the LS.Common.Utilities class** To access the Assembly info (for Product Name, etc), you must start by adding a Project Reference to: *'X:\DevGIT\ls.common\LS.Common.Utilities\bin\x86\Debug\LS.Common.Utilities.dll'* Then add using LS.Common; to any .cs file that requires access to the Assembly. // Change assembly information settings for your application here: // - Project | Properties | Application | <Assembly Information> using LS.Common; // required to access the Assembly static void Main() { // Grab the Product name from the Assembly so it is globally available as 'SystemSettings.AppName' var assemblyDataBuilder = new AssemblyDataBuilder(Assembly.GetExecutingAssembly(

2)

AssemblyProductAttribute)attributes[0]).Product;</del>

> ~~}~~
>
> ~~}~~

Other Options

// Simple, abrupt (no warning) form close/application exit private void rmiFileExit_Click(object sender, EventArgs e) { Close(); // Close the current form (the app in this case) } **// …AND - If you want to add a nice "Are you sure?" on application exit or form close, do this as well…** using Telerik.WinControls; // required by RadMessageBox using System.Windows.Forms; // required by RadMessageBox button names // Manually set the frmMain 'FormClosing' event to 'frmMain_FormClosing' private void rfrmMain_FormClosing(object sender, FormClosingEventArgs e) { if (RadMessageBox.Show("Exit TM Field Utility?", "TM Field Utility", MessageBoxButtons.YesNo, RadMessageIcon.Question) == DialogResult.No) e.Cancel = true; } // NOTE: You must add: using Telerik.WinControls; // to use RadMessageBox() ==== // 'Help Online' in web browser (Help menu) ==== private void rmiHelpOnline_Click(object sender, EventArgs e) { System.Diagnostics.Process.Start("http://www.legalitysoftware.com"); // Change the URL to launch product specific help on the Legality Software site } ==== // 'Legality Software Online' in web browser (Help menu) ==== private void rmiLegalitySoftwareOnline_Click(object sender, EventArgs e) { System.Diagnostics.Process.Start("http://www.legalitysoftware.com"); // launch the Legality Software site } You might want to check for errors per: http://msdn.microsoft.com/de-de/library/system.diagnostics.process.aspx Process myProcess = new Process(); try { // true is the default, but it is important not to set it to false myProcess.StartInfo.UseShellExecute = true; myProcess.StartInfo.FileName =

“http://some.domain.tld/bla”; myProcess.Start(); } catch (Exception e) { Console.WriteLine(e.Message); } ==== // About Dialog (Help menu) ==== See this RadAboutBox too private void rmiHelpAbout_Click(object sender, EventArgs e) { using (rfrmRadAboutBox rfrmRadAboutBox = new rfrmRadAboutBox(

From:
http://syncocoding.com/ - **My DokuWiki**

Permanent link:
http://syncocoding.com/doku.php?id=telerik_test&rev=1707271761

Last update: **2024/02/06 21:09**