



Make Your Apps Do More with Less Work

Visual Studio LightSwitch Technical White Paper

Author: Andrew Brust, Blue Badge Insights



Published: August, 2011

Applies to: Visual Studio LightSwitch 2011

Summary: This is the fifth in a series of white papers about Microsoft® Visual Studio® LightSwitch™ 2011, Microsoft's new streamlined development environment for designing data-centric business applications. In this paper, we discuss using LightSwitch extensions to bring extra features, connectivity, and UI sophistication to your applications.

Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2011 Microsoft Corporation. All rights reserved.

Microsoft, Microsoft® LightSwitch® 2011, Microsoft® Excel, Microsoft® SQL Server®, Visual FoxPro®, Visual Basic®, Microsoft® Windows® Azure™, are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

White Papers in this Series

1. [What is LightSwitch?](#)
2. [Quickly Build Business Apps](#)
3. [Get More from Your Data](#)
4. [Wow Your End Users](#)
5. [Make Your Apps Do More with Less Work](#)

Contents

- Introduction 5
- The Business Case for LightSwitch Extensions..... 5
- Themes and Shells 6
- Screen Templates..... 8
- Business Types 8
 - Corporate Business Types..... 8
 - Public Business Types..... 9
- Data Sources 9
 - RSSBus..... 10
- Custom Controls..... 12
 - DevExpress 12
 - Infragistics 14
 - First Floor Software Document Viewer..... 15
 - ComponentOne OLAP for LightSwitch..... 16
- How to Find Extensions..... 18
 - Extensions from Inside Visual Studio 20
- Installing and Using LightSwitch Extensions 21
- Conclusion..... 23

Introduction

This is the fifth in a series of white papers about Microsoft® Visual Studio® LightSwitch™ 2011, Microsoft's new streamlined development environment for designing data-centric business applications. In previous papers in this series, we explored the following:

- In the first paper, we provided an overview of LightSwitch, and an analysis of the market need it meets.
- In the second paper, we provided a walkthrough for creating important components of a LightSwitch application.
- In the third paper, we explored LightSwitch's data capabilities more deeply.
- In the fourth paper, we discussed the finer points of LightSwitch user interface design.

In this paper, we'll discuss using LightSwitch extensions to bring extra features, connectivity, and UI sophistication to your applications.

LightSwitch makes short work of much of the time-consuming coding that business applications require. LightSwitch is unique in its extensibility amongst business application productivity tools. It supports six types of extensions, helping ensure that LightSwitch developers can undertake a broad array of computation tasks, whether through extensions built by themselves, their IT departments, the LightSwitch community, and/or third party component vendors.

Not only can LightSwitch be extended through custom controls, but business types, data sources, screen templates, themes, and shells can be custom-built as well. This makes LightSwitch a productivity tool without roadblocks. When you want functionality beyond the product's current capabilities, you can get LightSwitch extensions through the Visual Studio gallery, either in your web browser or by using the Extension Manager right inside Visual Studio.

The Business Case for LightSwitch Extensions

LightSwitch's biggest asset is that it facilitates rapid development of business applications. The developers that LightSwitch attracts see the underlying software framework as a means to an end. More than the underlying technologies, LightSwitch developers are motivated by the application, its functionality, its usefulness, and its relevance to users.

Out of the box, LightSwitch serves well the scenario of the forms-over-data line of business applications. As LightSwitch developers acquaint themselves with the product's core capabilities, they grow accustomed to having them at their disposal. This facilitates wide adoption, and the efficiency it brings to the Microsoft development world is very beneficial to customers. As is always the case, developers want even more than what the core product provides. For example, LightSwitch developers may also want:

- data visualization
- web connectivity

- advanced data entry facilities
- professional reporting functionality
- screens, themes, and shells
- workflow
- content management
- credit card validation
- other eCommerce capabilities

When it's LightSwitch developers who want these capabilities, they probably don't want to write them from scratch or implement them through APIs.

LightSwitch developers are going to want the same capabilities available to core .NET developers, but they're going to want them without hassle. They'll want "grab-and-go" access to advanced functionality because it enables them to be more valuable to their customers. In short, they're going to want LightSwitch extensions.

For component developers, ISVs, and independent developers who want to make a contribution to the community, LightSwitch extensions represent a huge opportunity. Understanding the different types of extensions and seeing examples of each that are on the market now, will make the extensions concept and the value proposition more concrete. So in this last whitepaper in our series, we'll take a look at different kinds of extensions and how they can make LightSwitch applications shine.

Themes and Shells

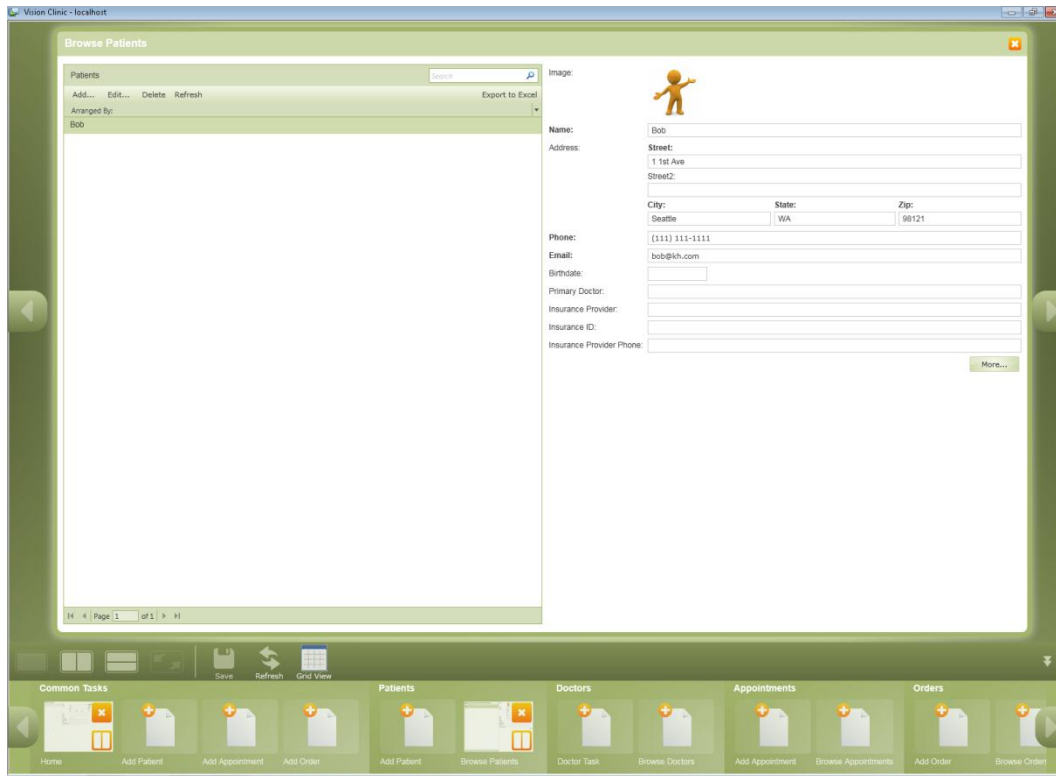
Since our last paper discussed the niceties of user interface and screen design, themes and shells seem like the logical starting point for our review of extensions. Each of these two extension types can bring real polish to your applications. Let's describe what each of these extension types does, how they can work together, and then look at an example of the visual impact they can have.

Themes are the extension type LightSwitch developers use to change the fonts and colors used in an application. If you're someone who works with web technologies, you could think of them as style sheets. Shells control the overall layout of your application. For example:

- where the navigation menu is displayed
- what the position of the application ribbon is
- how each of the items render

By using theme and shell extensions together, the look and feel of a LightSwitch application can be highly customized compared to the default appearance. **Figure 1** shows a LightSwitch application that uses both theme and shell extensions.

Figure 1 - A LightSwitch application using both theme and shell extensions



Let's take note of a few important customizations that appear in **Figure 1**:

- The appearance and position of the menu, rather than being text-based and off to the left, is graphical and along the bottom of the application window.
- The groups and individual screen links in the menu appear much differently than they would under the standard shell.
- The application ribbon appears below the active window area rather than above it.
- The standard **Save** and **Refresh** buttons appear with custom icons.
- There is a custom button labeled **Grid View** that has an icon as well.
- The active window has a Windows-standard looking caption bar, caption font, and close box rather than appearing with a tabbed interface.

Themes and shells assure that LightSwitch applications can be powerful and easy to develop without looking generic. They underlie a core theme: LightSwitch supports mainstream functionality out-of-the-box and allows for you, and third parties, to extend that functionality in specific ways. LightSwitch gives you what you need, and lets you add what you want.

Screen Templates

In previous papers in this series, you saw screen templates at work. LightSwitch has built-in templates that make it easy to implement many commonly required data maintenance screens in your LightSwitch applications:

- Details
- Editable Grid
- List and Details
- New Data
- Search Data

You also saw that templates don't limit you. You can use one to generate a "starter" screen and then modify it minimally or significantly.

Built-in templates are not the whole story however. Corporate IT and commercial vendors can provide screen templates as well. For example, when capturing in-house employee information, a particular screen template may be very useful. Alternatively, a screen template for capturing credit card information from the public might come in very handy. We could even imagine screen templates that use the Modal Window group as a container for these data capture templates with some UI on the main screen to display the popup.

Custom control extensions may also incorporate screen templates to enable LightSwitch developers to quickly generate screens that use those controls. In the Custom Controls section of this paper, we examine one such screen template and mention a second one.

Business Types

In previous papers in this series, we discussed the Money, Phone Number, and Email business types. Business types bring a combination of formatting, validation logic, and property options that add a lot of value and make LightSwitch data models even smarter than they would be if they worked only with base data types.

Developers can create their own business types, and this opens a wide array of productivity possibilities. There are a couple of scenarios where business types are especially compelling. One of these is for internal corporate development, the other for more broad use scenarios.

Corporate Business Types

Imagine a retail company with its own product ID scheme, where the IDs have a certain format. There may also be rules for maximum numeric values allowed within sections of the ID or a range of alpha characters that are permitted or required. One can imagine a number of LightSwitch applications developed by various business units throughout this organization. This company's IT department could develop a product ID business type extension and ask all business units to use it. This would ensure valid data across the company. It would also eliminate a lot of work for LightSwitch developers within

the individual business units and/or reduce fees paid to consultants implementing LightSwitch solutions. Even if this company used standard UPC codes for product IDs, a business type could still be used. For example, a business type could ensure that only codes corresponding to items sold by the company are entered into the entity fields to which it is applied.

Public Business Types

Another scenario involves a software company developing business types for specialized, but broadly used, data. This could include business types such as:

- postal codes for specific countries
- geocoding (latitude/longitude) coordinates
- airport or airline codes
- states or provinces for a particular country or set of countries
- Standard Industry Classification codes
- a host of other standardized data types with specific allowable values or value ranges

No matter whether for corporate or public data, LightSwitch business types make it easy to ensure valid data is entered and that the work involved in ensuring that is done once, rather than being duplicated for each application. If you think about it, LightSwitch business types serve both as productivity tools and data governance tools.

Data Sources

Databases that are designed and created in LightSwitch use Microsoft® SQL Server® or SQL Azure and LightSwitch can also connect to existing SQL Server and SQL Azure databases, as well as Microsoft® SharePoint® 2010 lists. LightSwitch also allows you to connect to your own .NET WCF RIA Services. Such WCF RIA Services can optionally be packaged as extensions, making their installation and use in a project quite straightforward. For data sets with known schema in corporate environments, data source extensions can be extremely useful.

Data source extensions will work extremely well for in-house use. An IT department can build a RIA Service around data from a particular application, enabling it to be shared, analyzed, and even updated in LightSwitch applications. Data source extensions may also be packed as companions to other extensions types. For example, the DevExpress XtraReports control that we will examine later in this paper provides its own data source extension to permit report preview screens to communicate with the report server.

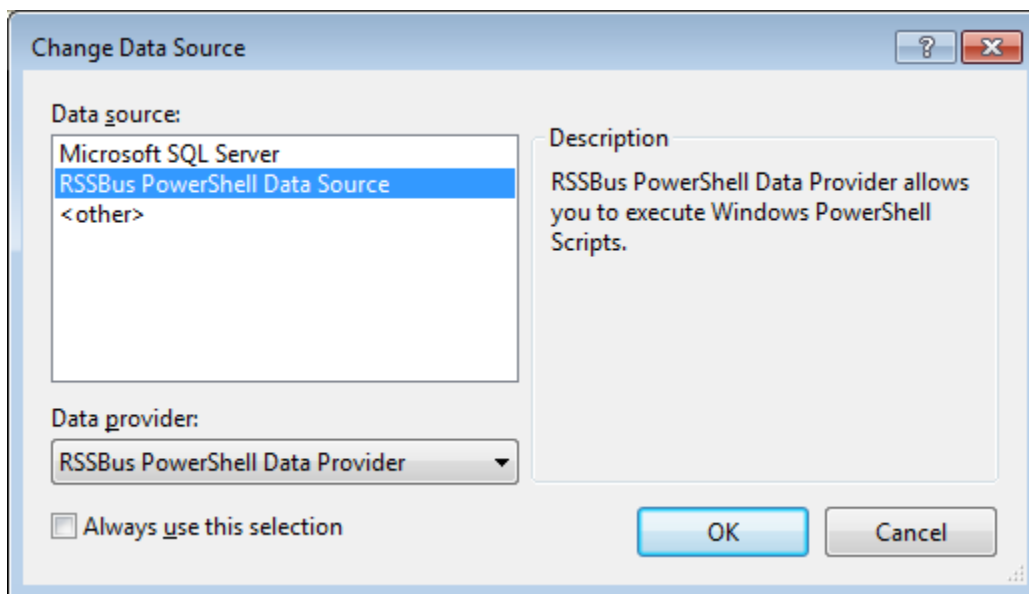
For more general structured and relational data sources, LightSwitch also allows using compatible ADO.NET providers with Entity Framework support. This allows even non-traditional data to appear as standard LightSwitch data entities and makes it really easy to build applications around such data. Let's look at one such provider now.

RSSBus

Microsoft partner RSSBus takes the compatible Entity Framework provider concept and extends it to data sourced from the web and social media, as well as other data sources. This allows LightSwitch applications to connect to data from Google, Salesforce, Quickbooks, Facebook, Twitter, and even Windows PowerShell scripts. Best of all, once connected, such data can be used by the LightSwitch developer in exactly the same manner as data from SQL Server or SharePoint. Controls bind to the entities and fields using exactly the same screen design techniques we've discussed in previous papers in this series, and the process of connecting to the data itself is very similar to connecting to an existing SQL Server database.

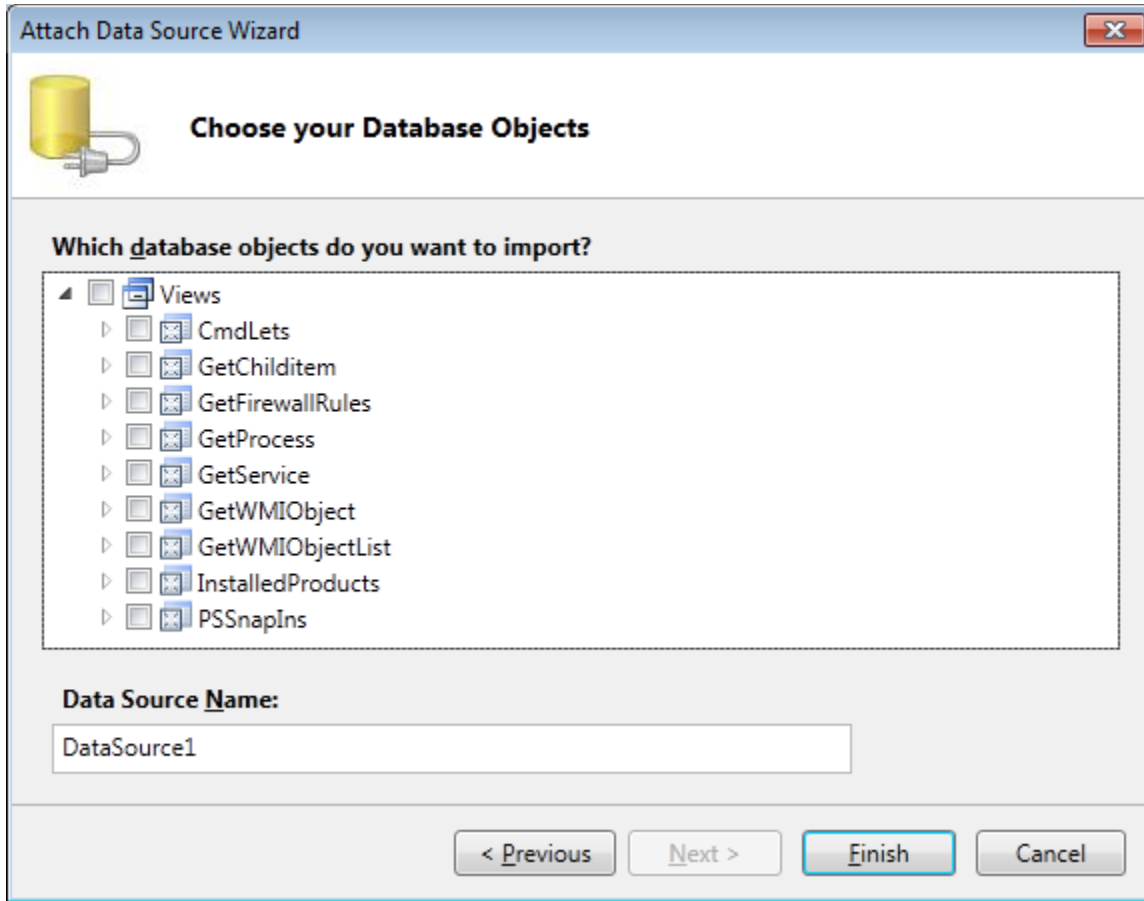
The **LightSwitch Attach Data Source Wizard** uses the standard Windows data link dialog boxes and lets you select the installed provider that you want to use in your application. This is shown in **Figure 2**.

Figure 2 - Selecting the RSSBus PowerShell Data Source provider from a standard data source selection dialog



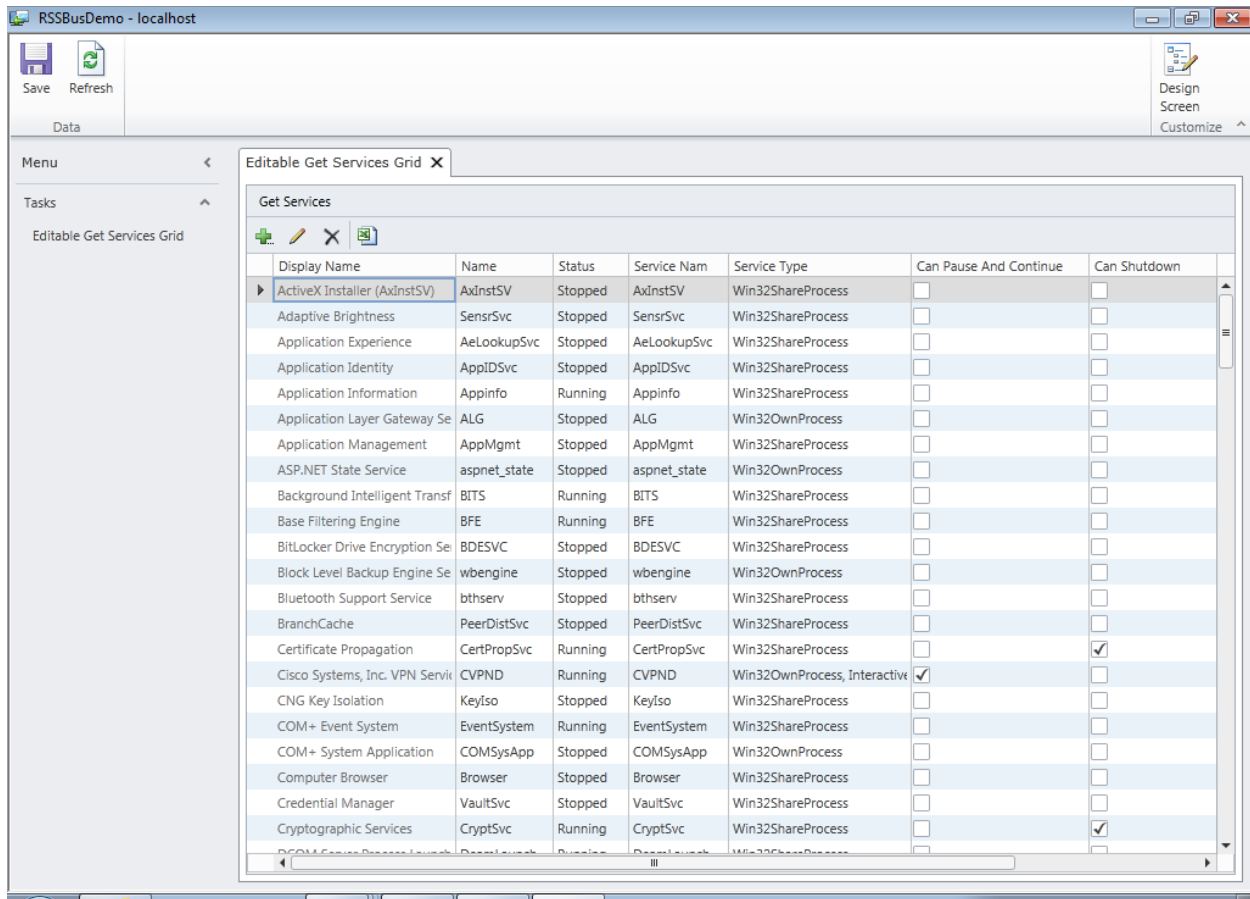
Once a provider is selected and a connection configured, the data source is inspected by LightSwitch just as would be any other. This allows the developer to select specific data, which appear as tables and views, from the data source. **Figure 3** shows the **Attach Data Source Wizard** and lists the data available using the RSSBus PowerShell Data Source provider. Each of the PowerShell script-provided data types is presented as a view and can be selected as an entity for the LightSwitch application's data model.

Figure 3 - Using an RSSBus provider to connect to PowerShell data



Once the data source is added, it's easy to create screens around it. **Figure 4** presents an example showing a LightSwitch *Data Grid* bound to a query of a PowerShell script that retrieves a list of running Windows services on the host machine.

Figure 4 – A screen with a *Data Grid* bound to a PowerShell script providing a list of running services



Custom Controls

LightSwitch may have six extension types, but perhaps the most important type is the custom control extension. From their name, custom controls may sound like they simply deliver extra user interface/presentation layer capabilities. But in fact, custom controls serve as packages of broad functionality. A reporting control doesn't just display reports, it generates them. An analysis control doesn't just display a cross tab view of data, but provides filtering and aggregation capabilities against data. Really, custom controls are like whole applications within your application. But since they're embedded and integrate with your user interface, they just make your application look great, instead of stealing its thunder.

Partners including Infragistics, ComponentOne, DevExpress, and First Floor Software have LightSwitch custom controls that add significant capability and visual sophistication to LightSwitch applications. Let's take a look at some of them.

DevExpress

We mentioned in a previous paper in this series that LightSwitch offers no integrated database reporting capabilities beyond its export to Excel function. We also explained that SQL Server Reporting Services

could be used against a LightSwitch application's database, but that authoring and rendering of SSRS reports is external to LightSwitch.

By contrast, DevExpress offers its XtraReports solution for integrated reporting in LightSwitch. With XtraReports, the report designer sits right in Visual Studio with your LightSwitch project. A single line of code in a LightSwitch screen's custom button loads the report. **Figure 5** shows how such custom buttons might appear in the user interface.

Figure 5 - A standard LightSwitch application with custom buttons on the ribbon to load XtraReports reports

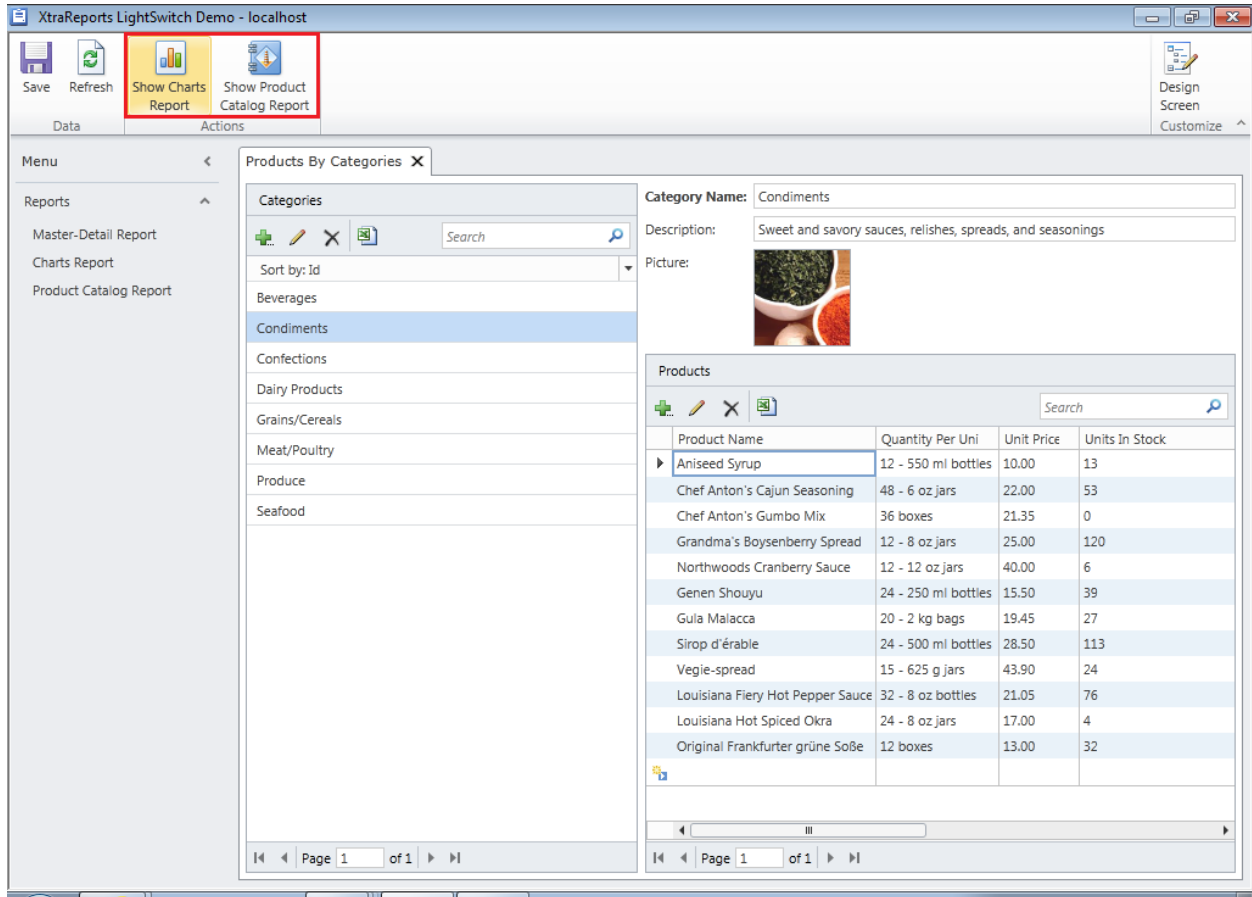
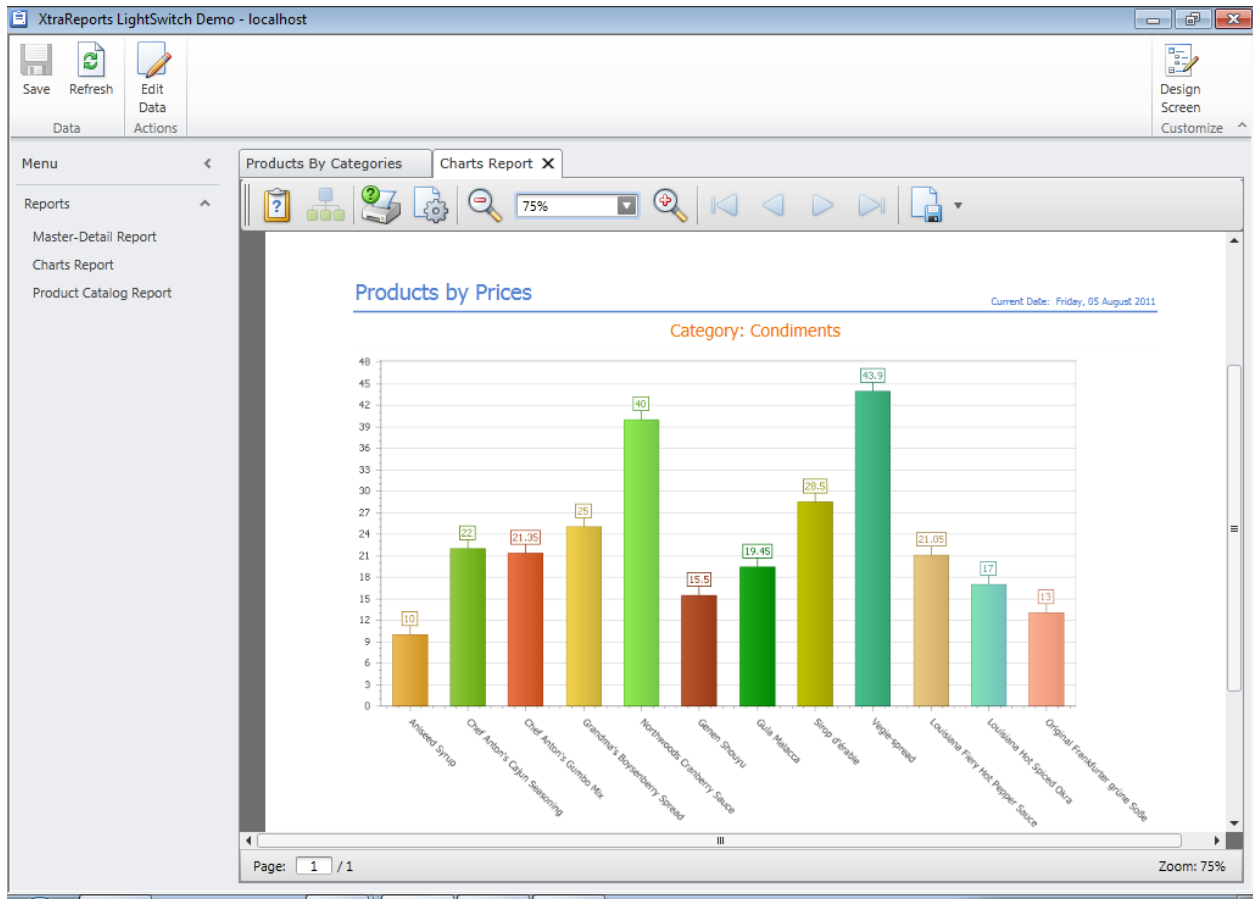


Figure 6 shows how a rendered XtraReports report might look.

Figure 6 – An XtraReports preview screen, with a rendered report displayed.

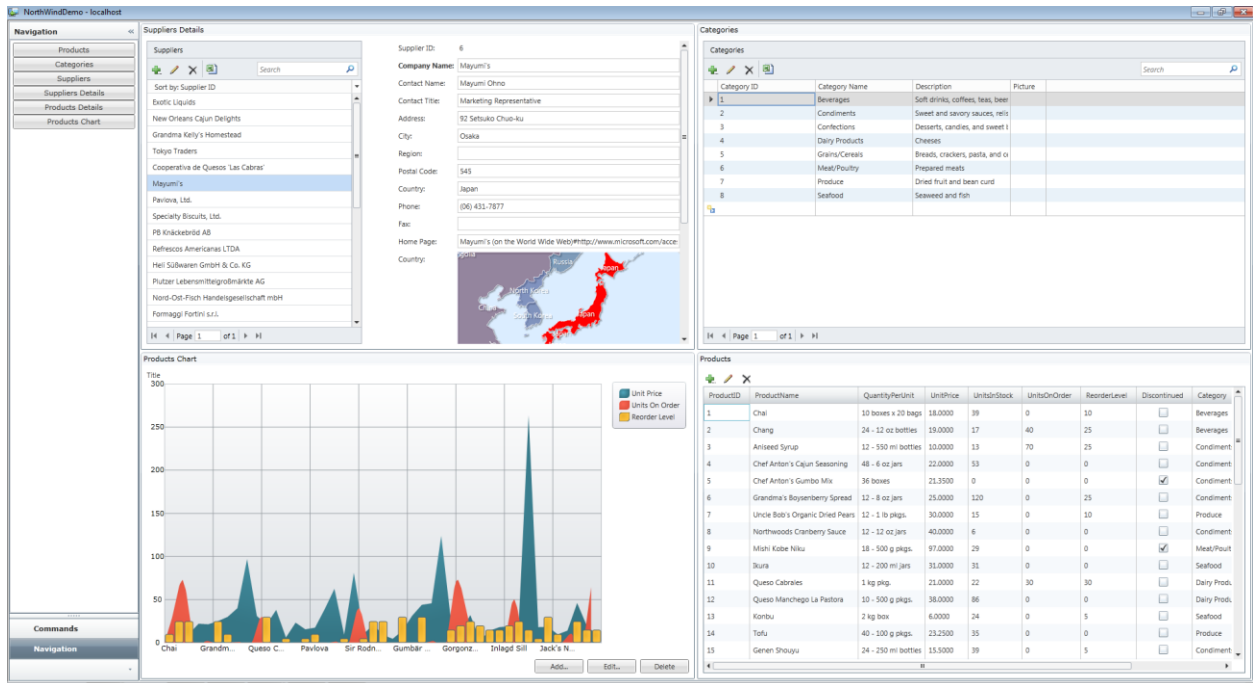


Notice how the report preview control is completely embedded in the LightSwitch application such that it looks as if it were created by the developer herself. A click of the ribbon button from the standard data screen shown in **Figure 5** launches this screen in a completely fluid, seamless fashion. This is the hallmark of LightSwitch custom control extensions: extra functionality for developers without disruption.

Infragistics

Infragistics, in addition to themes and shells, offers number of LightSwitch controls to help with everything from masked data input to data visualization. **Figure 7** shows a LightSwitch application with a combination of an Infragistics shell, map control, and chart control.

Figure 7 - Infragistics shell, map control and chart control



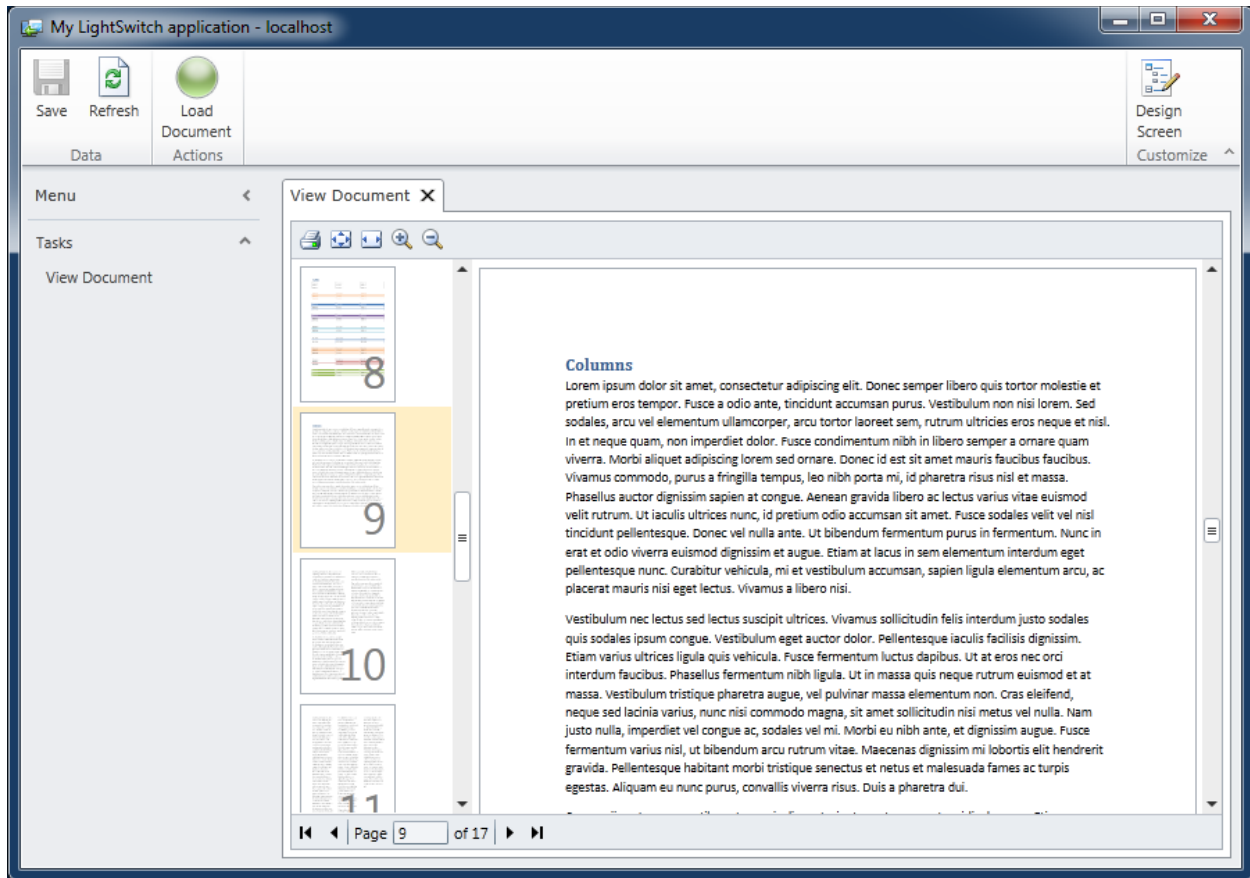
In addition to the two controls, notice the Outlook-style rendering of the menu and the tiled layout of the screens. Each of these UI customizations is accomplished by using of the Infragistics shell in the application.

This example demonstrates really well how the various LightSwitch extension types can be used in combination to great effect. In this case, controls and a shell are combined to deliver a differentiated overall user interface, plus advanced data visualizations within specific screens.

First Floor Software Document Viewer

Just as real-world business applications sometimes require reports, others require working with documents. First Floor Software's document viewer allows LightSwitch developers to integrate documents of various formats, including Acrobat/PDF, right in their applications. **Figure 8** shows the control integrated into a LightSwitch application, displaying one page of a PDF document and thumbnails for a range of other pages.

Figure 8 - First Floor Software's Document Viewer control, displaying a PDF document, in a LightSwitch application

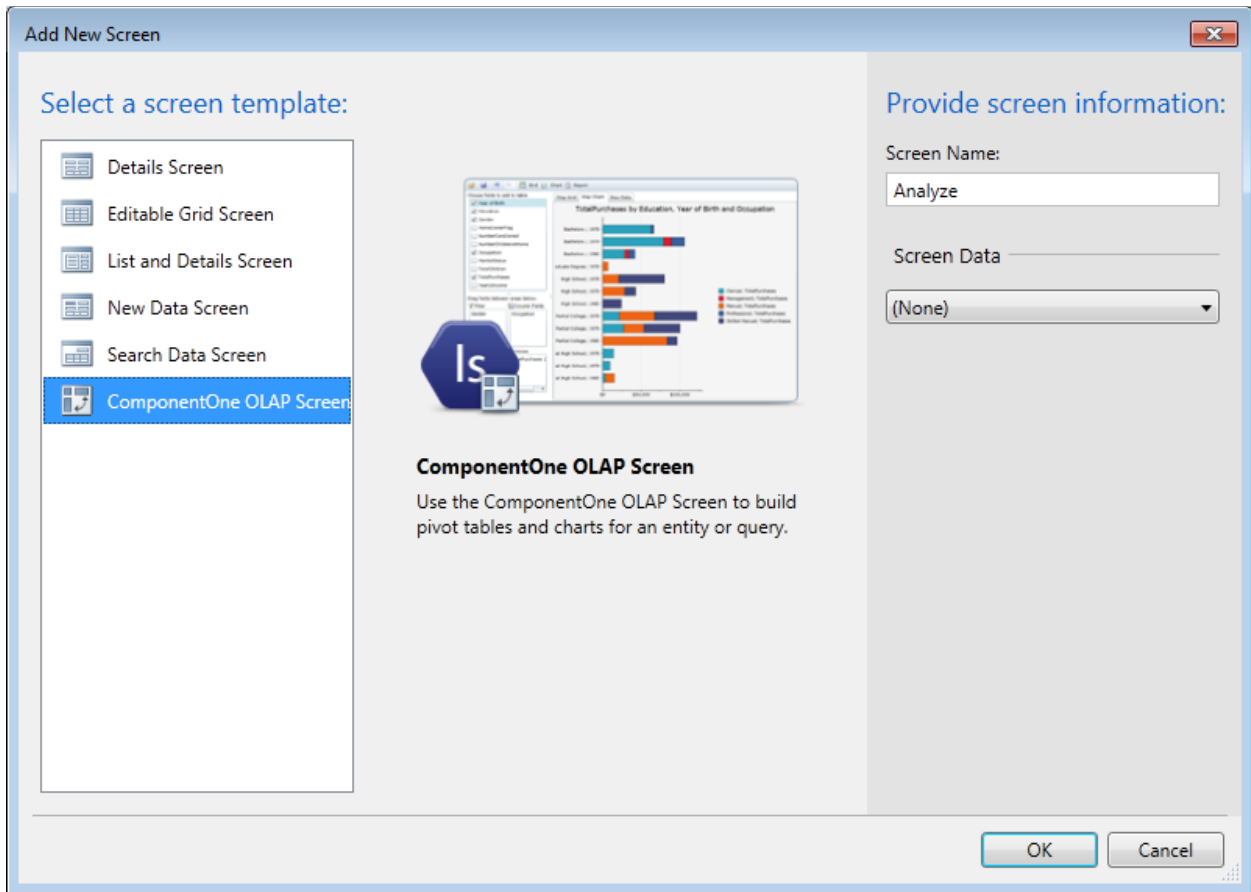


Notice that once again, a custom button (in this case the **Load Document** button to the right of the **Refresh** button) in the ribbon helps integrate the custom control's functionality into the application's user interface. Another thing to appreciate is that while the example in **Figure 8** shows the custom control being used with LightSwitch's standard shell and theme, developers could combine it with shell and theme extensions, even those developed by vendors other than the control vendor.

ComponentOne OLAP for LightSwitch

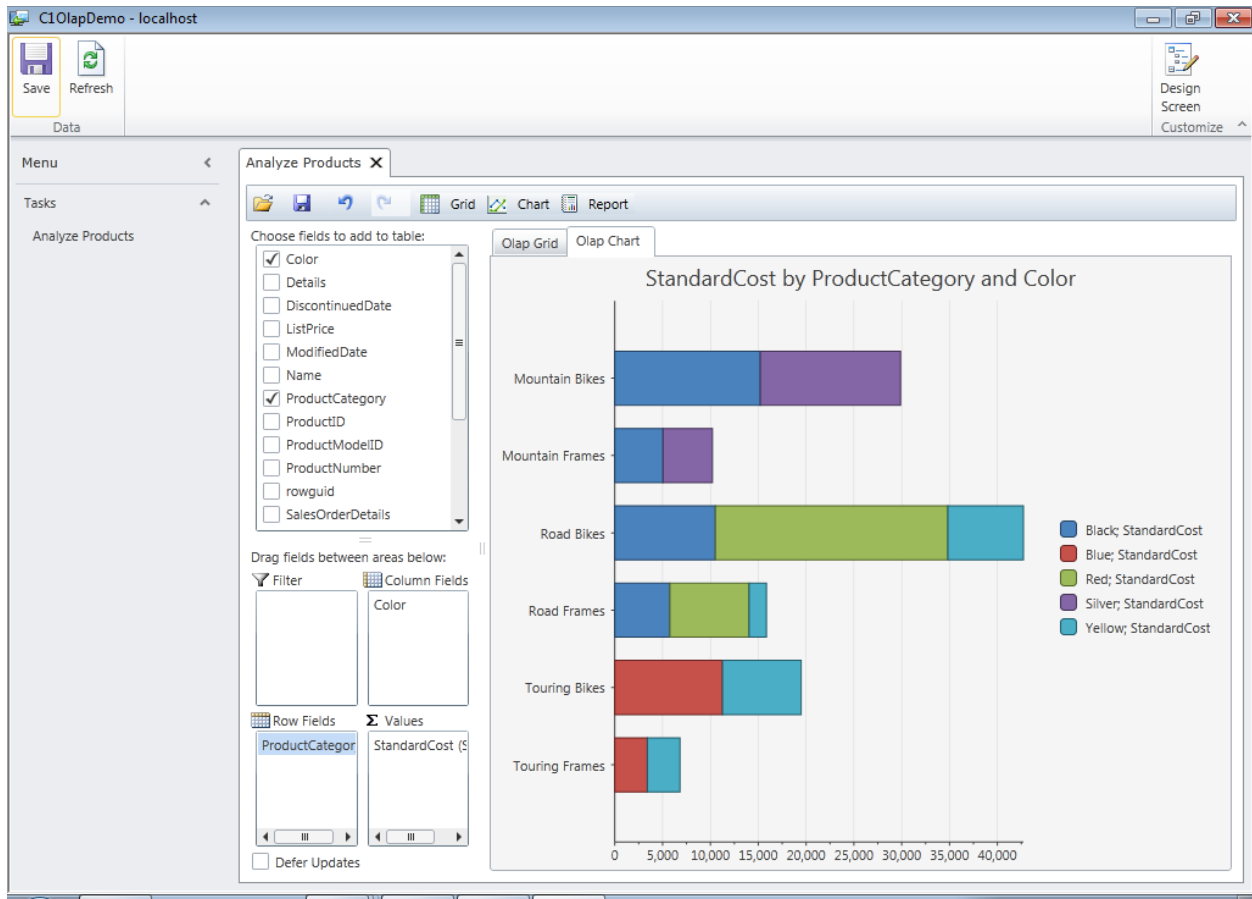
Data visualization may yet be a side specialty for Visual Studio LightSwitch, if the emerging custom controls are any indication. We've already seen such capabilities provided by DevExpress and Infragistics. Another example is ComponentOne's OLAP (OnLine Analytical Processing) for LightSwitch control. This control provides functionality similar to the Microsoft® Excel® popular PivotTable and PivotChart features. With OLAP for LightSwitch, a developer needs only install the extension, select a specific data entity or query at which to point the control, and then create a new screen using the accompanying ComponentOne OLAP screen template. Use of the screen template is shown in **Figure 9**.

Figure 9 - Creating a ComponentOne OLAP screen using the screen template of the same name



Once the screen is generated, the control is sized to fill the screen's client area. When the application is run and the screen is loaded, specific fields can be dragged and dropped onto the **Values**, **Row Fields**, **Column Fields**, or **Filter** panes. A hierarchical crosstab grid and an accompanying chart in an adjacent tab are generated. The **Olap Chart** tab is shown in **Figure 10**.

Figure 10 - The ComponentOne OLAP for LightSwitch control's Olap Chart tab



In **Figure 10**, notice the OLAP for LightSwitch control has a toolbar. The buttons on it allow the views created in the control to be opened and saved. There is also support for undo, re-do, and rendering the view as a print-ready report. The **Grid** and **Chart** buttons provide options menus for each of those two renderings.

The ComponentOne OLAP control is a specific example of a custom control extension and a screen template extension being packaged together to make the control more productively useable by the LightSwitch developer. The DevExpress XtraReports product does likewise, although we did not show its screen template.

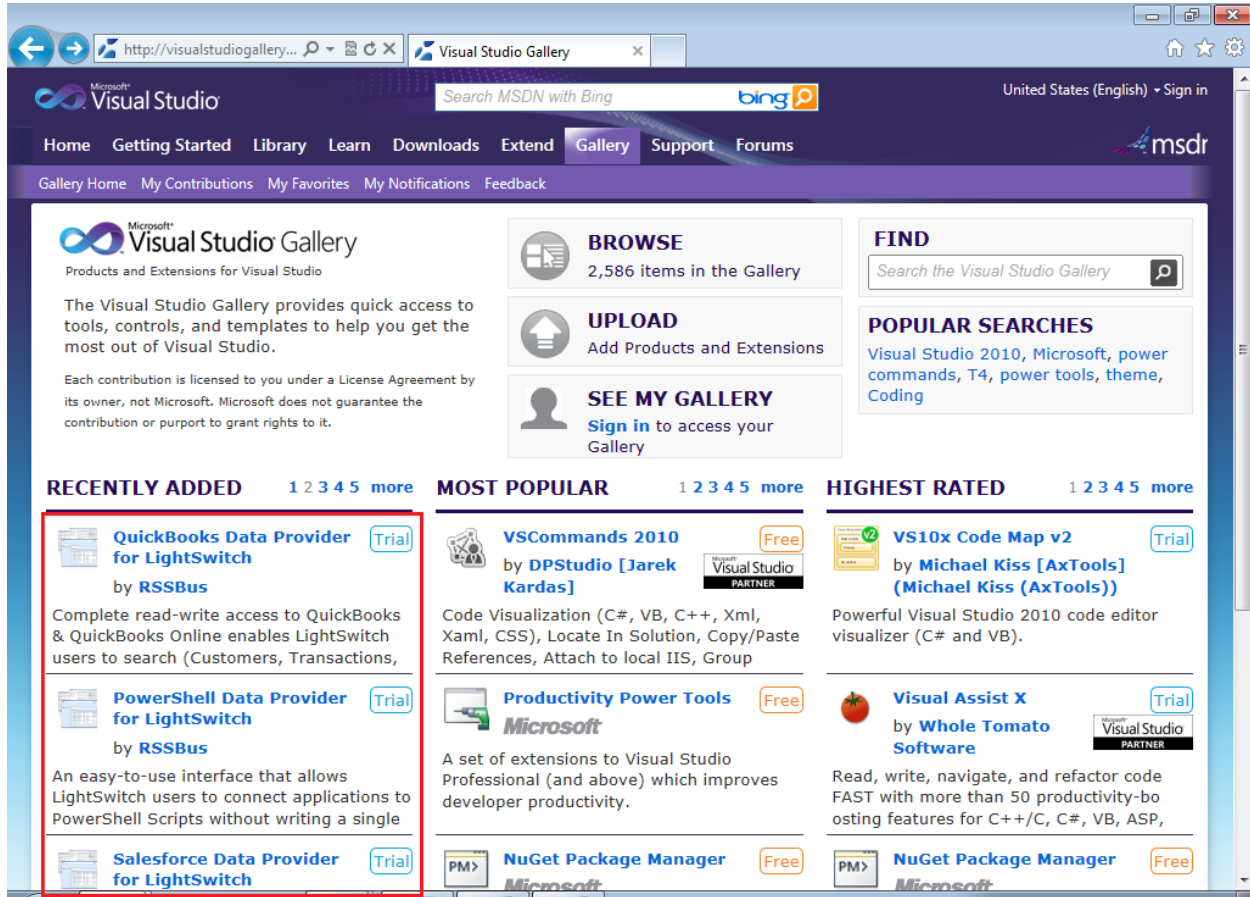
How to Find Extensions

One of the best things about extensions is how easy they are to find. While extensions can be developed privately and deployed through the same means as any setup package, including by email, most commercial and community extensions can be found in the Visual Studio Gallery, a marketplace for various Visual Studio extensions, including those for LightSwitch.

There are two ways to get to the Visual Studio Gallery. One way is through a web browser. Simply navigate to <http://visualstudiogallery.msdn.microsoft.com/> and you'll be able to search, read about,

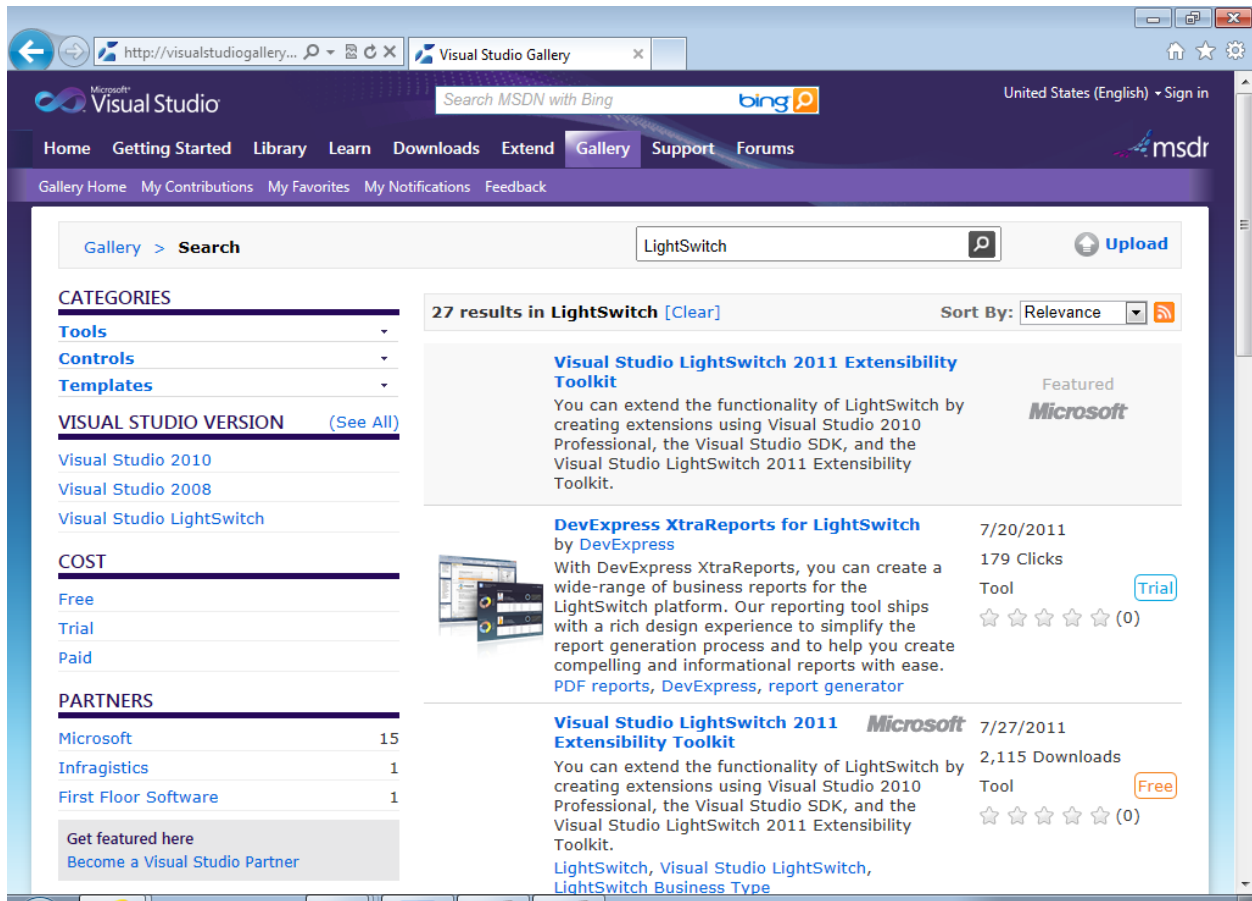
purchase (if applicable), and download from a wide array of extensions. **Figure 11** shows the Visual Studio Gallery home page in Internet Explorer, with a few RSSBus products highlighted.

Figure 11 - The Visual Studio Gallery on the web, with certain RSSBus products for LightSwitch highlighted



Use the **FIND** search box to search for LightSwitch extensions. An example results page for that search is shown in **Figure 12**.

Figure 12 - A Visual Studio Gallery results page for the search phrase "LightSwitch"

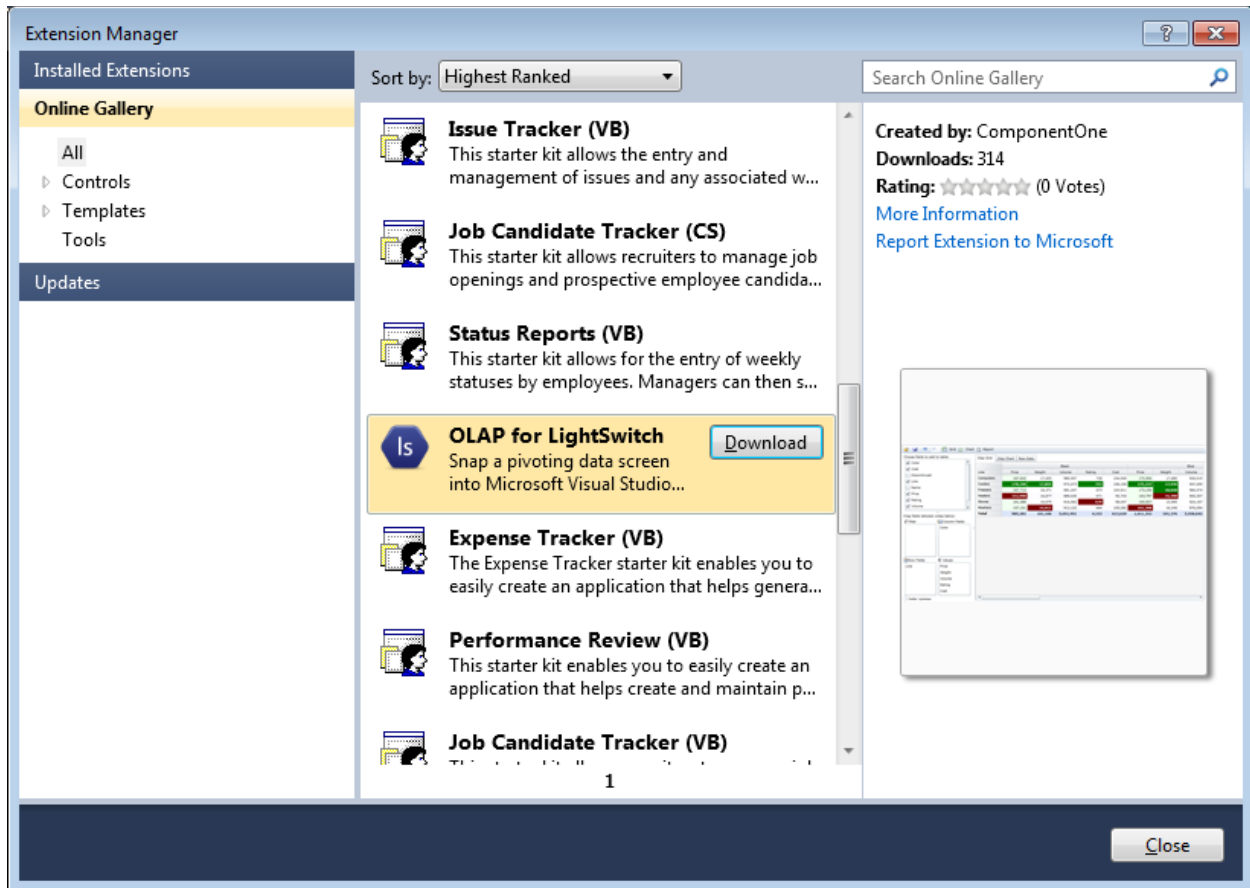


Extensions from Inside Visual Studio

A more integrated way to get to the Visual Studio Gallery is to do so from inside Visual Studio:

1. Double-click the **Properties** node in **Solution Explorer**.
2. Click the **Extensions** tab.
3. Click the **Browse for more extensions online** link (at the bottom) to browse the Visual Studio Gallery from right inside your project.
The **Extension Manager** window opens.
4. Click **Online Gallery**.
The result is shown in **Figure 13**.

Figure 13 - The Visual Studio Extension Manager window, displaying LightSwitch extensions

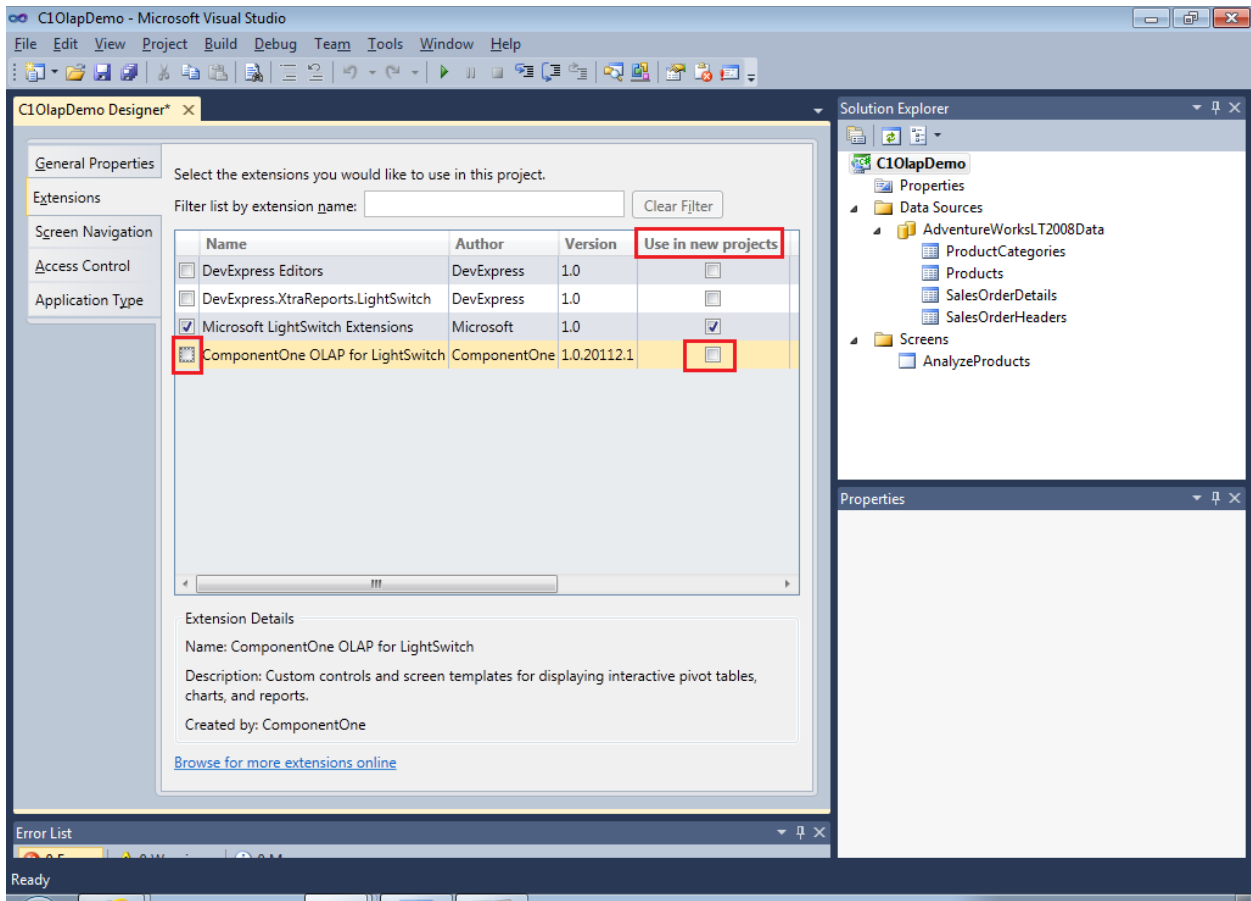


Regardless of whether you're using your browser or the **Extension Manager** window in Visual Studio, you can select the extension in which you're interested and click the **Download** button. From there you may be taken to a webpage to complete your purchase and download. Sometimes you have the option of installing a trial version instead of a paid edition. Installing trial editions is a great way to test out an extension if you're still unsure of a purchase. You can select the edition you want, and then start the installation.

Installing and Using LightSwitch Extensions

Once the installation is complete, restart Visual Studio (manually or by using the **Restart Now** button that may appear in the **Extension Manager** window), and the main grid in the **Extensions** properties page should display your extension in its own row. This is shown for ComponentOne OLAP for LightSwitch in **Figure 14**.

Figure 14 - The Extensions properties page, with a newly-installed custom control extension displayed



To use the extension in the current project, select the associated option (as shown highlighted in the leftmost column in **Figure 14**). To have the extension added by default to all future projects, select the option in the **Use in new projects** column. To use themes and shells, select them from the **Shell** and **Theme** drop-down lists in the **General Properties** tab.

Conclusion

Throughout this series of whitepapers, we've explored the high productivity of LightSwitch and we've discussed how that productivity is accompanied by power and extensibility. In this paper, we made the extensibility of LightSwitch a concrete value proposition. Microsoft has laid the groundwork for enterprise developers, community contributors, and independent software vendors to extend a product that is already compelling, but which leaves room for the addition of significant value. By creating a development environment centered on the definition of data and screens, through a process in which extensions directly integrate, Microsoft has assured that LightSwitch developers are pre-trained in the use of these extensions in their own work. LightSwitch developers are predisposed to the ready-made functionality extensions can provide, as it can enable them to enhance their applications while staying focused on rapid development and delivery. The extension development opportunity is huge, especially so early in LightSwitch's life.

For more information:

Visual Studio LightSwitch Website: <http://www.microsoft.com/lightswitch>

Visual Studio LightSwitch Dev Center: <http://msdn.microsoft.com/lightswitch>