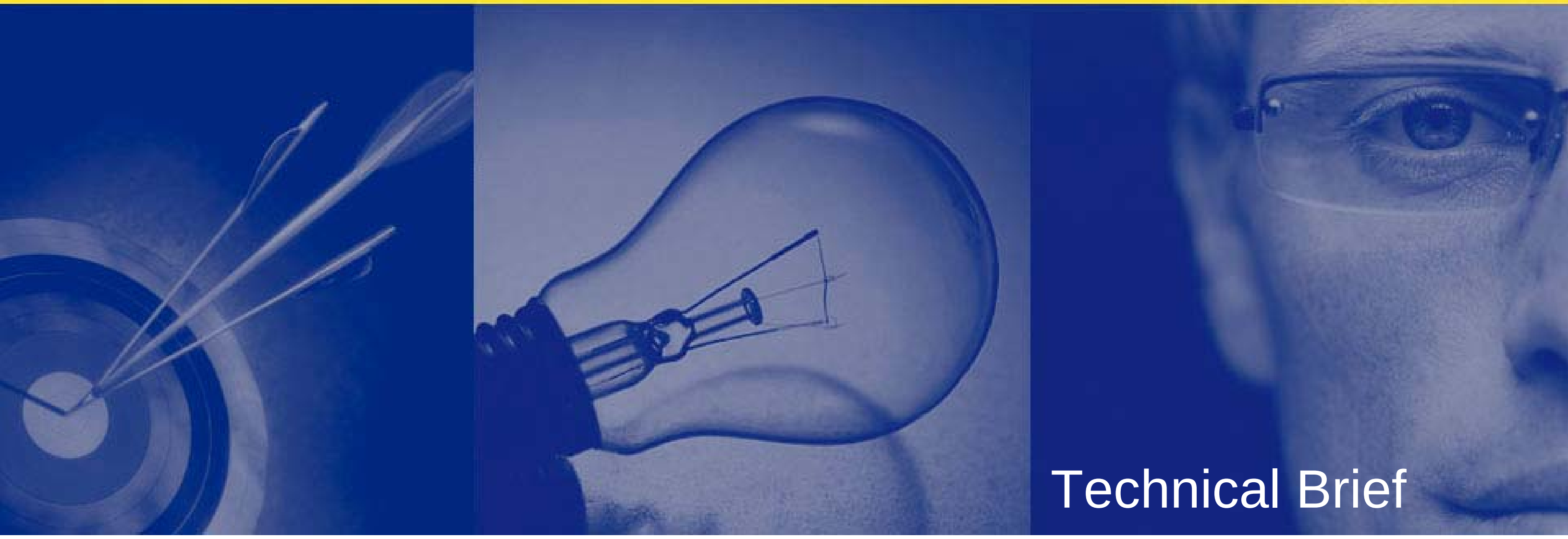
Getting the Most From SharePoint Through Intuitive Application Development Tools

Written by
Dan Kruger
Senior Product Manager
Quest Software, Inc.



Technical Brief

WARRANTY

The information contained in this document is subject to change without notice. Quest Software makes no warranty of any kind with respect to this information. QUEST SOFTWARE SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTY OF THE MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Quest Software shall not be liable for any direct, indirect, incidental, consequential, or other damage alleged in connection with the furnishing or use of this information.

TRADEMARKS

World Headquarters
5 Polaris Way
Aliso Viejo, CA 92656
www.quest.com
e-mail: info@quest.com
U.S. and Canada: 949.754.8000

Please refer to our Web site for regional and international office information.

Updated—May 29, 2008

CONTENTS

ABSTRACT

Organizations adopting SharePoint quickly recognize its potential. In fact, many are thinking beyond using SharePoint just as a document repository and a place for team sites. They are considering SharePoint for:

- Hosting simplified interfaces for their business applications
- Reworking applications based on aging technologies
- Building new applications as an alternative to .NET

But getting the advanced functionality it promises can take weeks of custom coding using native tools, leaving some to question their initial investment. Quest Development Studio for SharePoint helps overcome this by enabling organizations to quickly and effectively build the applications they need.

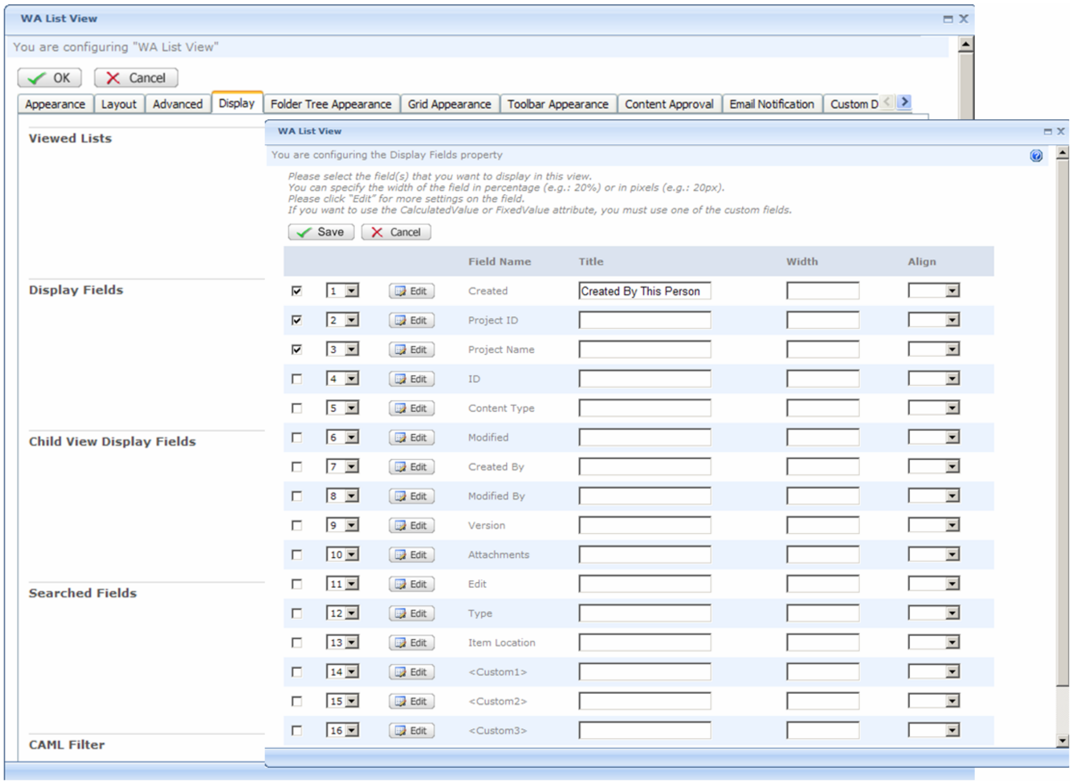
GETTING MORE FROM YOUR SHAREPOINT INVESTMENT

Organizations adopting SharePoint quickly recognize its potential. But getting the advanced functionality it promises can take weeks of custom coding using native tools, leaving some to question their initial investment. For example, SharePoint does not allow users to automatically pull data from lists into a centralized list or report, nor does it allow parent-child relationships. These functionalities would allow users to quickly roll-up or create inheritance relationships for data. Since these seemingly basic functions require coding, many users abandon building applications in SharePoint because they just don’t have the expertise to get it done.

Quest Development Studio for SharePoint makes it possible to quickly and easily build applications that use data from SharePoint and external systems. Development Studio also allows SharePoint power users and developers alike to get the functionality they require in SharePoint without having to deal with custom coding, saving significant time and money when building applications in SharePoint.

What is Development Studio?

Microsoft provides a set of web parts that do not expose all of the capabilities of SharePoint. Fully utilizing the capabilities of SharePoint requires additional web parts. Development Studio is an entire collection of additional web parts needed for sophisticated application development. In addition to making new SharePoint capabilities available, these web parts are easy to use allowing you to configure rather than code. The configurator interface shown here simplifies application building.



Please select the field(s) that you want to display in this view.
You can specify the width of the field in percentage (e.g.: 20%) or in pixels (e.g.: 20px).
Please click "Edit" for more settings on the field.
If you want to use the CalculatedValue or FixedValue attribute, you must use one of the custom fields.

			Field Name	Title	Width	Align
☑	1	Edit	Created	Created By This Person		
☑	2	Edit	Project ID			
☑	3	Edit	Project Name			
□	4	Edit	ID			
□	5	Edit	Content Type			
□	6	Edit	Modified			
□	7	Edit	Created By			
□	8	Edit	Modified By			
□	9	Edit	Version			
□	10	Edit	Attachments			
□	11	Edit	Edit			
□	12	Edit	Type			
□	13	Edit	Item Location			
□	14	Edit	<Custom1>			
□	15	Edit	<Custom2>			
□	16	Edit	<Custom3>			

Figure 1. Development Studio's configurator interface simplifies application development in SharePoint.

DEVELOPMENT STUDIO OVERCOMES THE LIMITATIONS OF NATIVE SHAREPOINT

Let’s take a look at three of the most pressing business needs that expose the limitations of native SharePoint:

- Connecting to external data
- Aggregating information from multiple sites into a dashboard
- Establishing a parent-child relationship between lists

Example 1: Connecting to External Data

At a large corporation, it’s budget time. Mark, a business analyst in the operations department, just found out that his manager wants a new budget process. This year, the manager wants one central location where she can not only view the budget, but manipulate and edit the budget data live. Mark knows from experience that the alternative is getting separate budget spreadsheets, pulling them into one file, and e-mailing them around for reviews and edits.

The team has started to use SharePoint for document collaboration, and it seems like a logical place to manage the budget information requested.

Doing the Project with Native Tools

A quick web search shows Mark he should be able to view data into SharePoint using something called the Data Viewer Web part. This web part would let him view data, but he doesn’t have the weeks required to learn how to use it. Further, he wouldn’t be able to manipulate the data, which is one of his manager’s requirements.

Another option is the Business Data Catalog (BDC). Mark quickly discovers that it is much more complex than it appeared, and the data is still read-only. Moreover, because Mark uses Windows SharePoint Services (WSS) v.3, this tool is not available: the BDC is only included in Microsoft Office SharePoint Server (MOSS). For this project, purchasing MOSS is not a justifiable expense.

Mark is discouraged until one of his team members mentions Quest Development Studio for SharePoint. Development Studio connects to data in SQL and allows the team to view and edit data that resides in SQL through the SharePoint user interface.

Doing the Project using Quest Development Studio

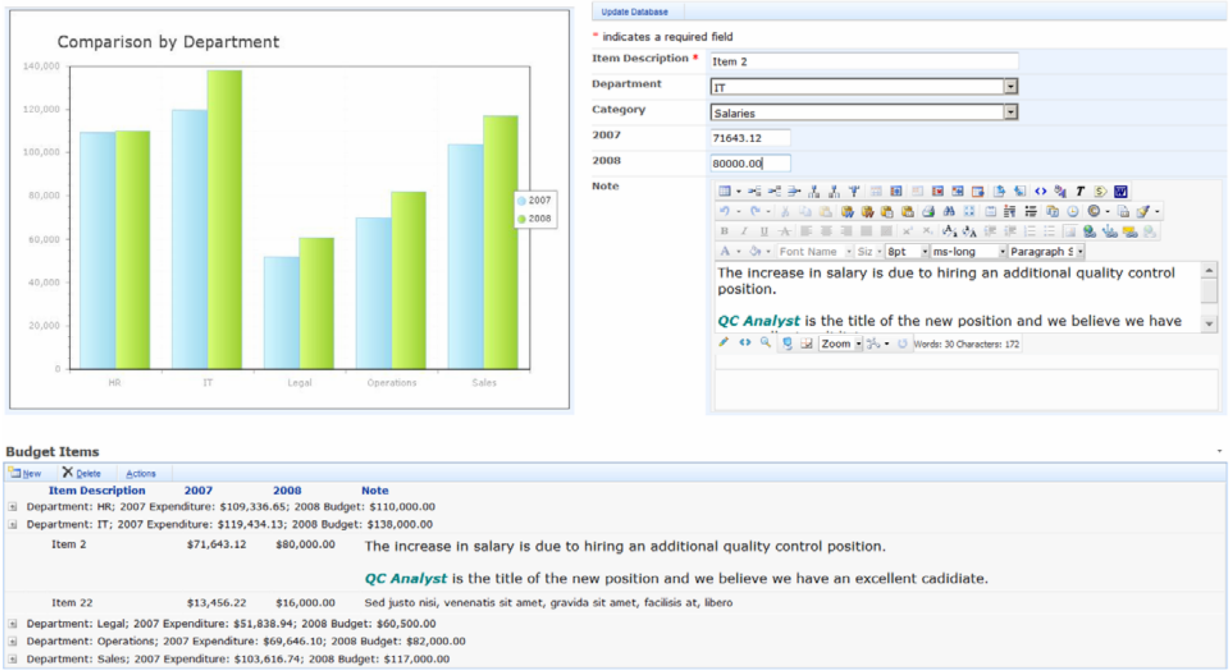
Mark investigates Development Studio and discovers he does not have to invest weeks to learn a new tool or any deep technical details. Mark is able to create a solution for his manager with only the following information:

- Location of the SQL databases
- Which fields he wants to display
- How he wants to filter the data

After a short conversation with the SQL administrator to configure a connection to the databases, Mark quickly configures a solution. He even has time to create charts to help visualize the budget.

The Details

The capability to filter, view, delete and edit external data is provided by Development Studio's four systems integration (SI) web parts: the SI List View, SI List Form, SI Chart View, and SI Selector. One of the pages Mark created is shown below:



* indicates a required field

Item Description *	Item 2
Department	IT
Category	Salaries
2007	71643.12
2008	80000.00
Note	The increase in salary is due to hiring an additional quality control position. ***QC Analyst*** is the title of the new position and we believe we have

Budget Items

Item Description	2007	2008	Note
Department: HR; 2007 Expenditure: $109,336.65; 2008 Budget: $110,000.00			
Department: IT; 2007 Expenditure: $119,434.13; 2008 Budget: $138,000.00			
Item 2	$71,643.12	$80,000.00	The increase in salary is due to hiring an additional quality control position. ***QC Analyst*** is the title of the new position and we believe we have an excellent cadidiate.
Item 22	$13,456.22	$16,000.00	Sed justo nisi, venenatis sit amet, gravida sit amet, facilisis at, libero
Department: Legal; 2007 Expenditure: $51,838.94; 2008 Budget: $60,500.00			
Department: Operations; 2007 Expenditure: $69,646.10; 2008 Budget: $82,000.00			
Department: Sales; 2007 Expenditure: $103,616.74; 2008 Budget: $117,000.00			

Figure 2: Mark's budget interface shows a live feed of data in a chart form in the upper lefthand corner, specific budget items in a list view along the bottom section, and a data edit form in the upper righthand corner.

Example 2: Aggregating Data from Multiple Sites

Robert is a project manager at a government agency. He manages all 12 of his projects in MOSS, but each project exists in a different site. Robert needs to create a dashboard to show management a single view of the progress on all of his projects.

Completing the Project with Native Tools

Microsoft web parts allow users to view information from only one list at a time, so Robert cannot use that to aggregate the information from his many projects in one site. He looks into re-architecting the projects so they all reside on one site, but that approach would create security problems. Native tools provide no viable solution.

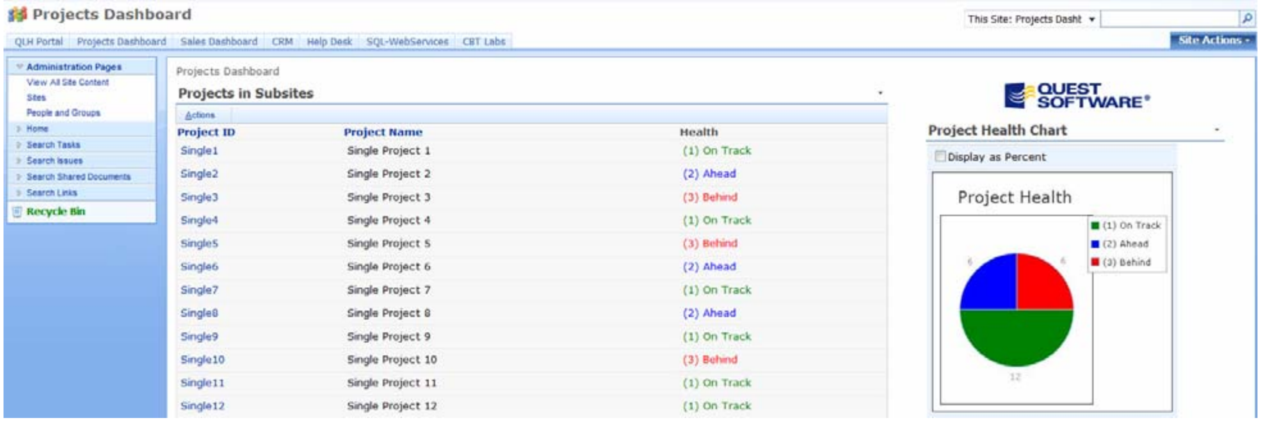
Completing the Project using Quest Development Studio

Robert investigates Development Studio and discovers he can display data from multiple lists residing in multiple sites, all in a single view. Despite the fact that Robert is not a developer, he is able to build his dashboard in a couple of hours by entering:

- The lists he wants in his view
- The fields he wants to display
- How he wants to filter the data
- How he wants to format the view

The Details

With Development Studio, data can be aggregated from multiple sites from anywhere within the bounds of SharePoint’s configuration database (that is, across site collections and applications). Robert created a dashboard site and used the Quest List View and Chart View web parts to aggregate the information from all 12 projects. His project status is now available at-a-glance, and he can search all tasks, issues, and documents across all projects, as shown in Figure 3.



Project ID	Project Name	Health
Single1	Single Project 1	(1) On Track
Single2	Single Project 2	(2) Ahead
Single3	Single Project 3	(3) Behind
Single4	Single Project 4	(1) On Track
Single5	Single Project 5	(3) Behind
Single6	Single Project 6	(2) Ahead
Single7	Single Project 7	(1) On Track
Single8	Single Project 8	(2) Ahead
Single9	Single Project 9	(1) On Track
Single10	Single Project 10	(3) Behind
Single11	Single Project 11	(1) On Track
Single12	Single Project 12	(1) On Track

Figure 3. Data from multiple lists can easily be aggregated into a single dashboard.

Example 3: Establishing a Parent-Child Relationship Between Lists

Leona is a business analyst at a mid-sized software company. The sales manager at Leona's company wants to be able to view and edit customer information and orders in one place. Leona's company implemented MOSS about six months ago, making it the logical solution for this project.

Completing the Project with Native Tools

Leona finds that Microsoft web parts don't support relationships between lists. In fact, there are no native options in MOSS to create a site that combines customer and order information.

Completing the Project using Quest Development Studio

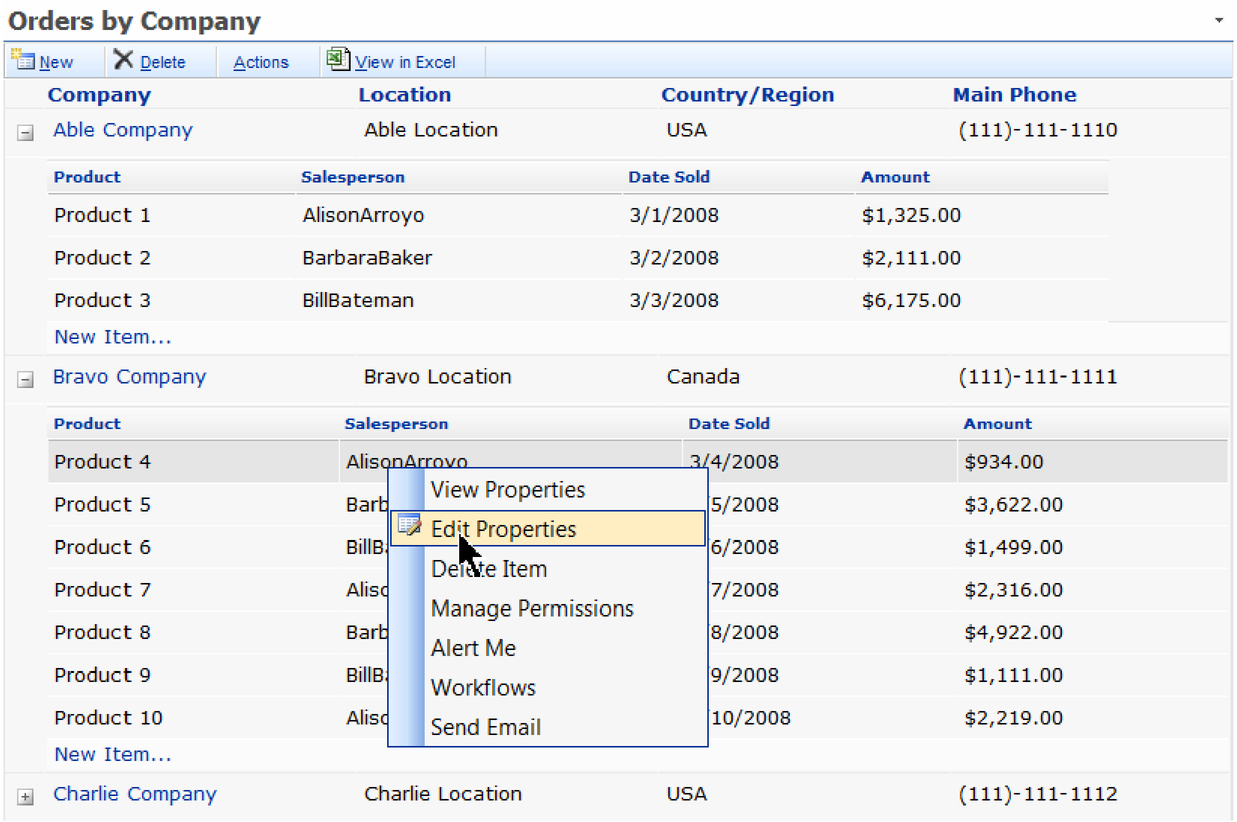
Leona investigates Development Studio and discovers she can create relationships between lists. To get started, she simply needs to know the following information:

- the lists she wants to relate
- the fields in the list she wants to relate
- how they relate
- how she wants the relationship displayed

Leona is able to build these relationships, organize the data, and run reports in a couple of hours.

The Details

Leona configured a relationship between the company list and the order list using the Quest List View web part to show in a hierarchical view, as shown in Figure 4.



Company	Location	Country/Region	Main Phone
Able Company	Able Location	USA	(111)-111-1110

Product	Salesperson	Date Sold	Amount
Product 1	AlisonArroyo	3/1/2008	$1,325.00
Product 2	BarbaraBaker	3/2/2008	$2,111.00
Product 3	BillBateman	3/3/2008	$6,175.00

New Item...

Company	Location	Country/Region	Main Phone
Bravo Company	Bravo Location	Canada	(111)-111-1111

Product	Salesperson	Date Sold	Amount
Product 4	AlisonArroyo	3/4/2008	$934.00
Product 5	Barb	5/2008	$3,622.00
Product 6	BillB	6/2008	$1,499.00
Product 7	Aliso	7/2008	$2,316.00
Product 8	Barb	8/2008	$4,922.00
Product 9	BillB	9/2008	$1,111.00
Product 10	Aliso	10/2008	$2,219.00

New Item...

Company	Location	Country/Region	Main Phone
Charlie Company	Charlie Location	USA	(111)-111-1112

Figure 4. Establishing a parent-child relationship between lists is easy.

Leona can also create the same kind of relationship using the Quest List Form web part. Employees can then automatically associate an order with a particular company when the order is created, which greatly simplifies the current manual process.

ARCHITECTURE

Overview

Development Studio contains two sets of web parts:

- SharePoint web parts that use data in SharePoint
- Systems Integration web parts that use data from external databases and applications

Both sets of web parts run only in the presentation layer of SharePoint; have a tabbed configuration interface that simplifies the process of customizing how the data is used in SharePoint; and can be enhanced by the developer using the custom actions capability.

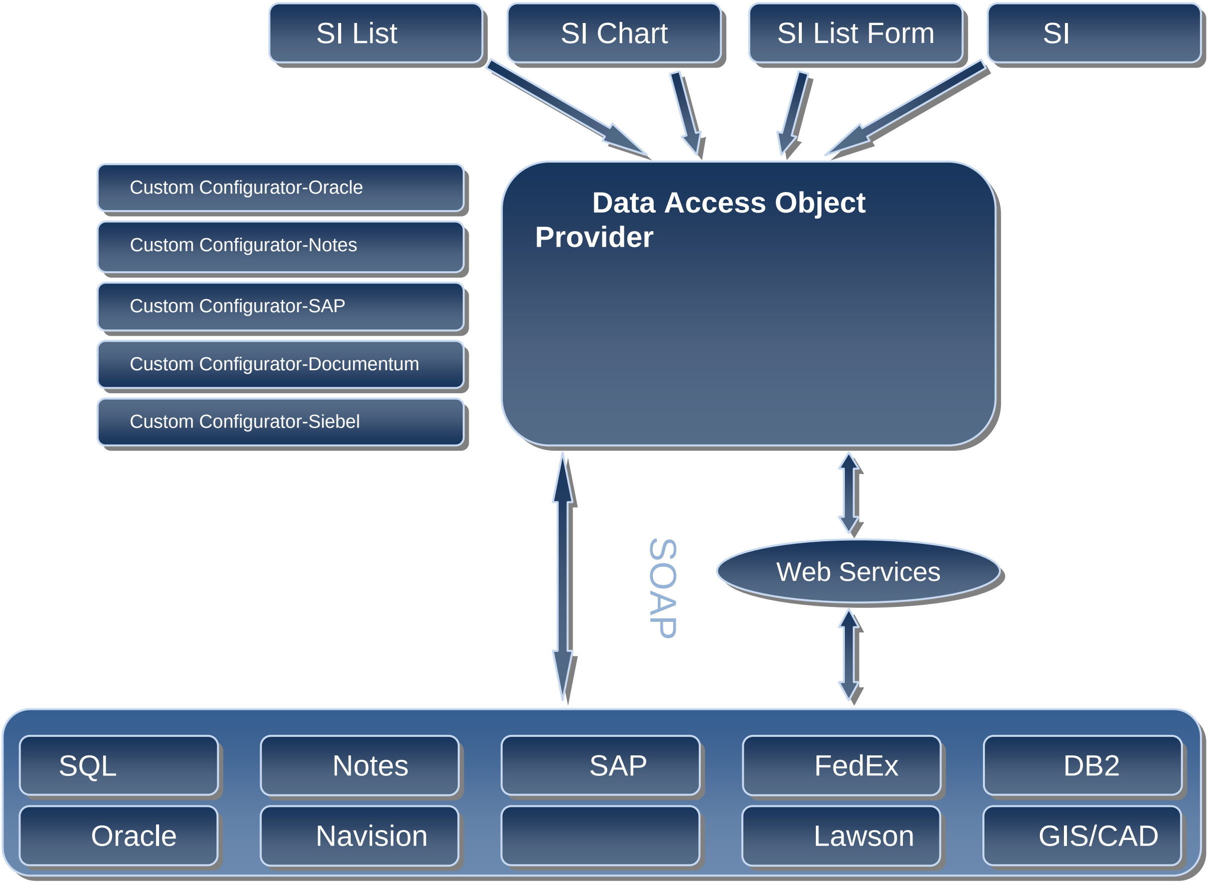
The configuration information is stored within each web part so alternate storage is not required.

SharePoint Web Parts

SharePoint web parts use the SharePoint object model, which is Microsoft's interface for interaction within SharePoint, and are compliant with the SharePoint security model. They do not run any additional services on the server and they do not alter SharePoint data. The SharePoint web parts support all list types except discussion board, survey, and picture library.

Systems Integration Web Parts

The architecture of the Systems Integration (SI) web parts can be diagrammed as follows:


SI List
SI Chart
SI List Form
SI
Custom Configurator-Oracle
Custom Configurator-Notes
Custom Configurator-SAP
Custom Configurator-Documentum
Custom Configurator-Siebel
Data Access Object Provider
SOAP
Web Services
SQL
Notes
SAP
FedEx
DB2
Oracle
Navision
Lawson
GIS/CAD

Web Parts and Their Capabilities

The following table lists the capabilities for each type of web part. Each capability is detailed in the section below.

Web Part	Capabilities
Systems Integration Web Parts** (The data's not in SharePoint? No problem.)	• Quest SI List View • Quest SI List Form • Quest SI Chart View • Quest SI Selector
Data Viewer Web Parts (View your data like you've always wanted to)	• Quest List View • Quest Calendar View* • Quest Chart View • Quest Excel Viewer
Data Entry Web Part (Sophisticated data entry–finally!)	• Quest List Form
Data Selector Web Parts (See only what you want to see)	• Quest Selector • Quest Multi-Selector • Quest Item Display
Navigation Web Parts (Get there faster)	• Quest Panel Menu • Quest Cascading Menu • Quest Bread Crumb* • Quest Page Redirector*
Display Web Parts (Create polished applications)	• Quest Caption Display • Quest Help Link • Quest User Display*

*Available for SharePoint Portal Server 2003 and Windows SharePoint Services 2.0 only

**Available for Microsoft Office SharePoint Server 2007 and Windows SharePoint Services 3.0 only

Descriptions of Capabilities

The following table describes the features of each web part capability:

List View	• Data can be aggregated from multiple lists across all sites defined in the configuration database. • Lists can be aggregated by one or more of the three different methods: select individual lists, select a list of sites and lists, or set recursive aggregation of all lists in a site structure. • Lists with parent-child relationships can be represented in a two-level hierarchical list view. • The data and actions available in the list view are constrained by the logged-in user’s security rights. • Users can perform a mass update of the values in a column. Items to be updated can be selected individually or through the use of a filter. • List views support multiple content types and display a **New** button for each content type so that the user can create items with different content types. • Users can copy or move documents (including their entire version history) between folders and between libraries. • Developers can configure *n*-level deep groupings. • Group headers can be configured to provide aggregate information in the group header on one or more fields using Sum, Min, Max, First, Last, and Count. • Users can dynamically create complex filters and sorts of the view. • List views can be implemented in "searcher mode" so that the view is empty until filter criteria have been selected to populate the view. • Users can create and manage folders with an interface similar to Windows Explorer. • Users can send e-mail from any list row or document that provides security aware links to the list, the list item, or the document. • The Custom Actions interface enables the developer to define additional buttons in the toolbar or context menu that run a program written in any .Net language.

	• Developers can show or hide any of the toolbar and context menu functions. • Developers can format the list toolbar, folders, and data presentation with styles, cascading style sheets, skins, HTML, and JavaScript. • Developers can implement logic-driven font formatting and graphics • Developers can create CAML using built-in CAML editors for both list items and folders. CAML editing supports all filter types, including date range filters. • List views can consume and provide connection filters. • List dependency is supported. That is, lists with child records can be set as dependent lists to prevent orphaned child records. • List views support and display recurrence and recurrence icons when used with event lists.
Excel Viewer	• The data in any list view can be exported to Excel. • Developers can determine which columns from a view are exported to Excel.
Chart View	• Supports 2D charting. • Supports the same data aggregation, data relationship, and filtering capabilities as List View with the exception of CAML filtering for folders. • Supports Sum, Avg, Count, Min, Max. StdDev, and VAR to the chart data. • Chart types are: Area, Bar, Bubble, Line, Pie, Point, and Spline. • Supports multiple data series of mixed chart types. • Supports formatting of chart size, background color, orientation, step values, legends, fonts, fills, etc. • Chart type and orientation can be changed by the user. • Chart views can be linked to list views—filters applied to the list view will be applied to the linked charts. • Charts can be saved or e-mailed as a graphic file in common graphics formats.

Calendar/ Timeline View (SP 2003 Only)	• Calendar views can be generated from any supported list types that contain a date field. • Timeline views can be generated from any supported list type that contains two date fields. • Supports the same data aggregation, data relationship, and filtering capabilities as List View with the exception of CAML filtering for folders. • Calendar View supports display in Year, Month, Week, Day and Outlook views. • Users can click an appointment to open the list form. • Supports background and foreground colors.
List Form	• Fields can be auto-filled from a parent list lookup or cross-site lookup field. • Supports master-detail relationships between fields in the form (for example, State filters County). Fields can be lookup or cross-site lookup columns. • Supports AutoPostBack, which causes the lookup field or cross-site lookup field to refresh the form when the user changes the selected item. • Supports list dependency: the list form can update entries in one or more dependent lists across sites. • Supports custom actions: developers can define additional buttons in the toolbar or context menu that runs a SharePoint function of the functions of programs written in any .Net language. • Supports implementation on any ASPX page and can connect to any list with appropriate security access. • Supports implementation of multiple forms on a page. • Supports all connected filter types. • Supports redirection when saving information: GoToNextPageUrl, GoToSource and TrySourceThenNextPageUrl. • Column titles can be customized and field descriptions can be added using plain text or HTML. • Column can be assigned a fixed value. • Columns can be hidden. • Columns can be set as read-only.

	• Multiple-choice columns can be displayed in *n* number of columns, as determined by the developer. • Text fields support data entry masks. • The List Form toolbar support formatting. • List Form e-mail provides links to the list, list item, and document.
Selector	• The selector displays data from a SharePoint list in a dropdown control. The selected item provides a filter that can be consumed across any scope, from a single list to multiple lists in multiple sites • The selector is a connection and session filter provider and consumer.
Multi-Selector	• A multi-selector contains one or more selectors. When multiple selectors are displayed, one selector will filter the next one.
Item Display	• Item Display shows information stored in a session variable and can pass it to a list form or list view, which is used for filtering long lists and supporting choices in complex data entry processes.
Cascading Menu	• Provides a list-driven hierarchical menu system that enables navigation to any URL. • Supports combinations of menu choices from one or more lists in any site and presents the combined information in a single menu, enabling both local and centralized menu management. • Constrains menu options displayed based on the security of the logged-in user. • Supports implementation in horizontal or vertical mode. • Provides extensive formatting options. • Supports icons. • Supports ALT-Key keyboard shortcuts.
Panel Menu	• Supports the same functions as Cascading Menu with the exception of horizontal mode. • Supports toggle open/closed through simple JavaScript on the page.

Caption Display	- Supports display of information from a central list of text or HTML, enabling the developer to implement text, graphics, and controls that can be implemented on multiple pages yet controlled from a central list.
Help Link	- Displays a help button or help text from a central list of URLs. The Help Link web part is invisible until the list it refers to is populated, which enables the developer to incrementally implement page-level user help.
Breadcrumb (2003 Only)	- Provides breadcrumb navigation between sites.
Page Redirector (2003 Only)	- Provides the capability to redirect a page to a specific URL for the logged-in user.

IMPLEMENTATION

Development Studio is implemented on front-end SharePoint Servers and must be implemented on all front-end servers in the farm. Once implemented, the web parts appear in the Server Gallery and are placed on the page in the same way as out-of-the-box Microsoft web parts.

Installation

Installation is via an executable installer that requires a license file. For each installation, the web.config file must be modified. A Configuration File Editor that evaluates the web.config file is provided to simplify installation. Farm installations support Web Garden and Sticky IP for routing URLs to the appropriate front-end server.

For installations supporting the Systems Integration capability, the Configuration File Editor supports the creation and graphical display of external content sources and parameters.

Detailed configuration information and labs are provided in the Development Studio Computer-Based Training. The administrator should review and understand the Installation and Systems Integration Configuration File modules before installing Development Studio.

Training and Help

Training is provided by the computer-based training application. This application does not require SharePoint; it can be installed on any Windows computer. It is freely distributable. The training application includes text, labs, and videos of the configuration labs.

The sites used for the labs are also bundled with the computer-based training. They can be found in the application installation folder along with installation instructions.

Help is provided through a .chm file that is accessible from the web part menu and from the All Programs Menu on the front end servers. The .chm file can be copied to and used on any Windows computer.

Application Templates

Three application templates are provided with Development Studio to enable developers to see practical examples of applications that can be used as starting points for application development. The templates are:

- Help Desk
- Project Management System
- Customer Resource Management

Evaluation Resources

Quest provides a number of evaluation resources:

- **Development Studio Evaluation Kit**

  This is a Microsoft 30-Day Test Drive Virtual Server 2005 R2 image that contains Development Studio, the computer-based training, the application template sites populated with data, and the labs for the computer-based training.

  This is the fastest and easiest way to set up an evaluation environment. The Evaluation Kit ships on two DVDs.

- **Development Studio Computer-Based Training**

  This training is described in the *Training and Help* section above.

- **Development Studio ROI Calculator**

  See how you could save time and cost on your SharePoint development project.

- **Development Studio Recorded Demonstration**

  This video shows the capabilities outlined in this document.

- **Development Studio Evaluation License**

  Your sales representative can provide you with a 30-day evaluation key that you can install in your SharePoint test and development environment.

All of these resources are available at http://www.quest.com/Development-Studio-for-SharePoint or from your Quest sales representative.

ABOUT THE AUTHOR

Dan Kruger is the Senior Product Manager for Development Studio for SharePoint. He is the former CEO of Workplace Architects, acquired by Quest in 2007.

Workplace Architects, Inc. and its predecessor company, Ability Engineering LLC, delivered strategy consulting, teamwork and communication training, systems integration, and custom application development services beginning in 1983.

ABOUT QUEST SOFTWARE, INC.

Quest Software, Inc. delivers innovative products that help organizations get more performance and productivity from their applications, databases and Windows infrastructure. Through a deep expertise in IT operations and a continued focus on what works best, Quest helps more than 50,000 customers worldwide meet higher expectations for enterprise IT. Quest Software can be found in offices around the globe and at www.quest.com.

Contacting Quest Software

Phone: 949.754.8000 (United States and Canada)
Email: info@quest.com
Mail: Quest Software, Inc.
World Headquarters
5 Polaris Way
Aliso Viejo, CA 92656
USA
Web site www.quest.com

Please refer to our Web site for regional and international office information.

Contacting Quest Support

Quest Support is available to customers who have a trial version of a Quest product or who have purchased a commercial version and have a valid maintenance contract. Quest Support provides around the clock coverage with SupportLink, our web self-service. Visit SupportLink at http://support.quest.com

From SupportLink, you can do the following:

Quickly find thousands of solutions (Knowledgebase articles/documents).

- Download patches and upgrades.
- Seek help from a Support engineer.
- Log and update your case, and check its status.

View the ***Global Support Guide*** for a detailed explanation of support programs, online services, contact information, and policy and procedures. The guide is available at: http://support.quest.com/pdfs/Global Support Guide.pdf