Business Insights Dashboard

Sage 500 ERP 2014

Contents

Business Insights Dashboard

1

The Sage 500 ERP offers Business Insights Dashboard allows executives to view company financial information from their Web browser.

In this chapter

Business Insights Dashboard Features

Overview

Introduction

Business Insights Dashboard includes a series of key business data components (also referred to as Web Parts) that generate summary information and charts from your financial information. These components appear in the user's Web browser and can be customized.

The Business Insights Dashboard contains three tabs: Dashboard, Explorer, and Links. The Dashboard tab contains a one-stop view of all your key business data. The Explorer tab allows you to view sales, purchases, inventory, project, and production information for the selected company. The Links tab allows you to view or add links to other Web sites.

Operation of Business Insights Dashboard

Overview

The Business Insights Dashboard is a Web site that provides executives with the ability to view and customize key business data for a selected company. You can also use the Business Insights Dashboard Web site to view Business Insights Explorer data, as well as set links to commonly used sites. Business Insights Dashboard uses ASP.NET technology to deliver consolidated data from the Sage 500 ERP database to the client Web browser. ASP.NET contains the Web Part Manager which is a server control that manages all functionality, events, and customization of Web Parts within the Dashboard Web pages.

To learn more about using the Business Insights Dashboard, access the *Business Insights Dashboard* tutorial by clicking Tutorials on the Sage 500 ERP Desktop.

Flow chart   The following figure shows how the Business Insights
                   Dashboard works:
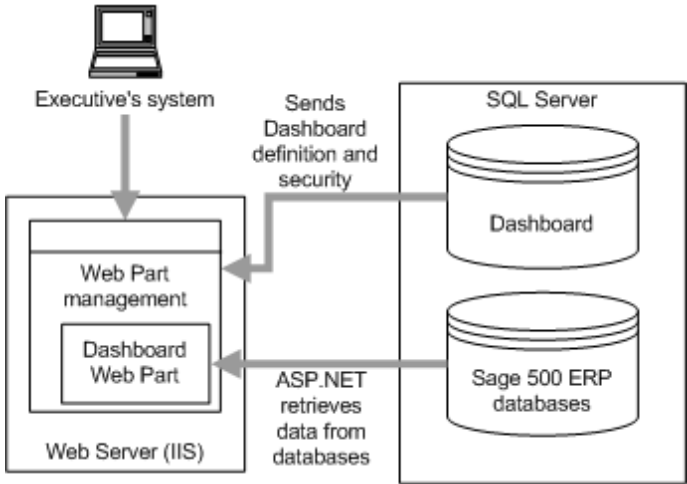
Updates   The data that appears in the Business Insights Dashboard
                   Web site is updated according to the schedule you define in
                   Set Up CI Options. You can also generate manual updates.

Installation of Business Insights Dashboard

Where Business   Business Insights Dashboard is installed as an IIS application
Insights   on the Web server.
Dashboard is
installed

Requirements   Business Insights Dashboard requires ASP.NET and AJAX
                   extensions, both of which are installed automatically if your
                   system does not have these installed.

Security   Business Insights Dashboard requires Windows
                   authentication. It cannot be configured for anonymous
                   access. For more information, see *Security* on page 7.

Site Hosting and Configuration

2

This chapter describes hosting and configuration issues to consider when setting up your Web sites. You need to configure the sites for the appropriate level of security for Business Insights Dashboard. The server where the dashboard Web site is hosted needs access to the Sage 500 ERP server, so you need to configure the necessary components to enable the Web site to communicate with the Sage 500 ERP databases.

In this chapter

Information Processing

Overview

Introduction   This section provides examples of how information is processed by the Business Insights Dashboard and Microsoft Web components. An understanding of this information will help you to plan your Web sites.

Business Insights Dashboard Processing

Introduction   When customers connect to the Business Insights Dashboard web sites, they interface with IIS applications.

Business Insights Dashboard   Business Insights Dashboard uses scheduled SQL Server tasks to update data. For more information, see *Setting SQL Server Schedules* on page 41.

Site Hosting Options

Overview

Introduction   This section describes options for hosting and security.

You can place the Web site on an intranet, an internal Web server, or a separate Internet service provider (ISP). This section explains how to configure Business Insights Dashboard for different types of hosting.

Internal Hosting

Introduction   If your company does its own Web hosting (either on-site or in a co-location), install BID on your Web server. Then surround the Web server with firewalls (one to protect it from the outside and the other to limit access by internal users).

NetBIOS connection requirement   To install Business Insights Dashboard on the Web server, you need to have local access on your system to the system's local drives through NetBIOS. You also need NetBIOS access to the shared folder for inventory images.

Hosting Through an ISP

Introduction   Business Insights Dashboard can also be hosted by an external ISP.

NetBIOS connection requirement   The ISP can grant you access through a virtual private network (VPN) to communicate to the Web server. If the ISP does not have a VPN available, you will need to either go to the physical server and install it directly or send the Business Insights Dashboard installation DVD-ROM to the ISP and have them install it for you.

Security

Overview

Introduction
To protect your data and control access, Business Insights Dashboard works with security provided with Sage 500 ERP, and the IIS applications.

This section describes features and issues that you should consider in setting up security for your site. For more information, see the Microsoft IIS documentation or the Microsoft Web site at www.microsoft.com. If you have concerns about setting up security yourself, work with your ISP or a consultant to set up the best security for your site.

General Security Considerations

Public and private site considerations
The amount of security you set for a Web site depends on whether the site should be public or private. For private sites, configure them for high security. For public sites, you must balance the need for security with the need to make your Web site accessible. Some forms of security can make it difficult for legitimate users to enter and discourage them from visiting your site.

Web Site	Recommended type of site	Security issues
Business Insights Dashboard	Private	Windows authentication required
		Cannot use anonymous access

Controlling access to the Web site and data
To gain access to portions of the Web site and data, users need to be authenticated either by having them enter a logon name and password or by using logons set up in the system.

This section explains how security and logons work. For more information, see *Configuring the Web Site* on page 91.

Firewalls

Surround the Web server with two firewalls: one protecting the links from Internet and the other protecting links from internal users. The firewall on the internal link also protects your internal systems if someone manages to breach the external firewall. Configure the firewalls to allow access to port 80.

Firewalls are your first line of defense, but they should not be the only one. You should configure the Web server to provide additional protection.

Access to the Web site

Business Insights Dashboard Web site use IIS to grant access. The settings for each site are different. The Business Insights Dashboard Web site uses Windows authentication. You need to configure the Web server to grant access to the Web site only to the users who need it.

You can provide additional security through IIS by granting authenticated access to folders, the system, or the domain. If you use this feature, you also need to create a local account on the Web server with a user name and password for the system or domain; this account does not create the logon for the Web site in Sage 500 ERP. The authentication of folders in IIS requires that the Web site be on an NTFS volume.

You can also use the options provided with NTFS to provide additional access to your Web site. For example, you can set some folders as read-only to prevent them from being modified by unauthorized users. You can also limit access to some folders entirely to all but specific users.

Secure Socket Layers (SSL)

Introduction   You can use SSL with any of the security methods described in this section to add greater security to the Web site. Your customers will appreciate the protection SSL gives them as they send their sensitive information through your site.

How SSL works   SSL provides a secured communications channel. Data is encrypted while it is in transit. The data is decrypted when it reaches its destination, but it is also decrypted at routers and bridges that it stops at on the way.

Web sites with SSL are connected with HTTPS instead of HTTP. A padlock icon appears on the status bar of the browser window when a user visits a site with SSL.

SSL and certificates   To use SSL on a Web site, you need to set up a certificate for the Web server. Certificates grant further security by using keys to authenticate the transaction: your public key and the customer's private key.

Certificates can be generated in the following ways:

- You can obtain certificates from a public certificate authority. These certificates are normally accepted automatically by external users.

- You can generate your own certificate using Windows certificate services. By default, these certificates are not trusted by external users. Users are prompted to accept the certificate each time they log on to your site, but they have the option of installing the certificate so that they accept the certificate automatically.

See the IIS documentation for instructions on obtaining and installing certificates.

SSL options   After you have set up a certificate for the server, you can select SSL options for the Web site.

One of the options you can use is 128-bit encryption. This offers a high level of security for data transmissions. Keep in the mind that if you use 128-bit encryption, the user of your site must have a 128-bit version of Internet Explorer, which is not available outside the United States. If you plan to support users from different countries, avoid using 128-bit encryption.

Effects of using graphics

If you use SSL with your Web site, you should put all images of items on your Web server or on another server with SSL. If you have images on a server without SSL, the following message appears each time the user connects to your site:
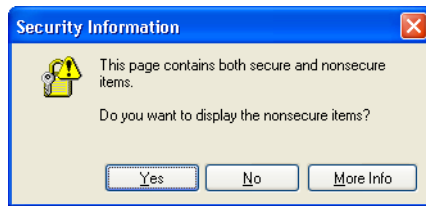
The message indicates that there are elements on the page that are not secure. This does not affect the secure communications between the client system and your Web site. To avoid having the message appear whenever customers enter the site, put all images on your server.

Addresses of the Web Sites

Introduction   An address of a Web site is based on the IIS application name of the Business Insights Dashboard Web site. The Web sites are typically installed in subfolders of \Inetpub\wwwroot. When you access the pages from a browser, replace \Inetpub\wwwroot with the domain name of the Web server.

This section describes how the addresses are determined, so that you can determine what URL to use for the Web sites.

IIS Folder Names

Default IIS application names   The default IIS folder names for the Business Insights Dashboard are as follows:

Application	Default Folder
Business Insights Dashboard	BusinessInsightsDashboard

Location of files   By default, files for Business Insights Dashboard are installed into a folder based on the IIS application name. You can have the files installed into a different folder by using Custom installation. For more information, see *Installing the Software on page 29*.

Changes on upgrades   If you are upgrading from a previous version of the Business Insights Dashboard that used the default folder names, the old folders and IIS application names are deleted and new ones are created with the new default names. Be sure to inform your users of the changes and have them update their bookmarks. The Business Insights Dashboard Chart Images folder may remain and requires manual removal.

Determining the URL

Domain and folder names

The URL for an Business Insights Dashboard is the following format:

> http://WebServer/ApplicationName

Where *WebServer* is the Web server domain name and *ApplicationName* is the name of the IIS application name.

For example, if the domain name of the Web server is www.mycompany.com, and you use the default IIS application name, Business Insights Dashboard, the URL is:

> http://www.mycompany.com/BusinessInsightsDashboard

Typing this URL opens the default page of the Web site after you finish configuring the Web site.

Using different folder names

During installation, you can select a different IIS application name. If you are running a publicly accessible site, you might choose a more descriptive name than the default.

Suppose you want to use the following URL:

> http://www.mycompany.com/bidashboard

In the Installation wizard, enter bidashboard as the IIS application name for the Web site.

Changing the computer name

Use caution when changing the name of the Web server. After renaming the Web server, edit the anonymous logon account for the Internet Information Services (IIS). Then change the logon to a valid domain and user account.

Preparing for Installation

3

This section covers prerequisite steps and information that you need before you start installing the Business Insights Dashboard. This information affects how the Web site operates, so read this information carefully before installing the Web components.

In this chapter

System Requirements

Overview

Introduction   To ensure a smooth installation, make sure that the servers and workstations you plan to use with the Business Insights Dashboard meet the system requirements shown in this section.

Overall System Requirements

Compatibility and Resource Guide   For more information about system requirements, supported platforms, and recommended configurations, refer to the *Sage 500 ERP Compatibility and Resource Guide* located in the Support area of the Sage Customer Portal.

Business Insights Dashboard Requirements

Required software   Business Insights Dashboard requires ASP.NET and AJAX extensions, both of which are installed automatically if your system does not have these installed.

Case Sensitivity

Description   This version of Sage 500 ERP supports case-insensitive SQL Servers only. To upgrade from a case-sensitive server, see your authorized reseller for assistance.

Configure Windows 2003 Server

The following information regarding enabling ASP.NET, Active Server Pages, and WebDAV applies to the Standard and Enterprise editions of Windows 2003 Server.

Enable ASP.NET

Use the following steps to enable the ASP.NET component in the Windows Component Wizard.

1 In the Control Panel window, open Add/Remove Programs.

2 In the Add/Remove Programs window, click Add/ Remove Windows Components.

3 In the Windows Components Wizard page, select the Application Server check box, and then click Details.

4 In the Application Server window, select the ASP.NET check box, and then click OK.

Enable the Active Server Pages and WebDAV extensions

Use the following steps to enable the Active Server Pages and WebDAV extensions in the Internet Information Services (IIS) Manager window.

1 In the Control Panel window, open Administrative Tools.

2 In the Administrative Tools window, open Internet Information Services (IIS) Manager.

3 In the Internet Information Services (IIS) Manager window, expand the Web Service Extension folder.

4 Highlight Active Server Pages, and then click Allow.

5 Highlight WebDAV, and then click Allow.

6 Close the Internet Information Services (IIS) Manager window.

Configure Windows 2008 Server

Enable ASP.NET   Use the following steps to enable the ASP.NET component in the Windows Component Wizard.

1   In the Administrative Tools window, open Server Manager.

2   Select the Roles folder, and click Add Role Services.

3   In the Select Server Roles window, select the Web Server (IIS) check box, and click Next.
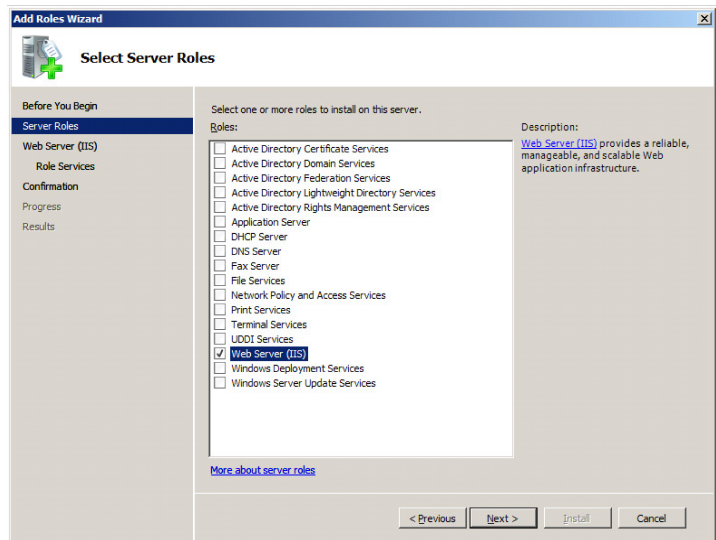
4   In the Web Server (IIS) window, click Next.

5 In the Select Roles Services window, select the ASP.NET and ASP check boxes, and then click Next.
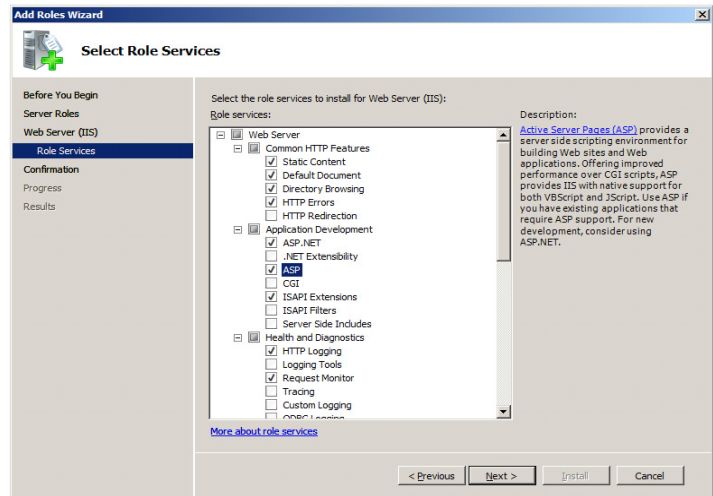
6 In the Confirm Installation Selections window, click Install.

Configure Windows 2012 Server

Enable ASP.NET   Use the following steps to enable the ASP.NET component in the Windows Component Wizard.

1   From Start, open Server Manager.

2   Select Add roles and Features.
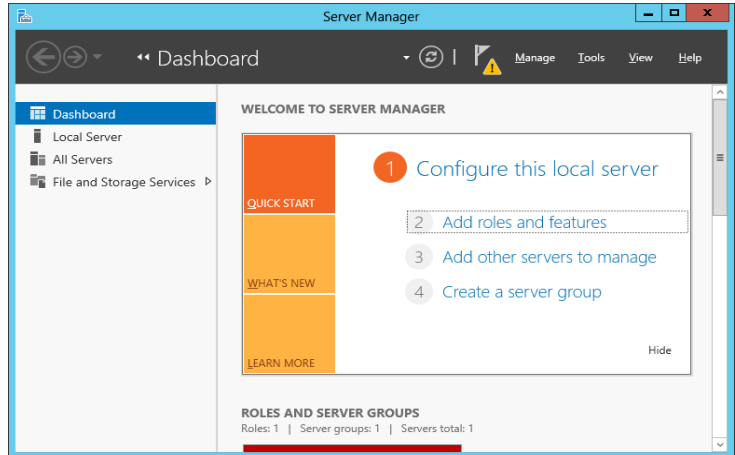
3   In the Select Server Roles window, Select the Web Server (IIS).
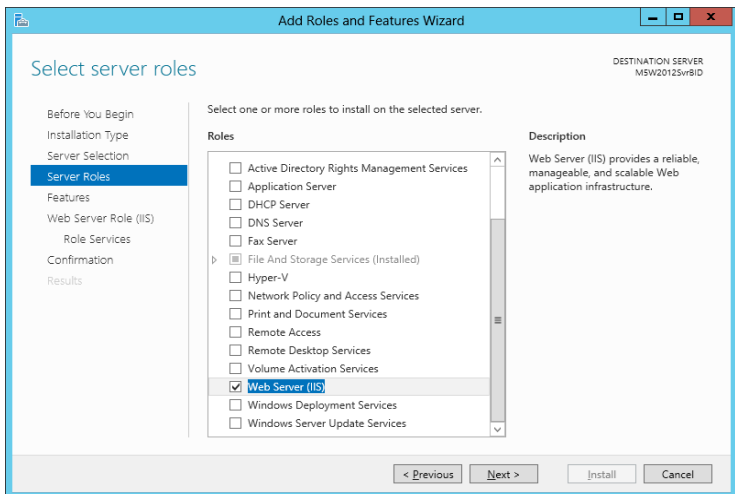
4 Add Roles and Features Wizard click on Add Features.
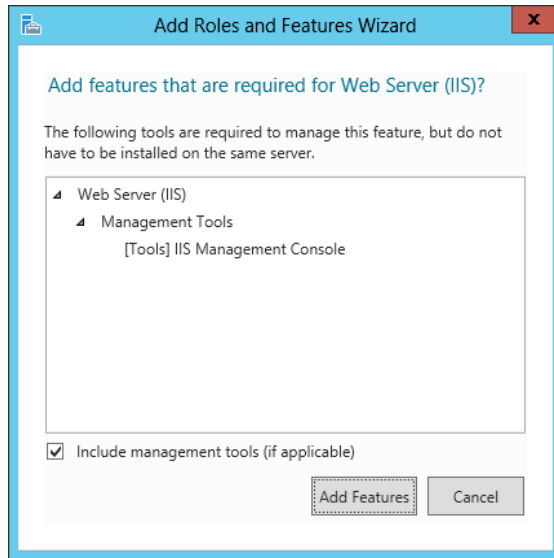
5 In the Web Server (IIS) windows, click Next.

6 In the Select Server Roles Window Enable .NET Framework4.5 features , select .NET Framework 4.5 and ASP.NET 4.5.

7 Enable the .NET Framework 3.5 Features (this includes .NET 2.0 and 3.0).
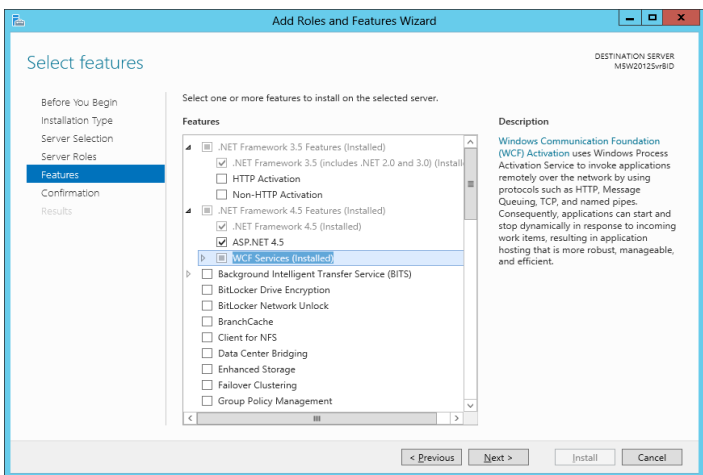
8   In the Select Role Services, select WebDAV publishing.
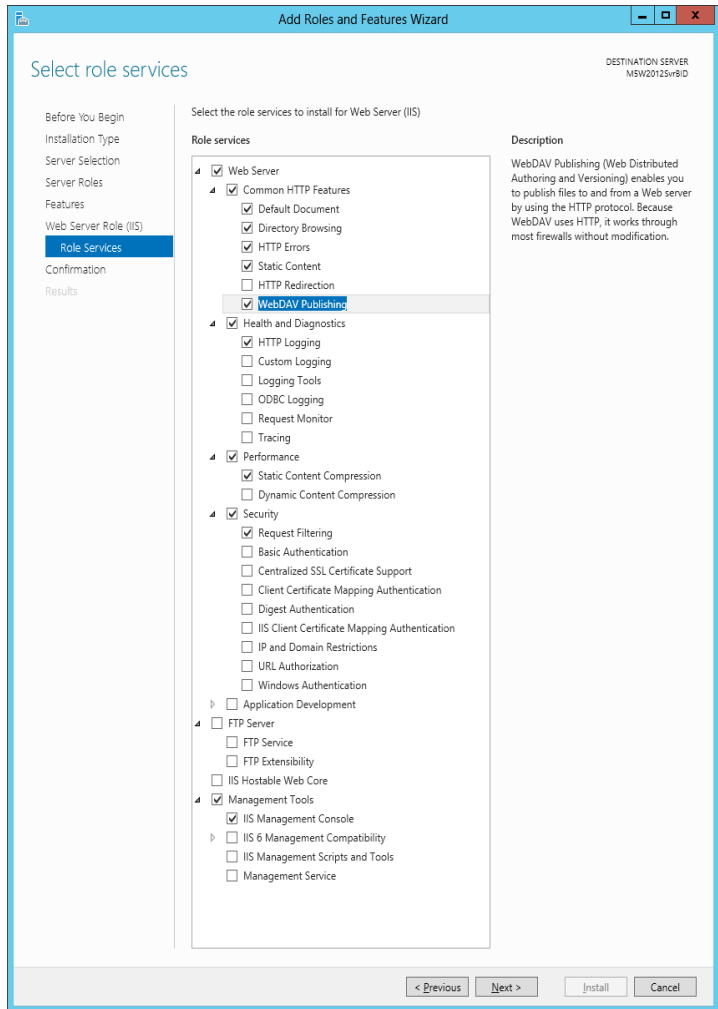
9 In the Select Server Roles, expand Application Development and check ASP, ASP.NET3.5 and ASP.NET 4.5 checkboxes.
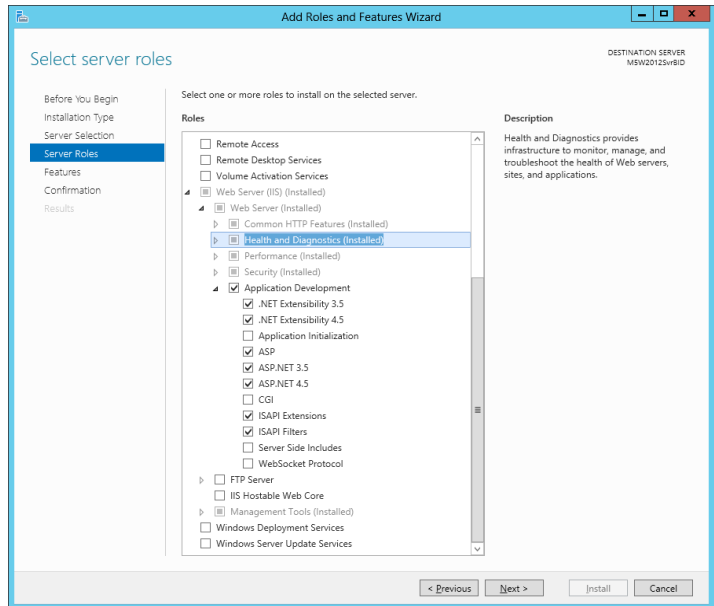
10 Add Roles and Features Wizard click on Add Features.

11 In the confirmation Installation, click Install.

Setting Website Security

To set security options, do the following.

1   Select Windows Start, Click on Internet Information Services.
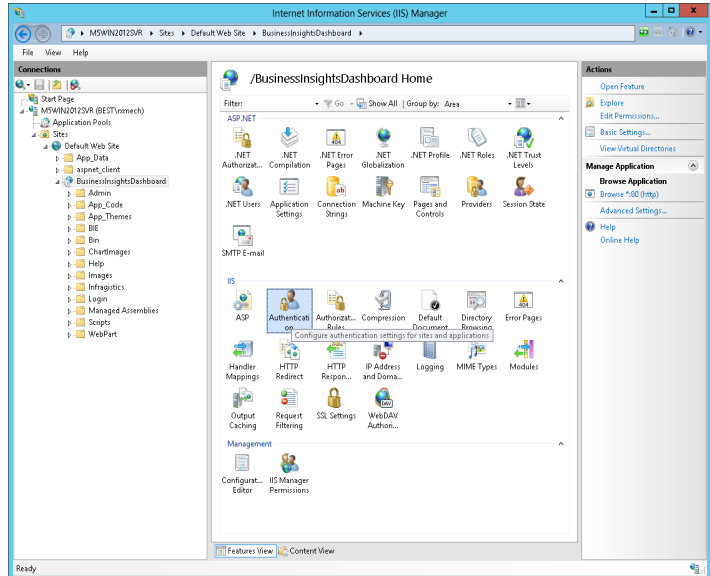
2   Select the application to configure, and double click on authentication.

3   Enable the appropriate security model.

Installing Required Software

Overview

Introduction

Before installing the Business Insights Dashboard, you need to install the software as shown in following sections:

This section covers issues you need to know when installing this software for use with the Business Insights Dashboard.

Installing and Upgrading Sage 500 ERP

General installation steps

Installing Sage 500 ERP and the Business Insights Dashboard consists of installing or upgrading Sage 500 ERP, and installing and configuring the Business Insights Dashboard on the Web server. For complete installation instructions for Sage 500 ERP, refer to the *Installation and System Configuration* guide.

Required tasks

Before installing the Business Insights Dashboard, install or upgrade Sage 500 ERP. Follow the instructions in the *Installation and System Configuration* guide and be sure to perform the following:

- For a new installation, install new system, application, and Business Insights Dashboard databases. For an existing installation, upgrade the current system and application databases to the new version.Install the Sage 500 ERP client software on at least one system.

- Register and activate the applicable modules for Business Insights Dashboard:

  - Accounts Receivable
  - Cash Management
  - Business Insights Dashboard *****
  - General Ledger
  - Inventory Management
  - Sales Order

  Note: Only Business Insights Dashboard is required to be registered and activated. The other modules provide data for Web Parts. Activate only the modules you purchased. Business Insights Dashboard Web Parts show data only for modules you registered and activated.

Security rights required   Users who administer and configure the Business Insights Dashboard needs Normal access rights to CI - Options Setup.

Installing Business Insights Dashboard

4

This chapter shows you how to install Business Insights Dashboard.

In this chapter

Web Component Installation

Overview

Introduction   After the Web server is ready, run the installation wizard to install the Web components with the parameters you specify.

Method of Installation

Description   You can install Business Insights Dashboard using the Business Insights Dashboard Installation wizard, which allows you to install the Business Insights Dashboard alone. Use this wizard for installing a production system.

For information about this installation method, run the Installation Advisor from the Installation screen or refer to the *Installation and System Configuration* guide.

Pre-Installation Checklist

Introduction   Before you begin installing the Business Insights Dashboard, make sure you have the following information. See the indicated pages of the Business Insights Dashboard guide for details.

Checklist   Complete the following checklist:

Information		See
Applications to install	Business Insights Dashboard alone or with another application (Use Custom)	

Information		See
IIS application names	BusinessInsightsDashboard	
Sage 500 ERP databases	_app _____ User name for databases _____ Password for user _____	

Starting the Business Insights Dashboard Installation Wizard

Introduction

This section describes how to install the Business Insights Dashboard by using the Installation wizard.

Ways of starting the Installation wizard

To start the Business Insights Dashboard Installation wizard, perform one of the following:

- Insert the Sage 500 ERP DVD-ROM into the DVD-ROM drive. From the installation window that appears, click Business Insights Dashboard.

- Use Windows Explorer to navigate to the InternetApps folder on the DVD-ROM. Run Setup.exe.

- If you send the installation files to an ISP by FTP, have them copy all the files to the same folder. Then, have them run Setup.exe.

Use of Windows Installer

All Sage 500 ERP installation programs use the Microsoft Windows Installer engine. The installation program uses the latest version of the Windows Installer engine if it is already on the system; however, if it is not currently on the system, a message appears indicating that the Windows Installer engine is installing. It may take a moment to install the software, and then the installation starts.

The installation program also installs Microsoft .NET Framework if it is not already installed on the system.

Uninstallation of Prior Versions

What gets removed

If you have an earlier version of the Web components on your system, the installation wizard detects it and prompts you to remove it before you install the new software. Removing an earlier version does the following:

Removed	Not Removed
All files that were installed. including HTML pages, graphics, scripts	Pages added to the site
	Customized pages
Customized pages that have the same names as the original	Queues
	Business Insights Dashboard ChartImages folder
Service packs and monthly updates added to the installation	Digital Dashboard

Click Remove to uninstall the old software. The wizard then starts the installation process.

> **Important:** The old folders and IIS application names are removed and new ones are created using either the new default names or the names you specified. For more information, see *IIS Folder Names* on page 11.

Installing the Software

Procedure   Use the following procedure to install the Business Insights Dashboard.

1   In the Welcome screen, click Next.

2   Review the license agreement that appears on the Customer Information page. Then, select "I accept the terms in the license agreement" and click Next to continue.

3   You are prompted to enter your information. The fields display the name and organization as recorded in the Windows Registry. You can change this information as needed and select whether the software is available to all users or just yourself. Click Next to continue.

4   The following setup options are available:

▶   Complete: Installs the component into the default folders. Use this option if you want to be sure you are installing all files and have sufficient space on drive C. For more information, see *IIS Folder Names on page 11*.

▶   Custom: Allows you to select the drive and folder where files will be installed. Use this option if you are limited on disk space or want more control over the installation. Business Insights Dashboard is installed in the Business Insights Dashboard folder under the folder you selected.

Important: Use the Custom option if the default installation is not an NTFS volume. Installing the Business Insights Dashboard on an NTFS volume is

not required, but it enables you to take full advantage of security features.
Select the option and click Next.

If you selected Custom, go to the next step. If you selected Complete, go to step 6.

Complete the page as follows:

-To change the drive and path where Business Insights Dashboard will be installed, click Change. Click Space to see how much space is available on your drives and how much space will be used by the database installation software. Note that some files will be installed on the local hard drive.

Click Next to continue.

5   When you are installing Business Insights Dashboard, the Installation Requirements page appears.

Enter the following for Business Insights Dashboard:

▶   The IIS application name for Business Insights Dashboard. This is used as the name of the folder where the Web components are installed and to identify the Web site in IIS, queues, and other places in the system.

▶   The name of the SQL server containing the Sage 500 ERP database you are using. If you have already selected a server for other Business Insights Dashboard, it will appear in this field and must be used. For more information on securing your connection, see Chapter 5, *Configuring Business Insights Dashboard*.

▶   The name of the application (_app) Sage 500 ERP database you are using. The database must be on the indicated or selected SQL Server.

> ► A user name and password for the application database.

Click Next to continue.

6 In the Ready to Install the Program page, click Install to install the software.

7 In the final page that appears, click Finish.

Changing the Installation

Introduction You can remove Business Insights Dashboard, change the configuration, or reinstall by running the Installation wizard. When you run the wizard on a system that already has Business Insights Dashboard installed, the wizard presents an appropriate set of options.

Procedure Follow these steps to run the client installation to make changes:

1 Start the Installation wizard as shown in *Starting the Business Insights Dashboard Installation Wizard* on page 27. When the Installation wizard starts, click Next to continue.

2 Select the desired option:

> ► To change the installation, select **Modify.**

> ► If you experienced problems with the software or need to repair a corrupted installation, select **Repair.** This reinstalls any missing or damaged files.

> ► To uninstall the software from the system, select **Remove**.

> **Note:** Selecting Remove from this screen does not uninstall service packs, monthly updates, or hot fixes you added to the current version. If you are running the installation wizard to update an older version, clicking the Remove button does remove those items.

3 A screen appears where you can confirm your action. Click Install (for Modify or Repair) or Remove to proceed.

Configuring Business Insights Dashboard

Introduction   You need to configure the Web component to connect to the Sage 500 ERP databases, set up the security, open the Web site, and set general options in Sage 500 ERP.

Follow all of the procedures in this section in the order listed in the following sections:

Setting Web Site Security

Introduction Run the Internet Information Services Manager to check security and other Web site settings. You can use the standard security or add other levels of security. For more information, see *Security* on page 7.

> **Note:** If you are setting up the Web site security for a Windows 2008 Server, refer to the *Setting Web Site Security on Windows 2008 Server* on page 35.

Procedure To set security options, perform the following:

1 Select Windows Start menu > All Programs > Administrative Tools > Internet Information Services.

2 Look for the name of the Web site and right-click it. From the menu that appears, select Properties. From this window, set the options you need for the site as shown in the following sections.

Setting up standard security To use standard security, follow these steps in the Web site Properties window.

1 Click the Directory Security tab. Under "Anonymous access and authentication control," click Edit. The following window appears:
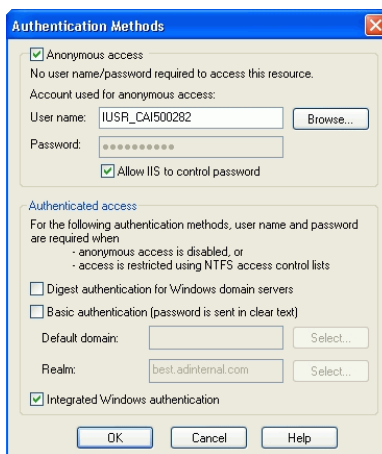
2 Depending on the Web site you are configuring, set the options as follows:

► Business Insights Dashboard: Clear the Anonymous access check box. Leave the Integrated Windows authentication check box selected.

3 Click OK to accept the changes.

Setting up authenticated access

For additional security for the Web server, you can set up IIS to authenticate people using the site. Users first need to log on to the Web server and then log on to the Web site. The logons are separate and unlinked. They can be different for the same customer. See the IIS documentation for instructions.

Setting Web Site Security on Windows 2008 Server

Introduction

Run the Internet Information Services Manager to check security and other Web site settings. You can use the standard security or add other levels of security. For more information, see *Security* on page 7.

Procedure

To set security options, perform the following:

1 Select Windows Start menu > All Programs > Administrative Tools > Server Manager.

2 Select Roles folder > Web Servers (IIS) > Internet Information Services (IIS) Manager.

3 Select the application to configure, and double-click Authentification.
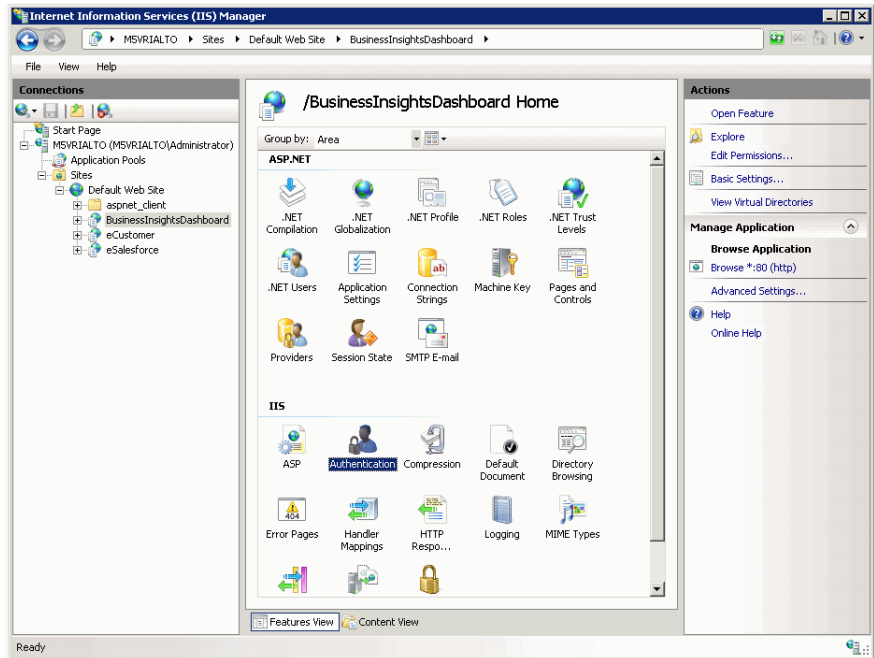
4 Enable the appropriate security model for your application, and close the window.

Securing Specific Pages and Folders

Introduction

To prevent pages and folders from being used by unauthorized people, apply security to them through IIS. If you installed Business Insights Dashboard on an NTFS volume, you can limit access to specific users or groups using NTFS security permissions.

Setting authenticated access in IIS

The first step in securing pages is to require authenticated access. An external user must log on with a valid user name for the Web server or its domain to gain access to the files.

To run Internet Information Services (IIS) Manager, follow the procedures in *Setting Web Site Security* on page 34. Then, perform the following:

1   In the Tree pane, expand the listing for the Web site. One of the folders in the listing is Administration. You can set security for an entire folder or individual files.

2   Right-click the folder or the file, and from the menu that appears, select Properties.

3   In the Properties window that appears. Click the Directory Security tab. Under "Anonymous access and authentication control," click Edit.

4   In the Authentication Methods window that appears, remove checks from all boxes, except Integrated Windows authentication, as shown in the following figure:
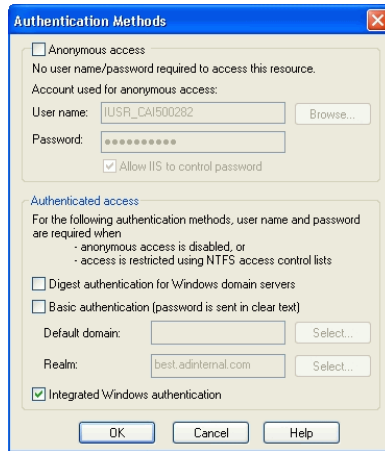
Note: If you are not using SSL, you will receive a warning that your password might be examined during authentication. Click Yes to continue.

5   By default, the system uses the local Windows domain where there Web server is located to authenticate. To authenticate using another domain, click Edit. In the dialog box that appears, select the domain to use and click OK.

6 Click OK to close the Authentication Methods dialog box. Then close Internet Information Services.

Limiting access to the file or folder

IIS only enables you to limit access to a folder or file to authenticated users of the Web server or domain. To restrict access to certain user groups, you need to set file sharing options in Windows Explorer. These options are available if you installed the Web components on an NTFS volume.

Follow these steps:

1 In Windows Explorer, navigate to the folder or files for which you want to control access.

2 Right-click the folder or file. From the menu that appears, select Properties.

3 In the Properties window that appears, click the Sharing tab.

4 Select "Share this folder." You can then set options for controlling access.
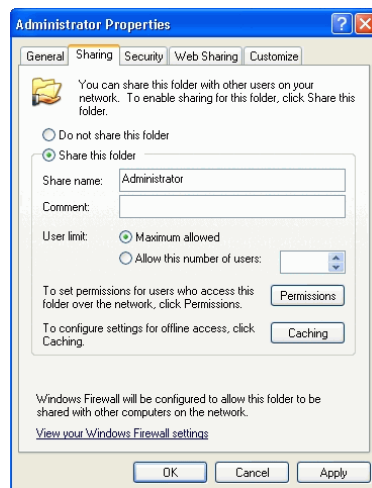
5 Click Permissions. The following window appears:
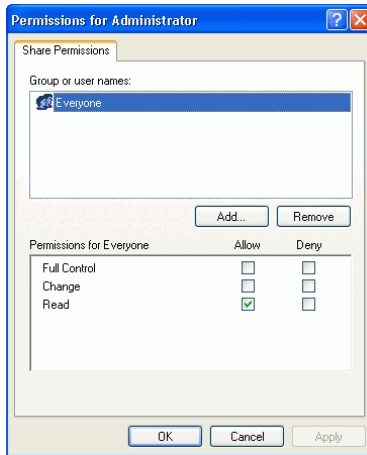
Select Everyone. In the Permissions box, select the Deny check boxes for all types of permissions.

6 Click Add. The following window appears:
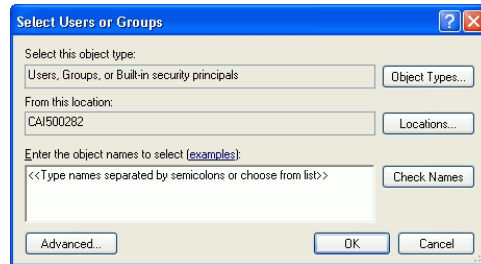
Select the computer or domain at the "Look in" field, and then select the users or user groups to which you want to grant access and click Add. Click OK when you are finished.

7 In the Permissions window, select the users or groups to which you want to grant access. Select the Allow check boxes for those users and groups. Click OK when you are finished.

Setting Site Options in Sage 500 ERP

Procedure   Use the Common Information (CI) module for general settings for all Business Insights Dashboard. Follow these steps in Sage 500 ERP:

1 Select Common Information Maintenance menu > CI Setup > Set Up CI Options. The following window appears:

Finish and Exit

2   Click the Other tab. The following fields appear:
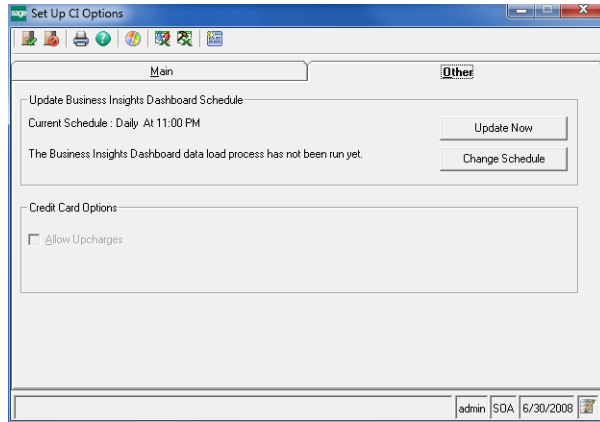
Complete the tab with the information for your configuration. For more information, refer to the Help system.

3   If you are using Business Insights Dashboard, set the update schedule and make sure that the SQL Server Agent is running. For more information, see *Setting SQL Server Schedules* on page 41.

To update Business Insights Dashboard data without scheduling, click Update Now.

4   When you are finished, click the Finish and Exit button.

Setting SQL Server Schedules
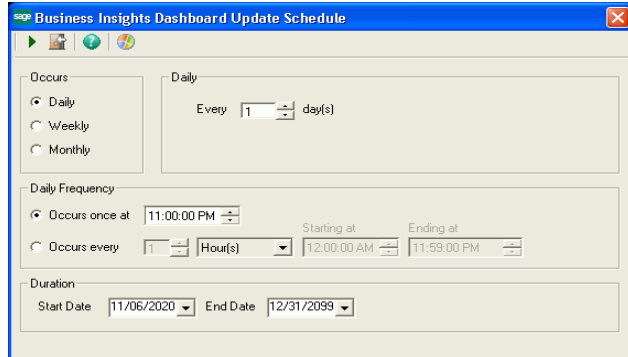
Introduction   For tasks with data from SQL Server databases, such as data for Business Insights Dashboard Web Parts, you can set an update schedule so that the data is updated automatically at the intervals you set. For more information about SQL Server scheduling, see *SQL Server Books Online*.

Procedure   Follow these steps to set the schedule:

1   On the Business Insights Dashboard tab in Set Up CI Options, click Change Schedule. The following window appears:

2   In the Occurs section, choose one of the following options. The section to the right of Occurs changes according to the option you select.

▶   Select Daily to pick a frequency of days, such as every day or every two days.

▶   Select Weekly to select a frequency of weeks and the days of the week that the export takes place, such as every two weeks on Monday, Wednesday, and Friday.

▶   Select Monthly to select a specific day in the month and an interval of months, such as the second day every three months or the third Sunday every month.

3   In the Daily Frequency section, select the time and frequency of the backup on the days you selected. Perform one of the following:

▶   To perform the export only once, select "Occurs once at" and select the time.

> ▶ To perform the export several times a day, select "Occurs every" and select the frequency and the period of time that the export occurs.

> **Note:** For the purpose of scheduling, midnight is 12:00 A.M. and noon is 12:00 P.M. The day ends at 11:59 P.M.

4 In the Duration section, indicate the period that the task takes place. If you do not want to specify a limited time, leave the fields at their defaults of the current date as the start date and December 31, 2099 as the end date.

5 Click Update to accept the changes. The schedule portion of the tab indicates the new setting and the last time it was run.

Running SQL Server Agent

To perform scheduled tasks, the SQL Server Agent must be running. Use SQL Server Enterprise Manager to perform the following:

1 In the Tree pane, expand the listing of the SQL Server. Expand the Management folder to reveal SQL Server Agent as shown:
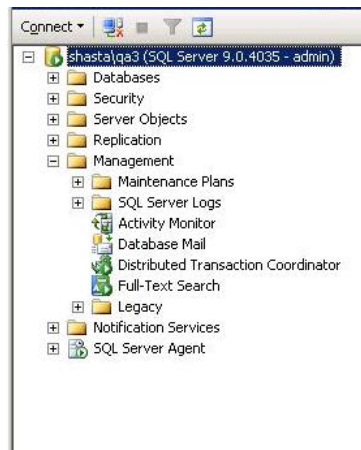
2 If a green arrow appears in the icon for SQL Server Agent, it is running. If there is a red square, you need to start the service. Right-click the icon, and from the menu that appears, select Start.

If you need information about using SQL Server Agent automatically, see *SQL Server Books Online*.

Viewing the scheduled events   Data update tasks are added to the jobs for the SQL Server Agent. To view these tasks, use SQL Server Enterprise Manager.

Setting SQL Server Schedules for a SQL Server Express Installation

Introduction   When using SQL Server Express, the SQL Server Agent is not installed. You can set an update schedule for Business Insights Dashboard so that the data is updated automatically at the intervals you set. For more information about SQL Server scheduling, see *SQL Server Books Online*.

Procedure   Follow these steps to set the schedule:

1   Select Windows Start menu > All Programs > Accessories > System Tools > Task Scheduler.
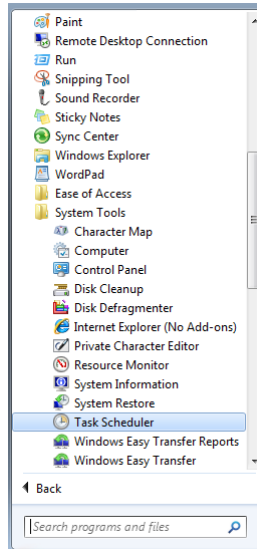
2   Select the Actions menu and click Create Task. The following window appears:
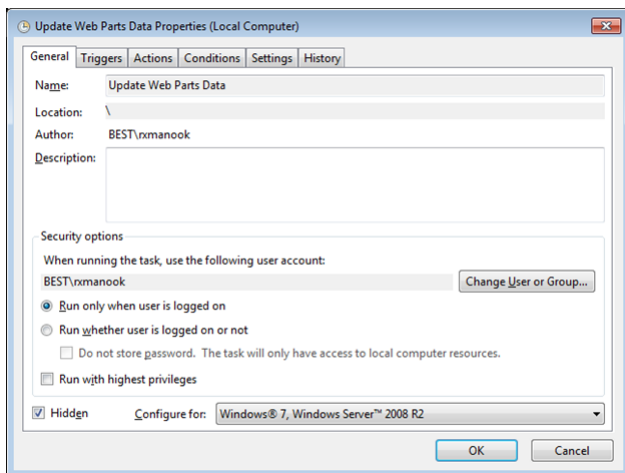
3   At the Name field, enter Update Web Parts Data.

4 In the Security Options section, select the appropriate user or group. If you are using a trusted connection, make sure this user or group can log on to SQL Server.

5 Select the Hidden check box and configure the security options for the appropriate operating system.

6 Click the Triggers tab. Click New to configure the schedule type. The following window appears:

7 Select the appropriate settings to run the schedule and click OK.
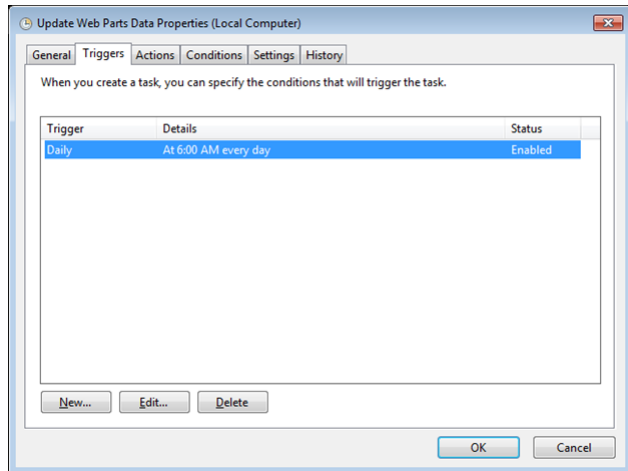
Click Edit, to change the schedule configuration.

8 Click the Actions tab. Click New to configure the task to execute. The following window appears:
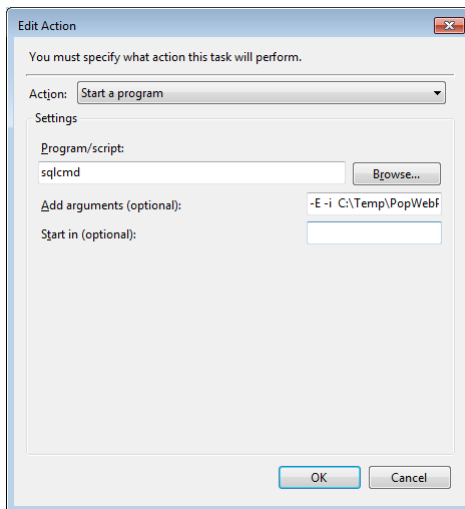
9   Select the action and script to execute using a trusted connection and click OK.
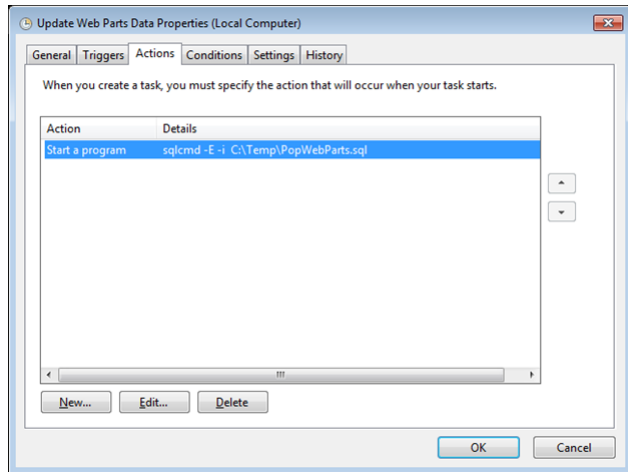
You can also assign the script to execute for a specific SQL user.

10   Click OK.

PopWebParts.sql contents

The PopWebParts.sql file contains the following web parts. Replace the sage500_app with the appropriate database name.

- Exec sage500_app..spbpPopWPFinancialInfo
- Exec sage500_app..spbpPopWPCashFlowProjections
- Exec sage500_app..spbpPopWPInvoiceAnalysis
- Exec sage500_app..spbpPopWPInventoryAnalysis
- Exec sage500_app..spbpPopWPIncomeTrend
- Exec sage500_app..spbpPopWPTopInventoryItems
- Exec sage500_app..spbpPopWPTopCustomers
- Exec sage500_app..spbpPopWPDailySalesOrdSummary
- Exec sage500_app..spbpPopWPCustTransSummaExec

Continuing the System Configuration

Where to find procedures

To continue configuring the Business Insights Dashboard, use Sage 500 ERP to set options for the Web site. See the following chapters:

- For Business Insights Dashboard, see Chapter 5, *Configuring Business Insights Dashboard*.

Configuring Business Insights Dashboard

5

This chapter shows how to configure Business Insights Dashboard. See the following sections for instructions on performing these tasks. For more information about individual windows, see the Business Insights Dashboard Help system.

In this chapter

Configuring the Dashboard for Users
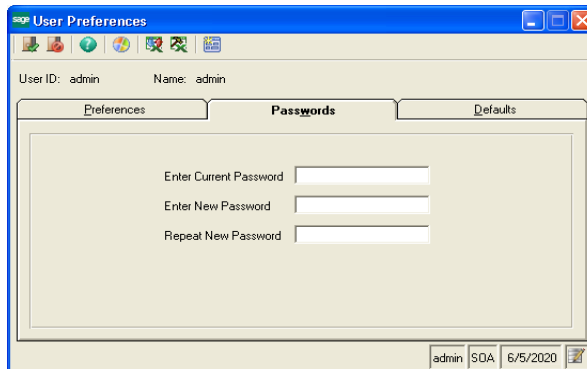
Introduction   Now that you have installed Business Insights Dashboard, use the instructions in this section to set up users, set up the Dashboard URL for users to access the Web site, set security for Web Parts, and update data.

Setting Up the Users

Setting Up Users   After installing Business Insights Dashboard, make sure the users running the Business Insights Dashboard Web site are set up in the Maintain Users task in Sage 500 ERP. For more information about setting up users in Sage 500 ERP, refer to your *Sage 500 ERP Installation and System Configuration* guide.

The Business Insights Dashboard Forms Authenticated users (Dashboard users who are not Windows authenticated) must have passwords set up for their Sage 500 ERP accounts.

Users who are Windows authenticated into the Dashboard Web site must also be Sage 500 ERP users. Use Maintain Users to add a user.
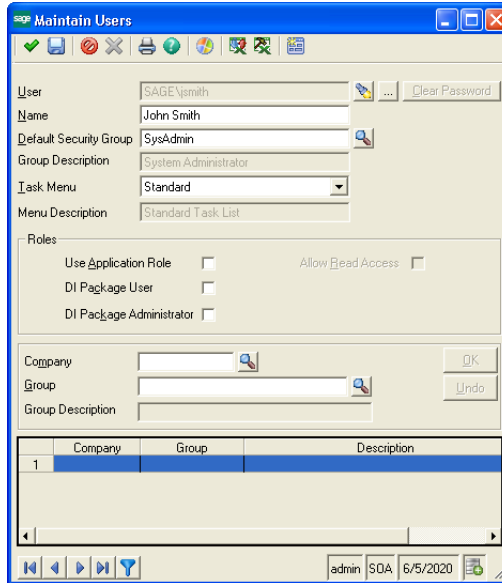
Updating Data

Updating data   The data that appears in the Business Insights Dashboard site is updated according to the schedule you set in Set Up CI Options. You can also generate manual updates. For more information, see *Setting Site Options in Sage 500 ERP* on page 40.

Setting security for Web parts Use Maintain Security Groups (accessed from the System Manager module) to establish permissions for the Web Parts available in Business Insights Dashboard. For more information, refer to your *Sage 500 ERP Installation and System Configuration* guide.
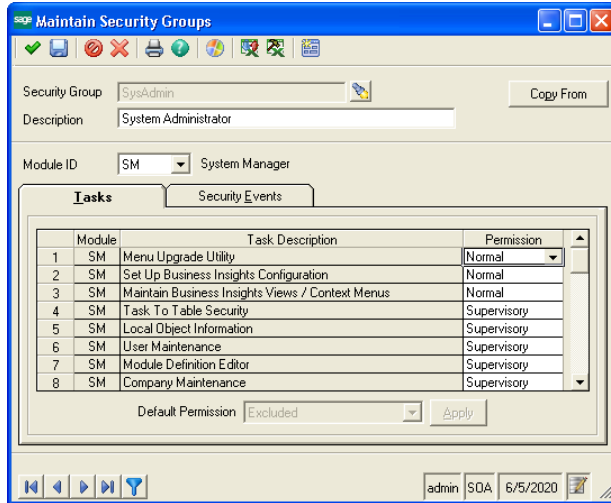
Setting Up Security for Dashboard Actions

In addition to the permissions for each Web part task, the Dashboard Enter Shared Scope and Dashboard Add Content tasks relate to the maintenance actions of the Dashboard pages.

Dashboard Add Content A user with normal or supervisory permissions to Dashboard Add Content can add or delete content on the Dashboard page in Personal view.

A user who is also granted normal or supervisory permissions to Dashboard Enter Shared Scope can add or delete content from the Shared view on the Dashboard page.

Dashboard Enter Shared Scope

A user with normal or supervisory permissions can perform the following tasks in Dashboard Enter Shared Scope:

- View the Dashboard page in Shared view, which only shows the Web part content that is shared among all users.

  Note: Users given permission to Enter Shared Scope should also be given normal or supervisory permission to Dashboard Add Content for maintaining shared content.

- Create additional Dashboard pages, name them, and provide display names and descriptions.

- Change display name and description of the standard Dashboard page.

- Delete any additional Dashboard pages. The standard Dashboard page cannot be deleted.

Setting up the Dashboard

Introduction

After installing the Business Insights Dashboard as an IIS application you can set a shortcut on the user's instance of Internet Explorer using the following convention:

\\servername\BusinessInsightsDashboard

You should also set the Business Insights Dashboard Web site as a trusted site in Internet Explorer.

Customizing Dashboard Web pages

You can customize the Business Insights Dashboard Web site with Microsoft Visual Studio 2010.

Configuring ASP.NET

Introduction   This section contains information about registering a different server or database to contain the ASP.NET tables, changing the property settings in ASP.NET without modifying the application code, and securing the Business Insights Dashboard connection.

ASP.NET Database Registration

Registering a different server or database   If you need to register a different server or database to contain the ASP.NET tables, perform the following tasks:

1   Run the aspnet_regsql.exe file located on your Web server at: drive:\\Windows\Microsoft.net\Framework\VersionNumber

2   In SQL Management Studio, load and execute the ASPNET_REGSAGE500.SQL script found on the Installation DVD for Business Insights Dashboard in the InternetApps\SupportFiles folder.

3   Reset IIS. You can restart the server or run the IISReset.exe command.

4   Verify the Web site has been started by launching the Internet Information Services (IIS) Manager from Administrative Tools. If the Web site has not been started, select the default Web site and click the Run button.

To enable Business Insights Dashboard to use the newly created ASP.NET database, change the ASP.NET LocalSqlServer connections string. For more information, see *Setting the Connection Strings*

Configuring ASP.NET Settings

Configuration resources

The ASP.NET configuration settings allow you to change the properties of the ASP.NET application without modifying the application code. These settings are stored in a Web.Config file for the Business Insights Dashboard application. Revising any configuration settings updates this configuration file.

For more information about configuring ASP.NET, visit the following sites:

  ▶ For Internet Information Services (IIS) versions 5 and 6, go to: http://msdn.microsoft.com/en-us/library/ms186189(VS.80).aspx

  ▶ For Internet Information Services (IIS) version 7, go to: http://msdn.microsoft.com/en-us/library/ms178477.aspx

Configuration settings

This section describes the configuration settings that can be changed for your Business Insights Dashboard application.

Procedure

Follow these steps to modify ASP.NET configuration settings in Windows XP and Windows 2003 Server:

1   Open the Internet Information Services (IIS) Manager program.

2   Select the name of the installed Business Insights Dashboard, which resides under Web Sites\Default Web Sites.

3   From the menu, select Action, and then Properties.

4 Click the ASP.NET tab.
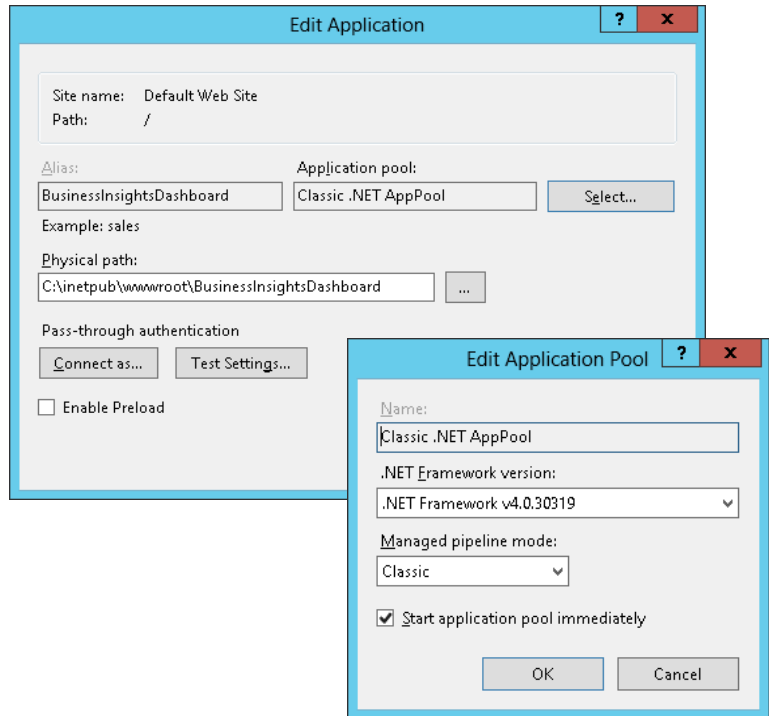
5 Click Edit Configurations.

Procedure Follow these steps to modify ASP.NET configuration settings in Windows Vista, Windows 2008 Server, Windows Server 2012, Windows 7, and Windows 8:

1 Open the Internet Information Services (IIS) Manager program from the Administrative Tools folder.

2 Select the name of the installed Business Insights Dashboard, which resides under Sites\Default Web Site.

3 On the Actions pane click Basic Settings. Ensure the Edit Application settings are as follows:
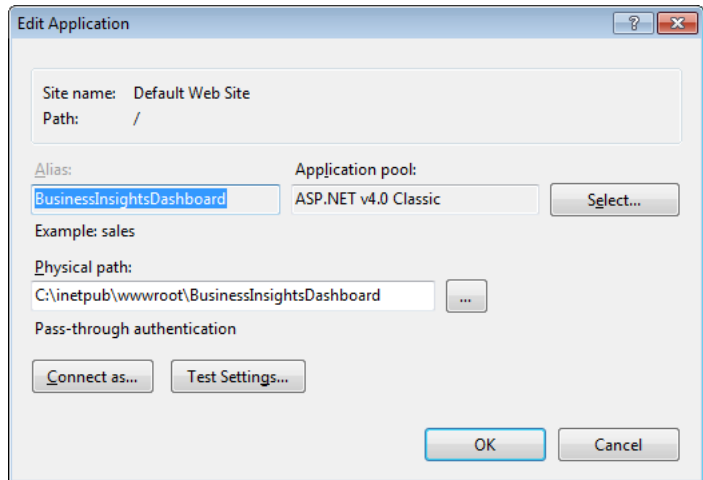
4 Double-click Authentication setting under BusinessInsightsDashboard and ensure Anonymous Authentication is enabled.

Setting the Connection Strings

Introduction   The following section contains information and procedures for setting and modifying LocalSqlServer and SAGE500App connection strings.

Setting the LocalSqlServer connection string   Use the following procedures to set the LocalSqlServer connection string. LocalSqlServer is used by ASP.NET to connect to the data used by ASP.NET.

Procedure   Follow these steps to set the LocalSqlServer connection string in Windows XP and Windows 2003 Server:

1 In the ASP.NET Configuration Settings window, click the General tab.

2 In the Connection string section, select the LocalSqlServer connection string.

3 Click Edit.

4 In the Edit/Add Connection String window, modify the connection string as needed.

Data Source: Name of the SQL Server

Initial Catalog: Name of the database in SQL Server

User ID: User ID used to log on to the server

Password: Password used to log on to the server

Note: Do not modify or remove Application Name.

5 Click OK.

Procedure

Follow these steps to set the LocalSqlServer connection string in Windows Vista, Windows 2008 Server, Windows Server 2012, Windows 7, and Windows 8:

1 Double-click Connection Strings setting under BusinessInsightsDashboard to set up LocalSqlServer connection string.

2 In the Connection Strings section, select the LocalSqlServer connection string.

3 On the Actions pane click Edit.

4 In the Edit Connection String window, modify the connection string as needed.

Data Source: Name of the SQL Server

Initial Catalog: Name of the database in SQL Server

User ID: User ID used to log on to the server

Password: Password used to log on to the server

Note: Do not modify or remove Application Name.

5 Click OK.
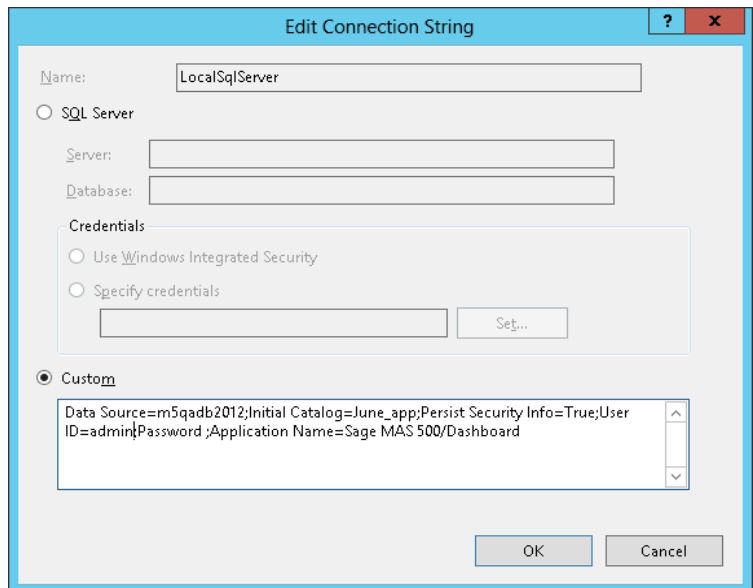
Modifying the SAGE500App connection string

Use the following procedures to modify the SAGE500App connection string. SAGE500App is used to access Sage 500 ERP data.

Procedure

Follow these steps to modify the SAGE500App connection string in Windows XP and Windows 2003 Server:

1 In the ASP.NET Configuration Settings window, click the General tab.

2 In the Connection string section, select the SAGE500App connection string.

3 Click Edit.

4 In the Edit/Add Connection String window, modify the connection string as needed.

Data Source:	Name of the SQL Server
Initial Catalog:	Name of the database in SQL Server
User ID:	User ID used to log on to the server
Password:	Password used to log on to the server

Note: Do not modify or remove the Application Name and Persist Security Info entries.

5 Click OK and close the ASP.NET Configuration Settings window.

6 On the Properties window, click OK.

7 Close IIS.

Procedure

Follow these steps to modify the SAGE500App connection string in Windows Vista, Windows 2008 Server, Windows Server 2012, Windows 7, and Windows 8:

1 Double-click Connection Strings setting under BusinessInsightsDashboard to set up SAGE500App connection string.

2 In the Connection Strings section, select the SAGE500App connection string.

3 On the Actions pane click Edit.

4　In the Edit Connection String window, modify the connection string as needed.

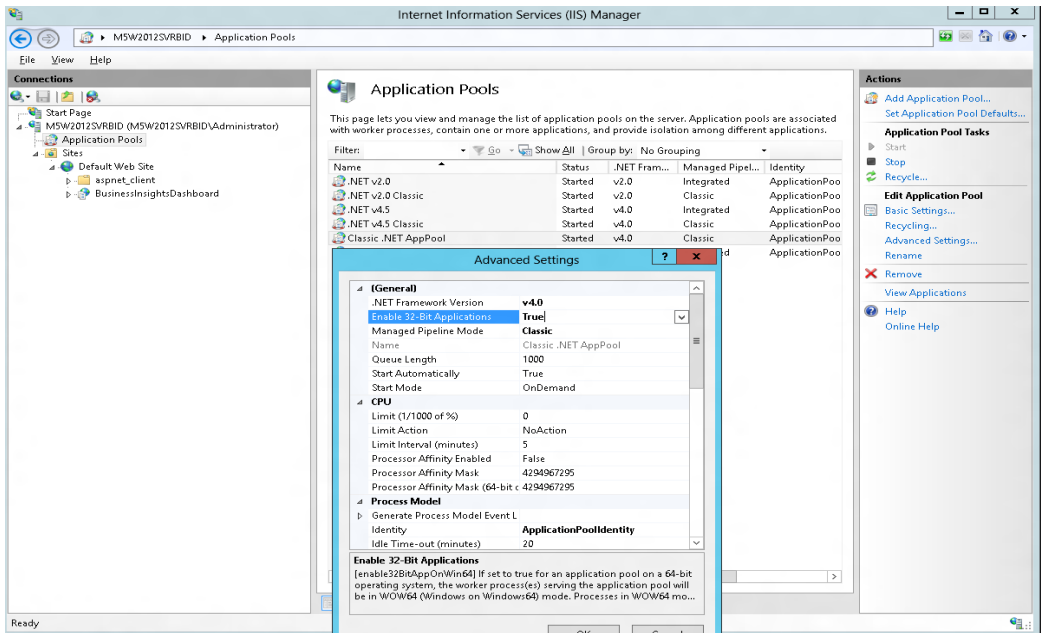Data Source:　Name of the SQL Server

Initial Catalog:　Name of the database in SQL Server

User ID:　User ID used to log on to the server

Password:　Password used to log on to the server

Note: Do not modify or remove Application Name.

5　Click OK.

Securing the Business Insights Dashboard Connection

Security measures Business Insights Dashboard stores two connections to the database in the web.config file. One connection, LocalSqlServer, is used by ASP.NET to connect to the data used by ASP.NET. The other connection, SAGE500App, is used to access Sage 500 ERP data. Typically, these connections are created using a SQL Server user ID and are stored in the web.config file. This is usually sufficient because the web.config file is protected by Internet Information Services (IIS) Manager by not allowing the file to be transferred to a client. There also is protection from access outside of IIS using normal NTFS protection.

Additional steps can be taken to further increase the security of these connection strings, such as encrypting the connection strings in the web.config file, implementing a trusted subsystem configuration, and impersonating the application level. The following section describes these methods.

Encrypting the connection string Encrypting the connection string is not difficult to do and will protect the connection string information from being accessed in case the web.config file is compromised.

> **Note:** The steps in the following procedure can be implemented independently.

To encrypt the connection string:

1 From the machine that is running the Web server, run the following command line utility:

aspnet_regiis -pe "connectionStrings" -app "/BusinessInsightsPortal" -prov "DataProtectionConfigurationProvider"

- aspnet_regiis is located in the %systemroot%/ Microsoft.Net/Framework/<version> folder where %systemroot% is typically C:\Windows and <version> is the version of the .NET framework in use.

- /BusinessInisghtsDashboard is the name of the web application.

- DataProtectionConfigurationProvider is implemented in .NET Framework 4.0 and provides the encryption facilities.

The .NET Framework 4.0 Software Developer's Kit (SDK) provides two different Protected Configuration providers, which you use with the aspnet_regiis.exe tool:

- RSAProtectedConfigurationProvider. This is the default provider and uses the RSA public key encryption to encrypt and decrypt data. Use this provider to encrypt configuration files for use on multiple Windows Servers in a Web farm.

- DPAPIProtectedConfigurationProvider. This provider uses the Windows Data Protection API (DPAPI) to encrypt and decrypt data. Use this provider to encrypt configuration files for use on a single Windows Server.

Implementing a trusted subsystem

In the trusted subsystem model, the database server trusts the Web application identity and allows the Web application to make calls on behalf of the original caller.

Procedure

To implement a trusted subsystem:

1 Modify the SAGE500App Connection String to use integrated security as follows:

Data Source=YourSQLServerMachine;Initial Catalog=SAGE500_app;Integrated Security=True

Where YourSQLServerMachine is the name of the SQL Server, and SAGE500_app is the name of the SAGE500 database.

2 Grant access to the SQL Server for the identity of the ASP.NET application. Typically this is <machinename>/ASPNET. You can determine the correct user ID using Task Manager and looking at the User Name for the aspnet_wp object.

Allowing the <machinename>/ASPNET user access to your SQL Server can pose a potential risk because anyone using ASP.NET on the Web server will have access to the SQL Server. The potential of risk can be reduced by also impersonating the application level.

Impersonating application levels

A good practice for building a trusted subsystem is to also use the application impersonation method. After implementing a trusted subsystem, isolate the access to the database for the Business Insights Dashboard application. This allows the Business Insights Dashboard application to have its own identity, separate from the identity used by other ASP.NET applications. This allows you to grant database access tailored to the needs of the Business Insights Dashboard application.

To implement application impersonation, set the impersonation identity for the Business Insights Dashboard application by accessing the IIS properties for the Business Insights Dashboard application and updating the ASP.NET configuration to set the local impersonation setting as shown in the following window.
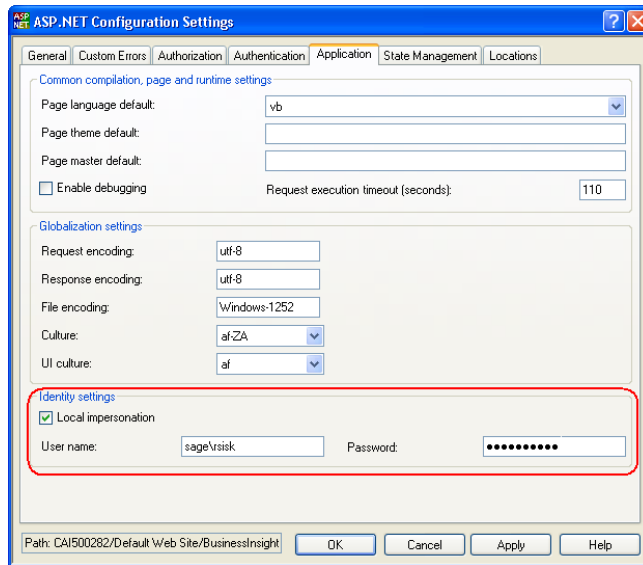
For more information, refer to the following article on the Microsoft Web site:

http://msdn2.microsoft.com/en-us/library/ms998292.aspx

Protecting the LocalSqlServer connection string

In Business Insights Dashboard, AppRole functionality is not supported for the LocalSqlServer connection string. The best approach for securing this connection string is to limit the access to only the database objects accessed using this connection. Use the following procedure to secure this connection string.

Procedure

To secure the LocalSqlServer connection string:

1 Create a database role (DBRole) that has access to objects that start with aspnet.

2 Select a Windows Authenticated user ID that is a Sage 500 ERP user, has the SAGE500 AppRole option selected, and is a member of the database role created in step 1.

3 Modify the LocalSqlServer and SAGE500App connection strings to use Integrated Security.

4 Modify the Dashboard application to impersonate the user selected in step 2.

5 Encrypt the web.config file.

Encrypting and decrypting connection strings

You can encrypt and decrypt sections in the web.config file using the aspnet_regiis.exe command-line tool, which can be found in the %WINDOWSDIR%\Microsoft.Net\Framework\version folder, normally located at:

C:\Windows\Microsoft.Net\Framework\v4.0.30319.

Encryption:

aspnet_regiis.exe -pef "connectionStrings" "C:\Inetpub\wwwroot\BusinessInsightsDashboard" -prov "DataProtectionConfigurationProvider"

-- or -

aspnet_regiis.exe -pe "connectionStrings" -app "/ BusinessInsightsDashboard" -prov "DataProtectionConfigurationProvider"

Decryption:

aspnet_regiis.exe -pdf "connectionStrings" "C:\Inetpub\wwwroot\ BusinessInsightsDashboard"

-- or -

aspnet_regiis.exe -pd "connectionStrings" -app "/ BusinessInsightsDashboard"

Note: You may need to replace BusinessInsightsDashboard with the name given the site at time of install.

Encrypting still allows you to use the ASP.NET configuration to maintain the connection strings through the ASP.NET configuration.

Handling Large Sets of Data

Overview   To handle performance issues with large sets of data, Business Insights Explorer data within the Business Insights Dashboard uses the Top feature of SQL Server to limit the amount of data delivered back to the server and client. This value is stored as an application setting and can be modified at the installed site. The following steps describe how to modify this value.

Procedure   Follow these steps to modify the Top value in Windows XP and Windows 2003 Server:

1   In the ASP.Net Configuration Settings window, click the General tab.

2   Select BIExplorerTopRows from the Application settings section.

3   Click Edit.

4   Type the integer value, and then click OK.

5   In the ASP.NET Configurations Settings window, click OK.

6   In the Properties window, click OK.

7 Close IIS.
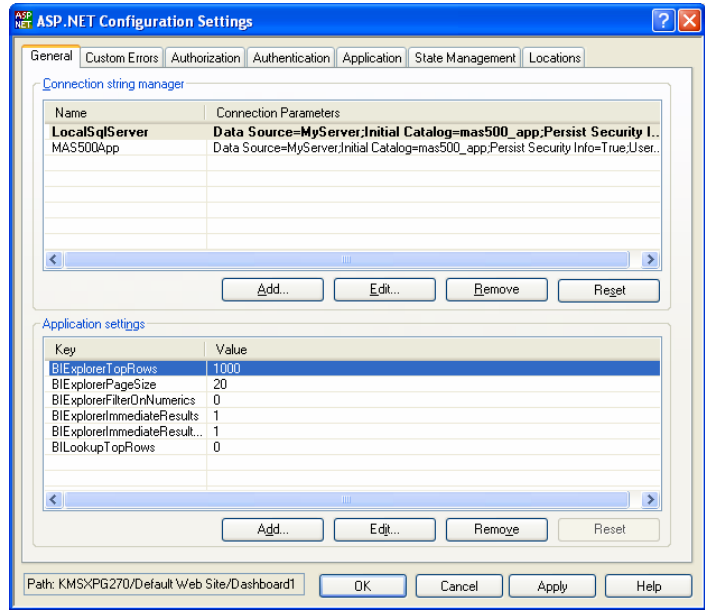
You can also handle large sets of data for the selection filter in the Lookup window. The default is 0 to return all the rows, but can be modified by selecting BILookupTopRows from the Application settings section.

Procedure   Follow these steps to modify the Top value in Windows Vista, Windows 2008 Server, Windows Server 2012, Windows 7, and Windows 8:

1 Double-click Application Settings under BusinessInsightsDashboard to maintain the Top value.

2 In the Application Settings section, select BIExplorerTopRows.

3 Click Edit.

4 Type the integer value, and then click OK.
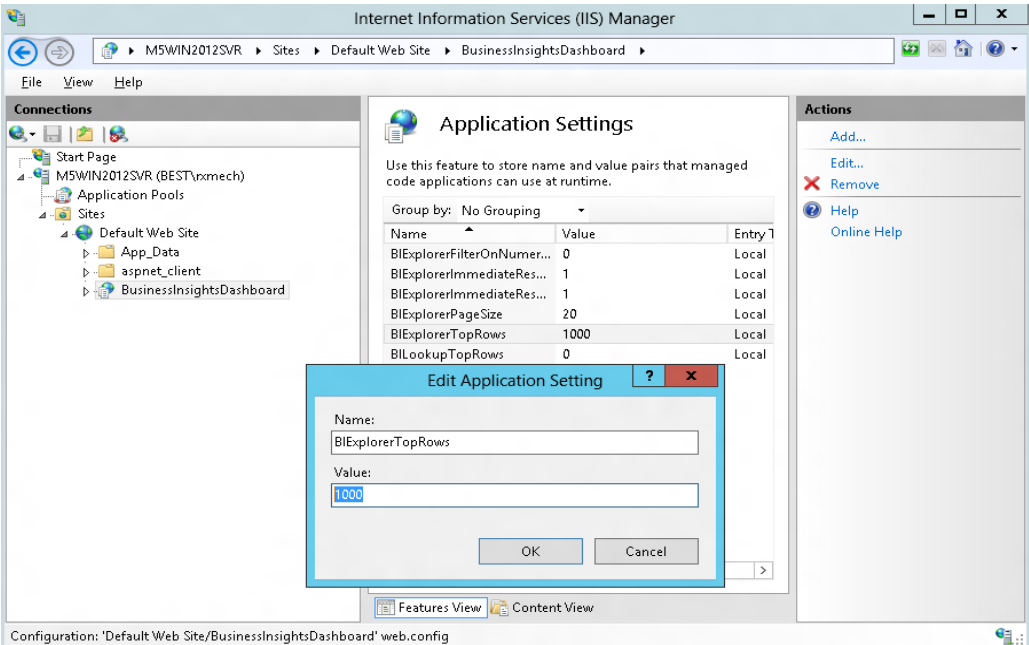
5 Close IIS.

You can also handle large sets of data for the selection filter in the Lookup window. The default is 0 to return all rows, but it can be modified by selecting BILookupTopRows from the Application Settings section.

Displaying a Specific Number of Rows

Overview   Business Insights Explorer within Business Insights Dashboard uses a paging model to display only a specified number of rows to the user. The default size is 20 but can be modified at the installed site. The following procedure describes how to modify this value.

Procedure   Follow these steps to modify the number of rows to display in Windows XP and Windows 2003 Server:

1   In the ASP.NET Configuration Settings window, click the General tab.
2   Select BIExplorerPageSize from the Application settings section.
3   Click Edit.
4   Type the integer value, and then click OK.
5   In the ASP.NET Configurations Settings window, click OK.
6   In the Properties window, click OK.

7 Close IIS.
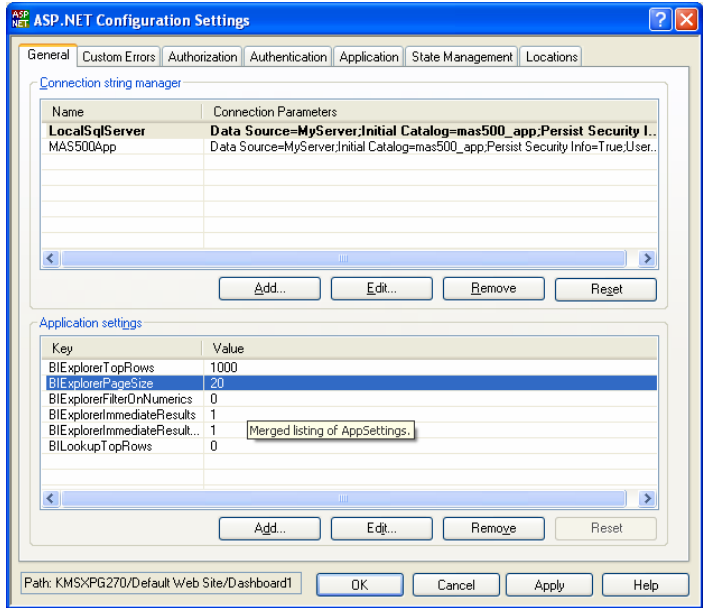
Procedure   Follow these steps to modify the number of rows to display in Windows Vista, Windows 2008 Server, Windows Server 2012, Windows 7, and Windows 8:

1 Double-click Application Settings under BusinessInsightsDashboard to maintain the Top value.

2 In the Application Settings section, select BIExplorerPageSize.

3 Click Edit.

4 Type the integer value, and then click OK.

5 Close IIS.
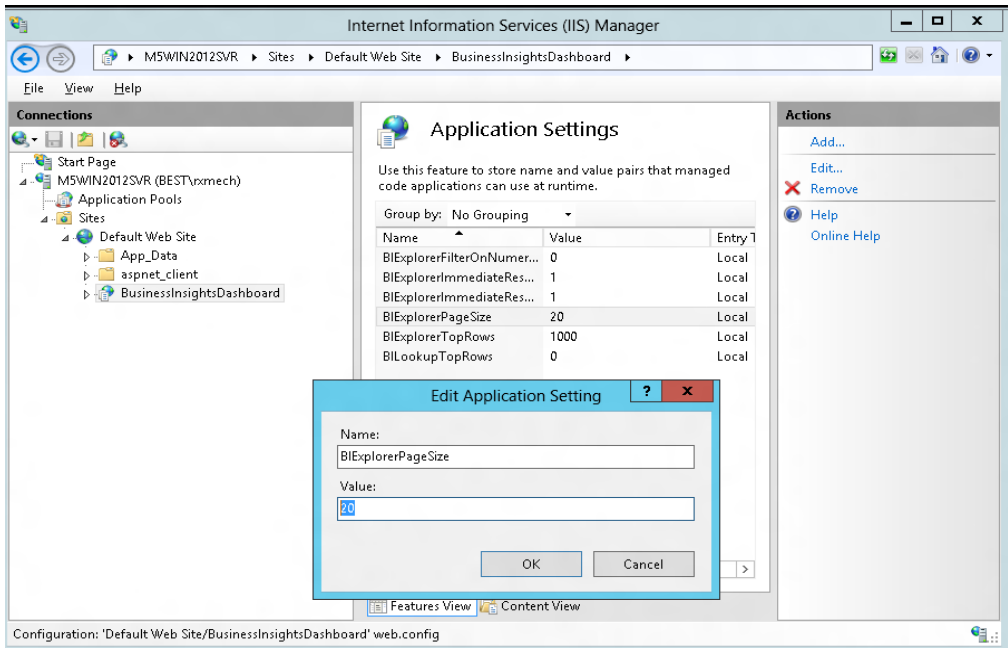
Modifying the Filter On Numerics Setting

Overview   The Filter on Numerics setting is used to enable or disable filtering on numeric data. The version of the Infragistics control used by the Business Insights Dashboard has problems with numerics formatted with a group separator or percent symbol. Performing the following procedure provides a way to allow filtering on numeric values; however, doing so does not allow group separators.

> **Note:** Percentage columns do not allow filtering.

Procedure Follow these steps to modify the Filter on Numerics setting in Windows XP and Windows 2003 Server:

1 In the ASP.NET Configuration Settings window, click the General tab.

2 Select BIExplorerFilterOnNumerics in the Application settings section.

3 Click Edit.

4 Type the bit integer (0, 1) value, and then click OK.

Note: Percentage columns do not allow filtering.

5 In the ASP.NET Configurations Settings window, click OK.

6 In the Properties window, click OK.

7 Close IIS.
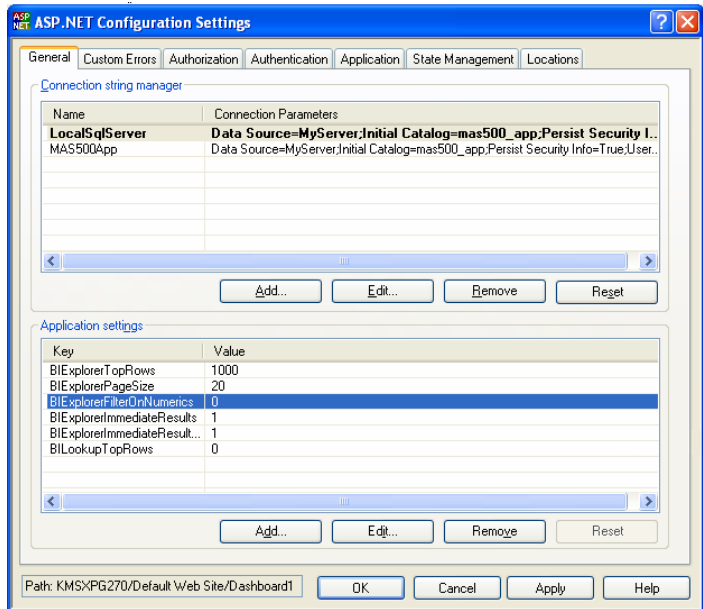
Procedure

Follow these steps to modify the Filter on Numerics setting in Windows Vista, Windows 2008 Server, Windows Server 2012, Windows 7, and Windows 8:

1. To maintain the Top value, double click Application Settings under BusinessInsightsDashboard.

2. In the Application Settings section, select BIExplorerFilterOnNumerics.

3. Click Edit.

4. Type the bit integer (0, 1) value, and then click OK.

   Note: Percentage columns do not allow filtering.

5. Close IIS.
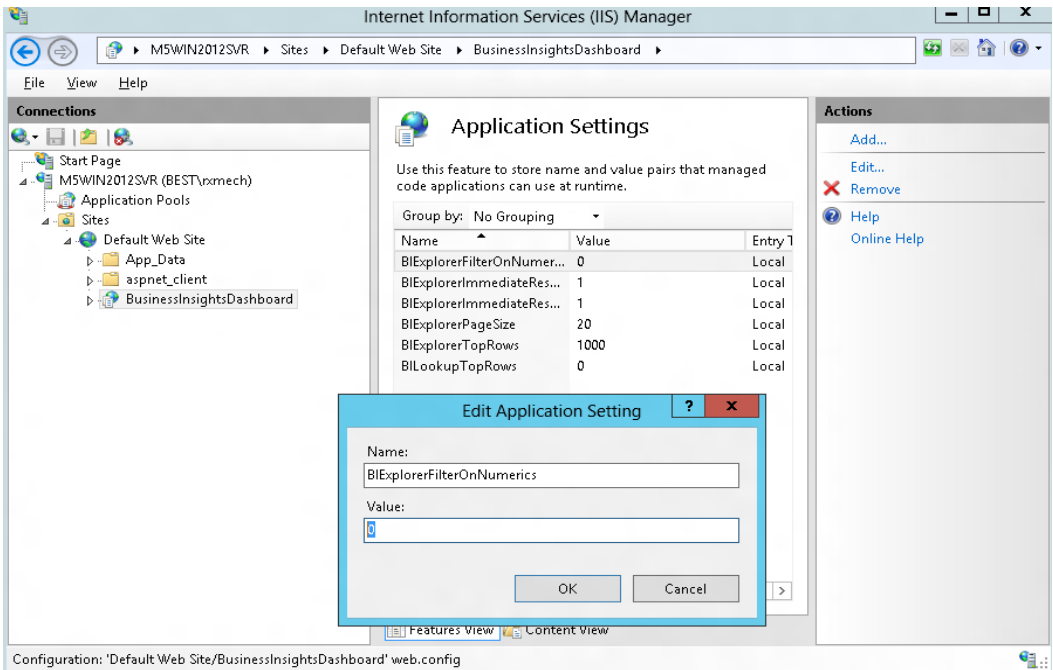
Modifying the Immediate Results Setting for Explorer

Overview   The Immediate Results setting is used to enable or disable the ability to view Explorer results immediately when selecting the Explorer task. A Web site can disable immediate results when there is the possibility of large data sets being returned, or to allow being able to enter a filter condition to limit the results instead of viewing the larger set of data first. The Sage 500 ERP client default setting for immediate results is enabled as is the Business Insights Explorer Immediate Results setting. You can disable the immediate results for the client as a setting per Explorer task. The Business Insights Dashboard observes this setting when it is disabled.

> **Note:** To view immediate results, both client and Dashboard setting must be enabled.

Procedure   Follow these steps to modify the Immediate Results setting in Windows XP and Windows 2003 Server:

1   In the ASP.Net Configuration Settings window, click the General tab.

2   Select BIExplorerImmediateResults from the Application settings section.

3   Click Edit.

4   Type the bit integer (0, 1) value, and then click OK.

5   In the ASP.NET Configurations Settings window, click OK.

6   In the Properties window, click OK.

7 Close IIS.
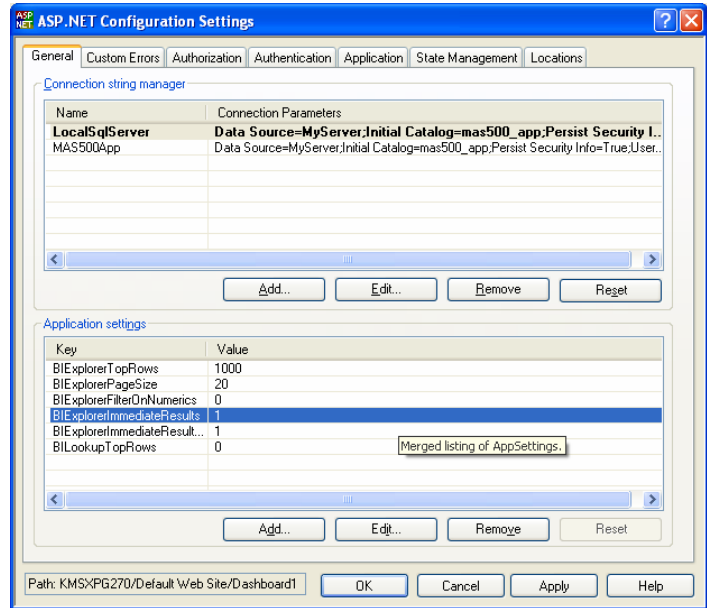
Procedure

Follow these steps to modify the Immediate Results setting in Windows Vista, Windows 2008 Server, Windows Server 2012, Windows 7, and Windows 8:

1 To maintain the Top value, double click Application Settings under BusinessInsightsDashboard.

2 In the Application Settings section, select BIExplorerImmediateResults.

3 Click Edit.

4 Type the bit integer (0, 1) value, and then click OK.

5 Close IIS.
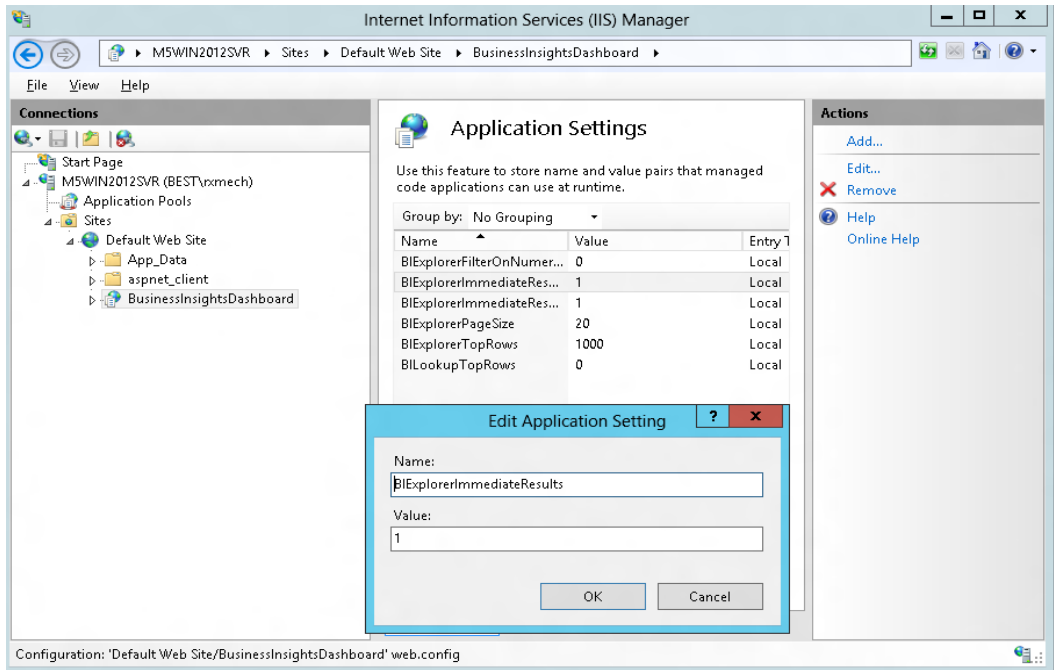
Modifying the Immediate Results for Filter Setting

Overview

The Immediate Results for Filter setting is used to determine if the data is loaded when a filter is applied for the Explorer view. When a filter exists and the value for this setting is 1, this setting will display the data for the filtered view. Setting this value to 0 is designed to force the user to select the Refresh button to view the results. This setting is ignored when Drill Into is performed to another data view. The Business Insights Dashboard observes this setting when it is disabled.

> **Note:** To view immediate results, both client and Dashboard setting must be enabled.

Procedure   Follow these steps to modify the Immediate Results for Filter setting in Windows XP and Windows 2003 Server:

1 In the ASP.Net Configuration Settings window, click the General tab.

2 Select BIExplorerImmediateResultsForFilter from the Application settings section.

3 Click Edit.

4 Type the bit integer (0, 1) value, and then click OK.

5 In the ASP.NET Configurations Settings window, click OK.

6 In the Properties window, click OK.

7 Close IIS.
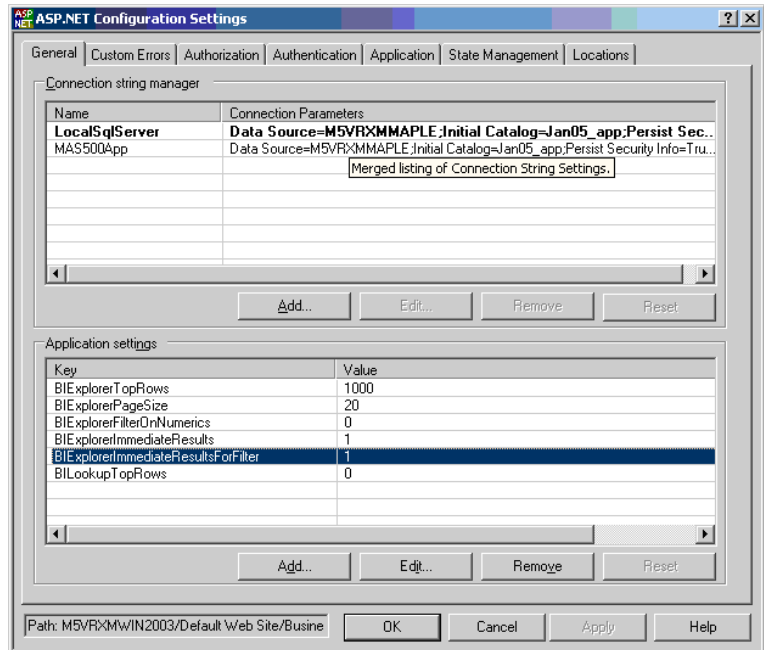
Procedure Follow these steps to modify the Immediate Results for Filter setting in Windows Vista, Windows 2008 Server, Windows Server 2012, Windows 7, and Windows 8:

1 Double click Application Settings under BusinessInsightsDashboard to maintain the Top value.

2 In the Application Settings section, select BIExplorerImmediateResultsForFilter.

3 Click Edit.

4 Type the bit integer (0, 1) value, and then click OK.

5 Close IIS.
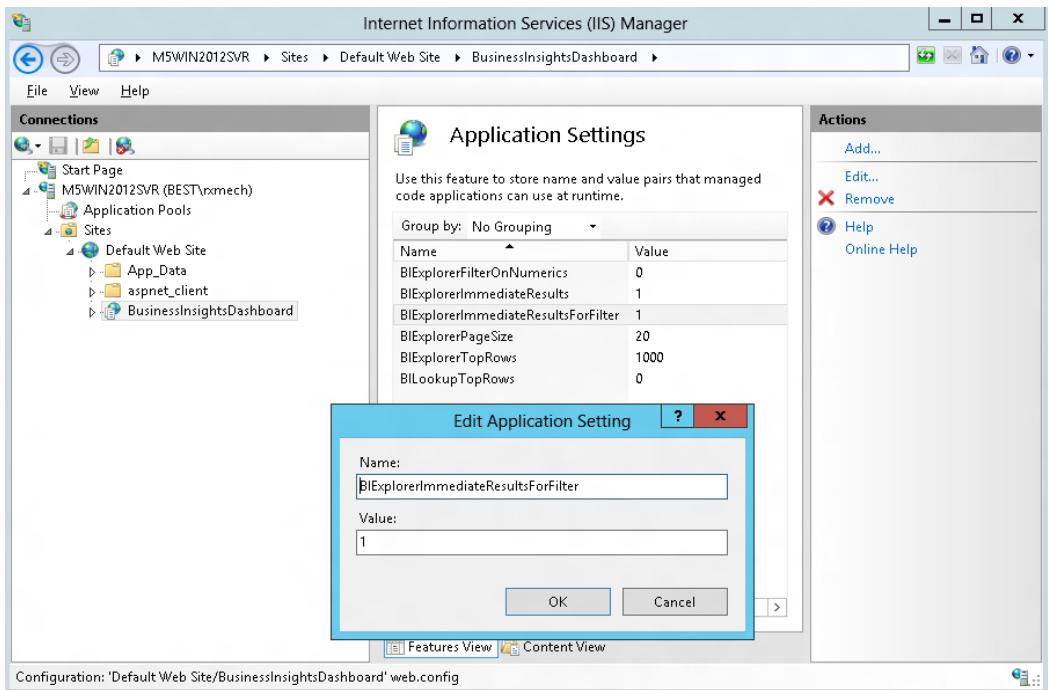
Enabling or Disabling Custom Errors

Overview

The capability to implement custom errors is enabled to provide messages within Business Insights Dashboard. If the custom error messages do not provide enough detail, you can disable custom messaging, which allows you to provide more detailed descriptions to problems.

Custom error setting for IIS 5 and 6

To modify the custom error mode setting for Internet Information Services (IIS) versions 5 and 6

1   In the ASP.NET Configuration Settings window, click the Customer Errors tab.

2   At the Custom error mode field, select the custom error mode to use. The following mode options are available:

▶   On: Custom errors are displayed for local and remote clients.

▶   Off: Detailed custom errors are displayed for local and remote clients.

▶   Remote Only: Custom errors are displayed for remote clients and detail errors are displayed for local clients. This option is the installation default option.

3 In the ASP.NET Configurations Settings window, click OK.
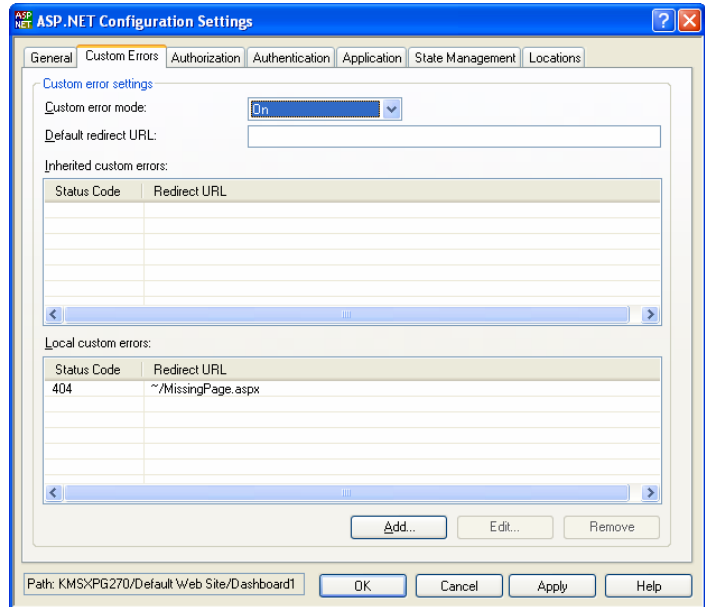
4 In the Properties window, click OK.

5 Close IIS.

Custom error setting for IIS 7

To modify the custom error mode setting for Internet Information Services (IIS) version 7

1 Locate the installation folder for the Business Insights Dashboard. Unless a virtual folder is used for this application, the folder is under the IIS Server hard drive as shown in the following example:

C:\Inetpub\wwwroot\BusinessInsightsDashboard

2 Open the web.config file in Notepad.

3 Locate the customErrors mode section in the web.config file as follows:

<customErrors mode="RemoteOnly">

4 Enter the custom error mode. The following options are available:

- ► On: Custom errors are displayed for local and remote clients.

- ► Off: Detailed custom errors are displayed for local and remote clients.

- ► RemoteOnly: Custom errors are displayed for remote clients and detail errors are displayed for local clients. This option is the installation default option.

5 Save the web.config file.

Modifying the Maximum Request Length Exceeded

Overview
An error may occur if the value for the MaxRequestLength is not large enough to allow for data to be posted back to the server. When custom errors are disabled, this error is identified by a "Maximum request length exceeded." error message.

Procedure
To resolve this problem, use the following method. The web.config file for Business Insights Dashboard requires modification.

1 Locate the installation folder for the Business Insights Dashboard. Unless a virtual folder is used for this application, the folder is under the IIS Server hard drive as shown in the following example:

C:\Inetpub\wwwroot\BusinessInsightsDashboard

2 Make a backup copy of the web.config file.

3 Open the web.config file in Notepad.

4 Locate the httpRuntime markup in the web.config file as follows:

<httpRuntime   maxRequestLength="12288" />

5 Adjust the value by some increment (for example, by 1024).

6 Save the web.config file.

7 You may need to continue to adjust this value until the problem is resolved.

> **Note:** When changing the maxRequestLength property, be aware of the setting provided for the executionTimeout property. This property sets the time (in seconds) for a request to attempt to execute to the server before ASP.NET shuts down the request (whether or not it is finished). You receive a timeout error notification in the browser if the time limit is exceeded. If you are going to permit larger requests, remember that they take longer to execute than smaller ones. If you increase the size of the maxRequestLength property, you should examine whether to increase the executionTimeout property as well.

For more information, refer to the following article on the Microsoft Web site:

http://support.microsoft.com/default.aspx?scid=kb;EN-US;295626

Troubleshooting Business Insights Dashboard

For more information about troubleshooting Business Insights Dashboard issues, see *Troubleshooting Business Insights Dashboard* on page 88.

Modifying the Execution Timeout Value

Overview   Execution timeout indicates that maximum number of seconds that a request is allowed to execute before being automatically shut down by ASP.NET.

Procedure   Follow these steps to modify the execution timeout value in Windows XP and Windows 2003 Server:

1   Launch Internet Information Services (IIS) Manager from the Administrative Tools folder.

2   Select the name of the installed Business Insights Portal, which resides under Web Sites\Default Web Sites.

3   Select Action menu > Properties.

4   Click the ASP.NET tab.

5   Click Edit Configurations.

6   Click the Application tab.

7   Enter the numbers in seconds for the Request execution timeout entry.
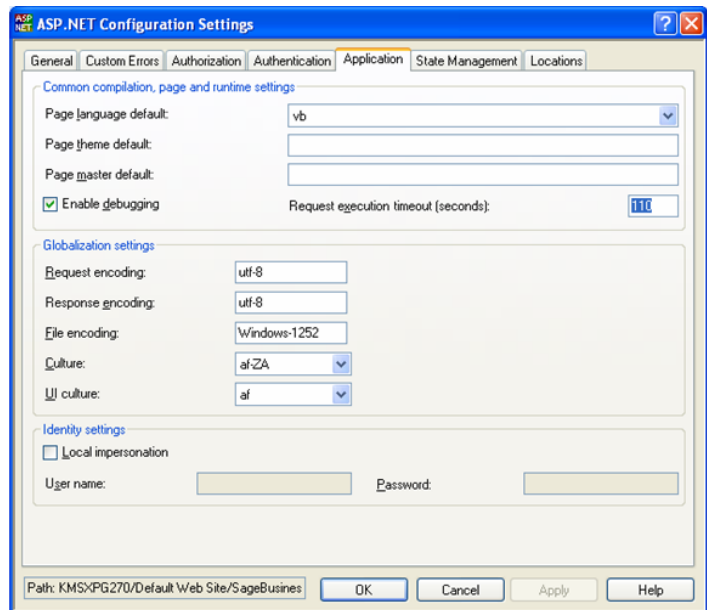
8 In the ASP.NET Configurations Settings window, click OK.

9 In the Properties window, click OK.

10 Close IIS.

Procedure

Follow these steps to modify the execution timeout value in Windows Vista, Windows 2008 Server, Windows Server 2012, Windows 7, and Windows 8:

1 Open the Internet Information Services (IIS) Manager program from the Administrative Tools folder.

2 Select the name of the installed Business InsightsDashboard, which resides under Sites\Default Web Site.

3 Double click ASP Settings under BusinessInsightsDashboard to set the Request Queue Time-out.

4 In the ASP Settings section, open the Limits Properties node.

5 Select Request Queue Time-out.

6 Enter appropriate time in hours, minutes and seconds. Enter 00:00:00 to wait indefinitely.

7 Close IIS.
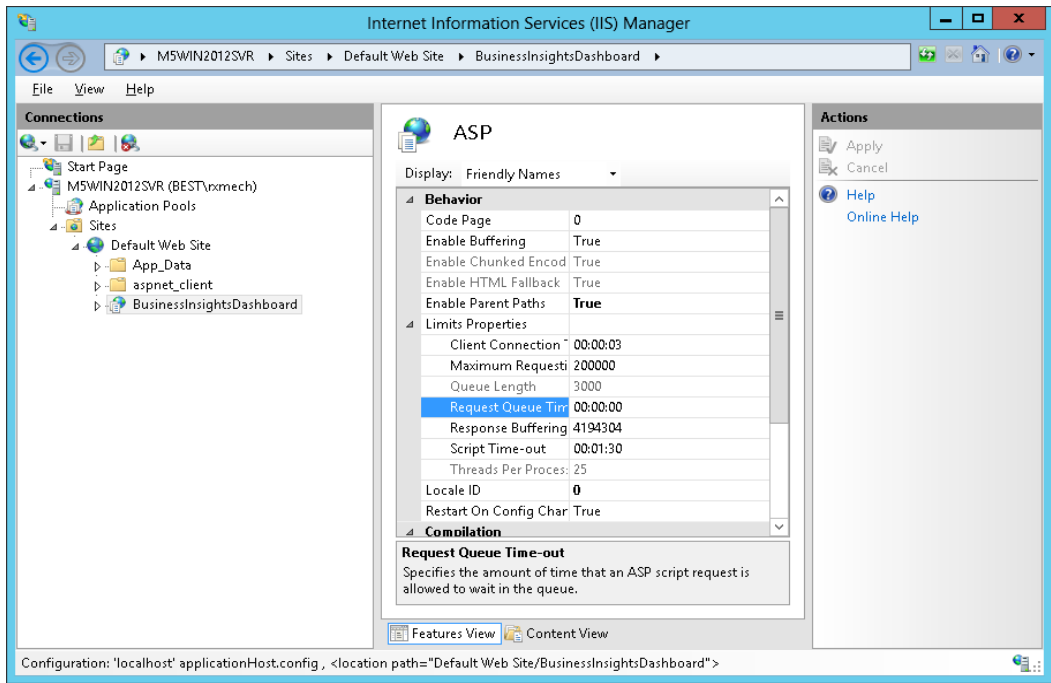
Troubleshooting Business Insights Dashboard

6

This chapter gives you suggestions on how to resolve issues if they occur with the Web site. You have a number of diagnostic tools you can use, including special pages provided with the Web site and Windows administrative tools such as Component Services and Computer Management.

In this chapter

Verify the Correct ASP.NET Version Is Used

Overview Use the following procedure to verify the version number when using ASP.NET with Business Insights Dashboard.

Windows XP and Windows 2003 Server Procedure To determine the ASP.NET version number

1   Launch Internet Information Services (IIS) Manager from Administrative Tools folder.

2   Select the name of the installed Business Insights Dashboard, which resides under Web Sites\Default Web Sites.

3   From the menu, select Action, and then Properties.

4   Click the ASP.NET tab.

5   Verify the site is using the supported ASP.NET version. For supported version information, see the *Sage 500 ERP Compatibility and Resource Guide*.
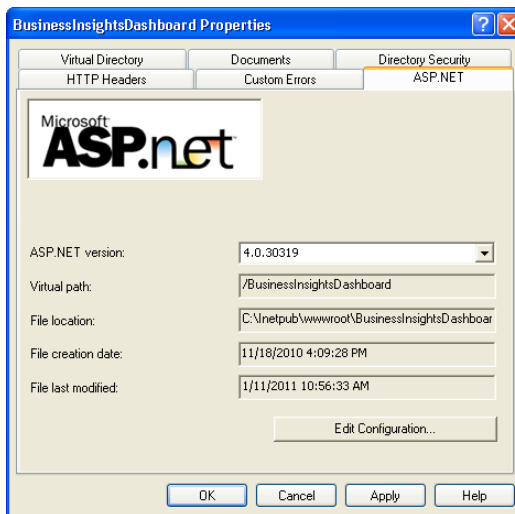
Windows Vista, Windows 2008 Server, and Windows 7 Procedure

To determine the ASP.NET version number

1   Open the Internet Information Services (IIS) Manager program from the Administrative Tools folder.

2   Select the name of the installed Business Insights Dashboard, which resides under Sites\Default Web Site.

3   On the Actions pane click Basic Settings. Verify the site is using the supported ASP.NET version. For supported version information, see the Sage 500 ERP *Compatibility and Resource Guide*.

Script Error Message on Login Page

Overview   The script error "Sys is undefined" appears only on the login page and does not affect the behavior of the site or the functionality of any of its features. This error can occur on Internet Explorer version 8 clients using Forms Authentication for the site installed on a Web server running Internet Information Services (IIS) version 7. To change the site application pool to Classic.Net, perform the following procedure.

Procedure   To change the site application pool to Classic.Net

1   Launch Internet Information Services (IIS) Manager from Control Panel > Administrative Tools.

2   Select the <server> > Sites > Default Web Site.

3   Right-click the BusinessInsightsDashboard site, and select Managed Applications > Advanced Settings.

4   In the General section of the property list, select the Application Pool row.

5   Click the ellipses button on the right section of the current value.

6   At the Application Pool field, select Classic.Net AppPool, and click OK.

7   Click OK again until you are returned to the Internet Information Services (IIS) Manager.

8   Close the IIS Manager window.

Additional information

For more information, refer to the following Microsoft Web sites:

- ▶ www.iis.net/learn/get-started/getting-started-with-iis

- ▶ http://msdn.microsoft.com/en-us/library/ microsoft.web.administration.applicationpool.manag edpipelinemode.aspx

- ▶ http://msdn.microsoft.com/en-us/library/ microsoft.web.administration.managedpipelinemode. aspx

Failure to Access the IIS Metabase

Overview   This problem may be caused by a missing ASP.NET windows process account. The ASP.NET windows process account is an operator for the IIS Default Web site. To correct the problem, perform the following procedure.

Procedure   To add the ASP.NET windows process account

1   Launch Internet Information Services (IIS) Manager from Administrative Tools folder.

2   Expand to find the Default Web site.

3   Right-click the Default Web site and select Properties.

4   Click the Operators tab.

5   If LOCALMACHINE\ASPNET is not in the list (where LOCALMACHINE is the name of the IIS Machine, for example KMSWINSQL), click Add.

6   Type LOCALMACHINE\ASPNET, and then click OK. For example (KMSWINSQL\ASPNET) as shown below:
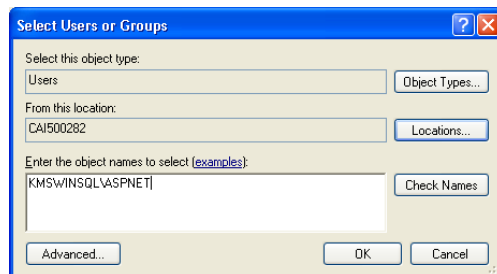
7   Click OK on the Default Web Site Properties window.

8 Run Business Insights Dashboard to determine if the problem is corrected. If the problem persists, refer to the following articles on the Microsoft Web site:

- ▶ http://support.microsoft.com/kb/267904

- ▶ http://support.microsoft.com/kb/555583

If the problem continues to persist, perform the following:

1 Select Windows Start menu > Run and type cmd. Click OK.

2 Change the IIS hard drive: + \inetpub\AdminScripts folder. For example, c:\inetpub\AdminScripts

3 Type cscript adsutil.vbs, and delete w3svc/1/ AdminACL.

HTTP 403 Error Message

Overview   Use the following procedure to verify that the default document is Default.aspx in the IIS Properties window.

Procedure   To verify that the default document is Default.aspx

   1   Launch Internet Information Services (IIS) Manager from the Administrative Tools folder.

   2   Select the name of the installed Business Insights Dashboard, which resides under Web Sites\Default Web Sites.

   3   Select Action menu > Properties.

   4   Click the Documents tab.

   5   Verify the Enable Default Document check box is selected and Default.aspx is the first entry in the list. If it is not in the list, then add it and make sure it is moved to the top of the list.
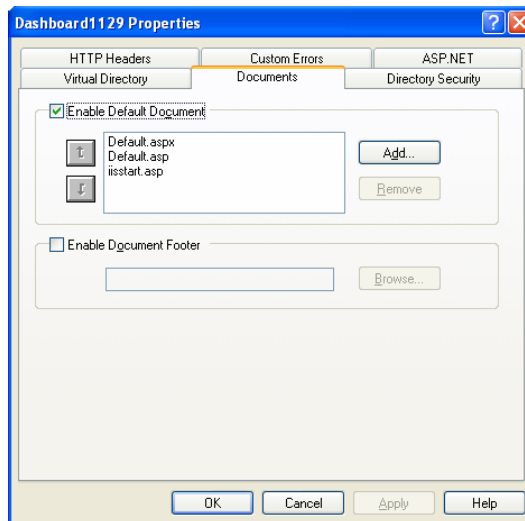
Verifying anonymous access

Verify the Anonymous access check box is cleared in Directory Security. To do this, perform the following procedure.

Procedure

To verify the Anonymous access setting

1 Launch Internet Information Services (IIS) Manager from Administrative Tools folder.
2 Select the name of the installed Business Insights Dashboard, which resides under Web Sites\Default Web Sites.
3 Select Action menu > Properties.
4 Click the Directory Security tab.
5 Click Edit for Authentication methods.
6 Verify that the Anonymous access check box is cleared.

Verifying ASP.NET allows permissions

For a Windows 2003 Server, verify the ASP.NET is allowed permissions.

Procedure

To verify ASP.NET allows permissions

1 Launch Internet Information Services (IIS) Manager from Administrative Tools folder.
2 Expand the appropriate nodes in the IIS to locate the Webservice Extensions node
3 Select Action menu > Properties.
4 Select Webservice Extensions.
5 Find a list of prohibited extensions on the right.
6 Select ASP.NET and allow it.

Links Not Working

Introduction This error can occur when using the Web server as the client. For example, when the site administrator is verifying a good installation. To change the default distributed COM properties, use the following procedure.

Procedure To change the default distributed COM properties, follow these steps:

1 On the Web server, select Windows Start menu > All Programs > Administrative Tools > Component Services.

2 In the Tree pane, click Component Services to expand the list. Then click the Computers folder, and then My Computer.

3 Right-click My Computer, and click the Default Properties tab.

4 At the Authentication Level field, select Connect.

5 At the Impersonation Level field, select Identify.

6 Click OK, and Yes at the message dialog box.

Internet Explorer Client Appearance and Script Errors

Introduction   These errors can occur when the form of styles is not applied, images do not appear, or script error problems appear when using Internet Explorer version 8. These errors usually occurs when the Compatibility View is disabled.

Procedure   To enable the Compatibility View on the client, follow these steps:

1   From Internet Explorer, select Tools.

2   Select Compatibility View.

Note: If the Compatibility View option is disabled, the option is already turned on for this site. This means there are other issues with the site.

Tip: The error may also occur if Internet Explorer version 8 with a Web server installation is running Internet Information Services (IIS) 5. Consider upgrading IIS 5 to a higher version.

Alternate setting   If the Compatibility View is disabled, follow these steps to add the site.

1   From Internet Explorer, select Tools.

2   Select Compatibility View Settings.

3   Select the Display intranet sites in Compatibility View check box, and click Close.

Note: If you do not want to select all intranet sites, click Add to add only this site in the compatibility list.

Unable to Export to Excel

Introduction   In certain instances the "The file is corrupt and cannot be opened" error message may appear when exporting the contents of the grid to Excel. This error indicates that Excel 2010 security settings are prohibiting the exported file from being opened.

Procedure   Follow these steps to change Excel 2010 security settings so the exported file can be opened:

1   Open Excel 2010.

2   Click File and select Options.

3   In the Excel Options window, click Trust Center.

4   In the Excel Options Trust Center pane, click Trust Center Settings.

5   In the Trust Center window, clear the following:

   ▶   Enable Protected View for files originating from the internet

   ▶   Enable Protected View for files located in potentially unsafe locations.

6   Click OK to close the Trust Center window.

7 Click OK to close the Excel Options window.
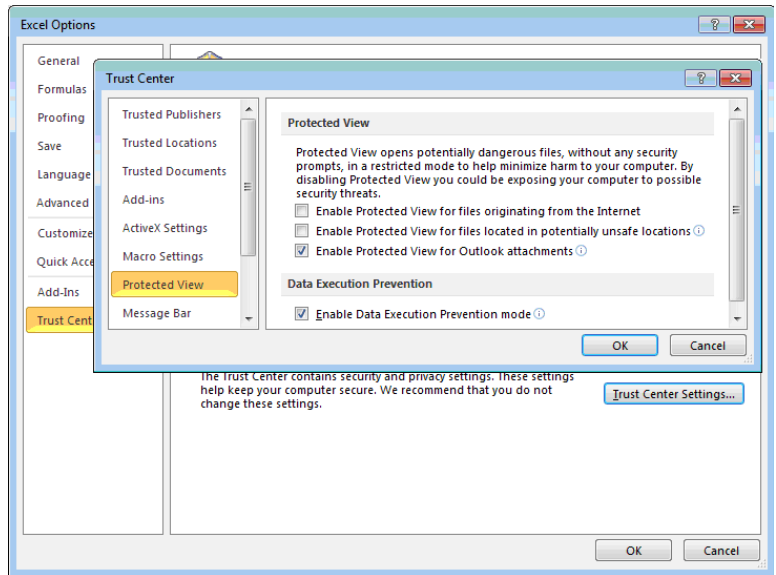
Error Messages

Connection failure

If you receive a connection failure message similar to the following, refer to *Configuring ASP.NET* on page 55.
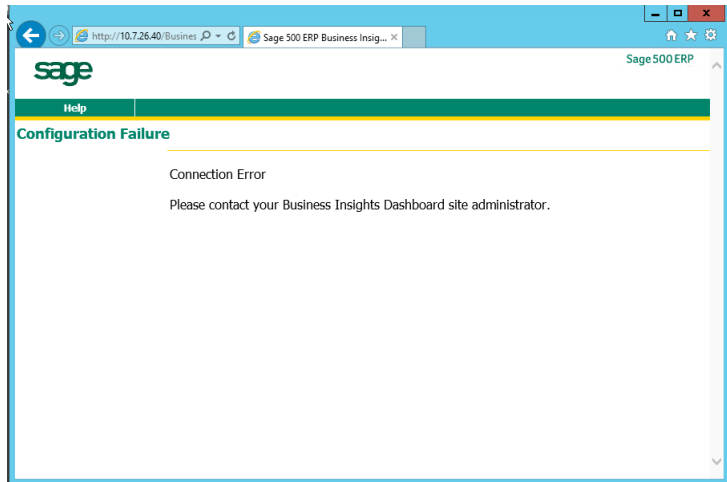
Invalid character error when viewing a Dashboard

The following error message appears when you open the Web page to view the Dashboard: "A name was started with an invalid character. Error processing resource 'http://ws2003ent/BusinessInsightsDashboard/'" Line 1.

To resolve this issue, run aspnet_regiis.exe.

No data appears for the financial Dashboard content

Use Set Up CI Options to update data.

1   Select Common Information Maintenance menu > CI Setup > Set Up CI Options.

2   On the Business Insights Dashboard tab in the Set Up CI Options window, click Update Now.

If the data is still showing all zeros in the Dashboard content, there is a possibility that the user set up to run the SQL Agent job to populate the tables does not have the appropriate security setup to execute the stored procedure the job uses.

Perform the following steps for SQL Server:

1 Launch SQL Server Management Studio.

2 Connect to the server you are using.

3 Under Management, launch the SQL Server Agent.

4 Select Jobs.

5 Find the job for the Dashboard (for example, Sage 500 SAGE500_app Dashboard).

6 Right-click the job and select Properties. Change the owner to a user who has permissions to run the stored procedure.

7 In Sage 500 ERP, run Set Up CI Options to update the data (accessed by selecting the Common Information Maintenance menu > CI Setup menu > Set Up CI Options).

Perform the following steps for SQL 2005:

1 Launch SQL Management Studio.

2 Connect to the server you are using.

3 Open the SQL Server Agent.

4 Find the Dashboard job (for example, Sage 500 SAGE500_app Dashboard).

5 Right-click the job and select Properties. Change the owner to a user who has permissions to run the stored procedure.

6 In Sage 500 ERP, run Set Up CI Options to update the data (accessed by selecting the Common Information Maintenance menu > CI Setup menu > Set Up CI Options).

Web Part Development 7

The Business Insights Dashboard is composed of Web parts that can be customized. This chapter provides information about how to create Web parts, modify their appearance, and serves as a guide to creating additional Web parts.

The Business Insights Dashboard Web parts can be customized using Microsoft Visual Studio 2010.

In this chapter

Introduction to Sage Web Parts

Introduction

Sage Web Parts were built using ASP.NET technologies as Web user controls. They were designed as a replacement for and share the same data structure as the Sage Business Insights Dashboard. This chapter is intended to show how these Web parts were created and to serve as a guide in creating additional Web parts for the Business Insights Dashboard.

During the course of this chapter, an example Web part will be created that shows the daily sales order total for the top five salespersons. It will use a data-bound grid to display the information.

Creating Web parts from user controls

There are advantages and disadvantages to creating Web parts as user controls as opposed to compiled assemblies. The advantages are a user-friendly design surface, control property pages, and available wizards. Creating Web parts using this technology allows easy access to global objects on the Web site such as connection and default company information. You also have the ability to use some of the Sage control components used on this Web site.

The main disadvantage is that the code must be in Visual Basic. The language selection is a site-wide option. Also, Web parts built using user controls are not as easily portable as assemblies.

Compiled Web parts that load user controls

The Web parts created for the Business Insights Dashboard use Web part classes that inherit from the Web Part object and load user controls at runtime. Compiling these classes to a .NET .dll file allows the Web part code to be language independent from the Web site. This also provides a design surface for the user interface (UI) of the Web part as well as allowing for simple deployment.

Setting up the development environment

Sage Web parts were developed in Visual Studio 2008 with no special libraries or templates added. SQL Server 2005 or 2008 are required for the back end with IIS. You must have Sage 500 ERP version 7.30 or higher and the Business Insights Dashboard installed along with the appropriate security setup.

Setting up the Project

Use the following steps to set up the Web part project.

1   Open Microsoft Visual Studio 2010.

2   Select File menu > Open Web Site.

3   At the Web site location field, type the URL of the Business Insights Dashboard. This will add all of the files from the Web site to your project.
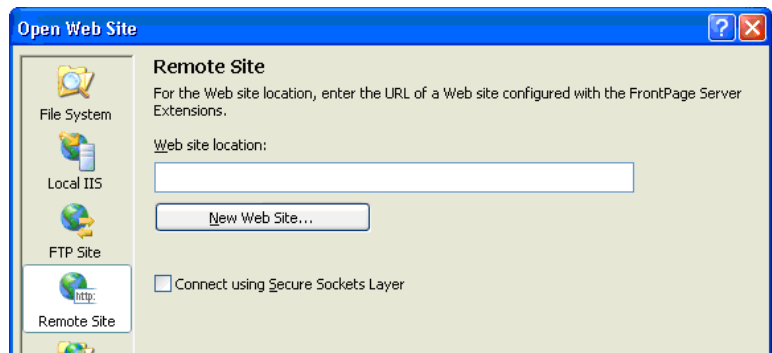
Global objects   There are two global objects in the Business Insights Dashboard that are important to discuss before creating a Web part. First, the connection used to connect to the Sage 500 ERP database is called "SAGE500App." This connection string is stored at the application level in the Web.config file and can be administered either through the IIS Management Console or by editing the Web.config file manually.

1   In the IIS Management Console, right-click the Business Insights Dashboard folder and select Properties to modify the connection string.

2   Click the Edit Configuration button on the ASP.NET tab to view configuration information.

The connection can be accessed in code by using the following reference:

```
Accounting.Web.Framework.Session.OpenConnection
```

The second global object that you need to reference is the default Sage 500 ERP company ID. This value is set on the Dashboard master page and is stored at the application level in the Profile object in the Web.Config file. This value can be accessed in code with the following reference:

```
HttpContext.Current.Profile.Item("CompanyID").
ToString.Trim
```

Creating the Web Part User Control

Use the following steps to create a Web part user control.

1   In Visual Studio, click the Explorer button in the WebPart Custom Gallery folder.

2   Click a folder and select Add New Item.

3   Type a name for the new Web User Control (for example, DailySalesTotal.ascx). Clear the Place code in separate file check box.
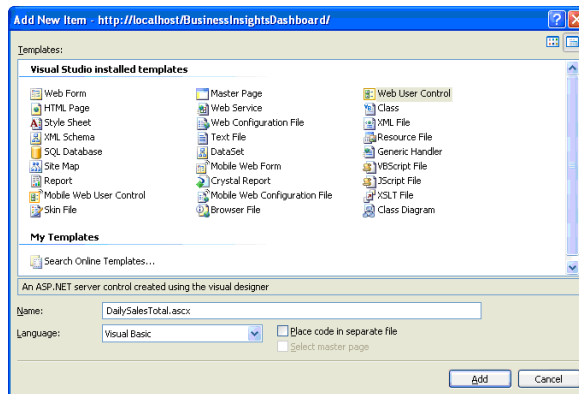
Adding controls to the ASCX file

The control will display in the development window with a blank design surface.

1   Add three panel controls from the standard toolbox to the User Control.

2   Set the ID property of the top control to pnlSelector, the middle to pnlContent, and the bottom to pnlErrorMsg.

3   Drag and drop a GridView control from the data toolbox on to the design surface in the middle panel.

4   Set the ID property of the GridView to gvSalesTotals.

5 Use the Property Manager for the GridView control to
edit the columns and add formatting.
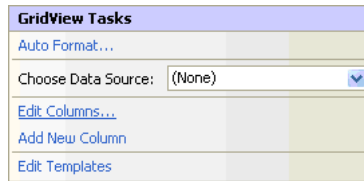
Click Edit Columns to add two BoundFields and configure
them as follows:

First BoundField

Property	Value
HeaderText	Salesperson
DataField	Salesperson
HTMLEncode	False

Second BoundField

Property	Value
HeaderText	Sales Total
DataField	SOTotal
HTMLEncode	False
DataFormatString	{0:$###,###,##0}
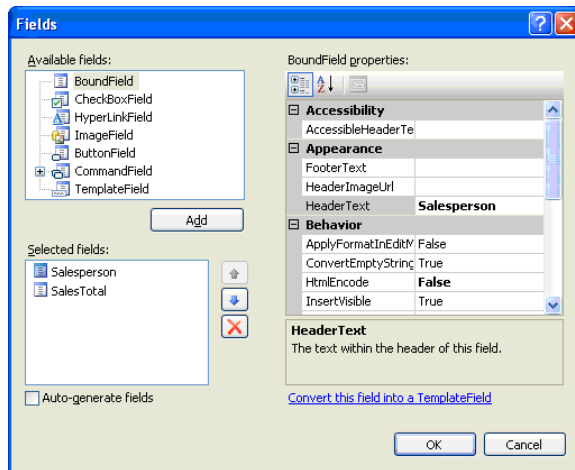
6   Clear the Auto-generate fields check box.

7   Select the .ascx file and click the Source button. Add
`SkinID="WebPart"` inside the `<asp:GridView>` tag.
This adds the style skin to match the Sage Web Parts.

8   For the grid cells to match the Web Parts, add
`<ItemStyle CssClass="DefaultCell" />` between
the `<asp:BoundField>` tag and the
`</asp:BoundField>` end tag.

9   For the first BoundField, set the CssClass cell equal to
DefaultCell1 and all the rest of the cells equal to
DefaultCell. This sets the borders around the cells to
match the existing Web parts.

10   Add a label control from the standard toolbox inside
the pnlErrorMsg control and set the ID property to
lblErrorMessage.

Adding the Company Selector control to the ASCX file

Use the following procedure to add Company Selector controls to the ASCX file.

1 Add a reference to the BIControls.dll to the top of the ascx file as follows:

```
<%@ Register Assembly= "Accounting.
Web.Framework. BIControls"
Namespace="Accounting.Web.Framework"
TagPrefix="cc1" %>
```

2 Add the Company Selector control to pnlSelector.

3 Add the following line between the beginning and ending tag of the Selector panel:

```
<cc1:CompanySelector ID="CompanySelector1"
runat="server" />
```

4 Save the ASCX file.

Note: Because the Company Selector control is late bound using the load control method, just as our Web part, a "can't render" error occurs in the pnlSelector control when in design mode for the DailySalesTotals.ascx file. This error can be ignored as the designer does not support this type of late binding.

Editing the Company Selector control to the ASCX file

In all the panel controls, verify there are no height and width properties specified. In addition to the SkinID for the grid, you can set the width of the grid cells to fit your specific data. Editing the ASCX file and adding an ItemStyle and HeaderStyle tag to the bound columns allow you to set the attributes of the grid control. The following is the code for the Daily Sales Totals grid:

```
<asp:BoundField DataField="Salesperson"
HeaderText="Salesperson" HtmlEncode="False" >
```

```
<HeaderStyle HorizontalAlign="Left" Wrap="false"
VerticalAlign="Bottom"></HeaderStyle >
```

```
<ItemStyle HorizontalAlign="Left" Wrap="False"
Width="80px" CssClass="DefaultCell1" ></ItemStyle>
```

```
</asp:BoundField >
```

```
<asp:BoundField DataField="SOTotal"
HeaderText="SalesTotal" HtmlEncode="False" >
```

```
<HeaderStyle HorizontalAlign="Left" Wrap="false"
VerticalAlign="Bottom"></HeaderStyle >
```

```
<ItemStyle HorizontalAlign="Left" Wrap="False"
Width="80px" CssClass="DefaultCell" ></ItemStyle>
```

```
</asp:BoundField>
```

Set the visible properties of all of the panel controls as follows:

```
pnlSelector Visible="true"
pnlContentVisible="true"
pnlErrorMsg Visible="false"
```

Creating the Web Control Project

Layout

Use the following steps to create the Web control project.

1 Open Visual Studio and create a new project.

2 Select the Class Library template.
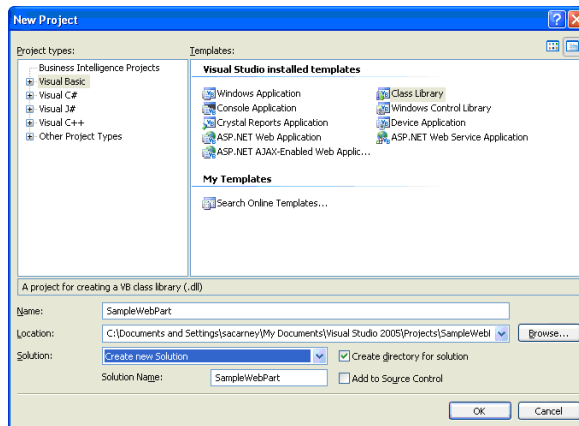
3 In the Solution Explorer, rename the class file Class1.vb to DailySalesTotal.vb.

4 Save the project and close Visual Studio. This project can now be added to the Business Insights Dashboard Web site project.

5 Select File menu > Add > Existing project. Navigate to the Sample Web Part project you just created.

6 Add the following .NET references to the Sample Web Part project:

```
System.Web
System.Data
```

7 Add the following Business Insights Framework references to the Sample Web Part project using the browse control:

```
Accounting.Web.Framework.dll
Accounting.Web.Framework.BIControls.dll
```

These framework dlls can be found in the Bin folder of the Business Insights Dashboard Web site.

8 Select the SampleWebPart.vb from the Solution Explorer. Edit the file and add the following import statements to the top of your class:

```
Imports System
Imports System.Web
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Data
Imports System.Data.SqlClient
Imports System.ComponentModel
Imports Accounting.Web.Framework
Imports System.Text
```

9 The class inherits from the WebPart class, so add the following to the class file:

```
Inherits
System.Web.UI.WebControls.WebParts.WebPart
```

Adding properties to the class

Use the following steps to add properties to the Web Part and override certain inherited properties from the Web Part class. Be sure to add a default value for each property.

1 Add two local private variables to hold the Property values:

```
Private _Title As String = "Daily Sales
Totals"
Private _CompanyID As String = ""
```

2 Add the CompanyID default constant:

```
Private Const DEFAULT_TEXT As String =
"[Default]"
```

3 Override the Title property:

```
<Personalizable(), DefaultValue("Daily
Sales Totals")> _
Public Overrides Property Title() As String
Get
Return _Title
End Get
Set(ByVal value As String)
_Title = value
End Set
End Property
```

4 Add The CompanyID property:

```
<Personalizable(),
DefaultValue(DEFAULT_TEXT)> _
Public Property CompanyID() As String
Get
Return _CompanyID
End Get
Set(ByVal value As String)
_CompanyID = value
End Set
End Property
```

Building the ASCX controls to DailySalesTotal class

A reference must be made to all the controls on the ASCX page in the class. The reference is performed in the Init event that is inherited from the WebPart class. You must declare the private variables in the declarations section of the class for all the controls on the ASCX user control as well as the User Control itself.

```
Private DailySalesUserControl As UserControl

Private WithEvents CompanySelector1 As
CompanySelector

Private pnlSelector As Panel

Private pnlContent As Panel

Private pnlErrorMsg As Panel

Private gvSalesTotals As GridView

Private lblErrorMessage As Label
```

To bind the controls, add the following code to the Init Event in the DailySalesTotals class. Binding the User Control uses the LoadControl command:

```
DailySalesUserControl =
DirectCast(Page.LoadControl("~/WebPart/
CustomGallery/DailySalesTotals.ascx"), UserControl)

DailySalesUserControl.ID = "DailySales"

        Me.Controls.Add(DailySalesUserControl
```

After the control is loaded, the variables can be set for the controls using the FindControl Method of each parent control. You can set the variable for the panel controls as follows:

```
pnlSelector =
DirectCast(DailySalesUserControl.FindControl
("pnlSelector"), Panel)

pnlContent =
DirectCast(DailySalesUserControl.FindControl
("pnlContent"), Panel)
```

```
pnlErrorMsg =
DirectCast(DailySalesUserControl.FindControl
("pnlErrorMsg"), Panel)
```

The child controls can now be set using the FindControl method of the panel controls as follows:

```
CompanySelector1 =
DirectCast(pnlSelector.FindControl
("CompanySelector1"), CompanySelector)

gvSalesTotals =
DirectCast(pnlContent.FindControl("gvSalesTotals"),
GridView)

        lblErrorMessage =
DirectCast(pnlErrorMsg.FindControl
("lblErrorMessage"), Label)
```

These calls should be surrounded by a Try/Catch block and some error handling should be added. Refer to the following Catch portion of the call. Because the controls are not rendered at this point, you must add a label control to display the error message.

```
        Catch ex As Exception

            Dim errLabel As Label = New Label

            errLabel.ID = "DailySalesTotalError"

            errLabel.Text = "DailySalesTotal_Init:"
& ex.Message
Me.Controls.Add(errLabel)

End Try
```

Populating the grid control and rendering the Web part

The bulk of the work in the Web part is performed in the PreRender event that is inherited from the WebPart class.

Because you are using the Sage 500 ERP common routine for error handling, you must first set the control's visible properties before rendering, in case the last rendering of the Web part returns no data or has an error. You must create a method to reset the controls to the correct state and call it at the beginning of the PreRender event. Create the method:

```
Private Sub ResetControls()

        lblErrorMessage.Text = ""

        pnlErrorMsg.Visible = False

        pnlContent.Visible = True

        pnlSelector.Visible = True

End Sub
```

Next, make the call at the beginning of the PreRender event:

```
    ResetControls()
```

This is also where you must link any Help files. In the following example, the DailySalesHelp.htm is created to demonstrate the linking method, and added to the Help folder in the Business Insights Dashboard Web site. The following is the code to link the file:

```
        Me.HelpMode = WebPartHelpMode.Modeless

        Me.HelpUrl = "~/Help/DailySalesHelp.htm"
```

Using the Company Selector control

The Sage 500 ERP Web parts also have a control to select a company other than the default company set in the portal master page. There are several steps that must be taken to implement this control. You have already added the control to your Web part .ascx file and a personalized property to store the CompanyID value. This saves the value for the user between sessions. You can also encapsulate the code in a Try/Catch block and use the Common Error routine to display any error messages that may occur.

You must first instantiate the Web part common class to use the Show Error method. Add the code to the declarations section of your class to instantiate the WebPartCommon class:

```
Private wpc As New WebPartCommon
```

Add the following code to the PreRender event:

```
Dim userProfileCompanyID As String = ""

Try

' get the default company from the Profile object

userProfileCompanyID =
HttpContext.Current.Profile.Item("CompanyID").ToStr
ing. Trim

Catch ex As Exception

lblErrorMessage.Text = wpc.ShowError(ex.Message,
DailySalesUserControl)

End Try


' variable to hold CompanyID that is used

        ' _CompanyID could be [Default]

        Dim UseCompanyID As String

     ' if there is no value for _CompanyID use the
default company from Profile

        If _CompanyID.Trim.Length = 0 Or _CompanyID
= DEFAULT_TEXT Then

          If userProfileCompanyID.Trim.Length = 0
Then 'if no default set then show error message

             lblErrorMessage.Text =
wpc.ShowError(wpc.NoDfltCmpnyMsg,
DailySalesUserControl)

             Exit Sub

          End If
```

```
                ' set the CompanyID property to save

                _CompanyID = DEFAULT_TEXT

                ' use the default company

                UseCompanyID = userProfileCompanyID.Trim

        Else ' if there is a saved value, use it

                UseCompanyID = _CompanyID

        End If

Try ' set the company selector to match the
companyID property

                CompanySelector1.CompanyID = _CompanyID

        Catch ex As Exception

                lblErrorMessage.Text =
wpc.ShowError(ex.Message, DailySalesUserControl)

                Exit Sub

        End Try
```

The wpc.ShowError method clears all the controls on the Web part and displays a message that it is passed in a label.

Lastly, the ValueChanged event must be handled by the Web part to change the CompanyID value when the selector's value changes. Add the following code to handle the event from the selector:

```
Protected Sub CompanySelector1_ValueChanged()
Handles CompanySelector1.ValueChanged

_CompanyID = CompanySelector1.CompanyID

End Sub
```

The Company Selector is now ready to be used.

Creating the SQL command object and binding the grid

A parameterized SQL statement is used for data retrieval with the CompanyID as the parameter. Normally, you can use a parameter for the date of the sales orders. For demonstration purposes, you can hard-code a date compatible with data in the SOA demo company. Use the following code for setting up the SQL Command object:

```
Dim SqlSelect As String

Dim _dtSales As New System.Data.DataTable

' set up the profit grid

Try

Dim str As New StringBuilder(277)

str.AppendLine("SELECT S.SperID[Salesperson],
SUM(SO.TranAmt)[SOTotal] FROM tsoSalesOrder SO")

str.AppendLine("INNER JOIN tarSalesperson S")

str.AppendLine("ON SO.PrimarySperKey = S.SperKey")

str.AppendLine("GROUP BY S.SperID, SO.TranDate,
SO.CompanyID")

str.AppendLine("HAVING SO.TranDate = '2008-06-05'")

str.AppendLine("AND SO.CompanyID = @CompanyID")

str.AppendLine("ORDER BY SUM(SO.TranAmt) DESC")


SqlSelect = str.ToString


Dim sqlCmd As SqlCommand = New SqlCommand


With sqlCmd
.CommandType = CommandType.Text
.CommandText = SqlSelect
.CommandTimeout =
Accounting.Web.Framework.Session.QueryTimeout
```

```
.Parameters.AddWithValue("@CompanyID",
UseCompanyID)

Dim con As SqlConnection =
Accounting.Web.Framework.Session.OpenConnection

.Connection = con

End With


Dim _daSales As SqlDataAdapter = New
SqlDataAdapter(sqlCmd)

_daSales.Fill(_dtSales)


Catch ex As Exception

lblErrorMessage.Text = wpc.ShowError(ex.Message,
DailySalesUserControl)

Exit Sub

End Try

' if no rows exist then clear and show no data
message

If _dtSales.Rows.Count = 0 Then

lblErrorMessage.Text = wpc.ShowError(wpc.NoDataMsg,
DailySalesUserControl)

Exit Sub

End If

gvSalesTotals.DataSource = _dtSales
gvSalesTotals.DataBind()
```

Upgrading Web Parts Created in Sage 500 ERP Version 7.05

This section provides information about deploying Web parts, such as compiling Web parts, using selectors other than the Company Selector, and upgrading user-created Web parts from the Sage 500 ERP version 7.05 Business Insights Dashboard.

Compiling the Web part

The Daily Sales Totals Web part is now be ready to be compiled.

1   Select Build SampleWebPart from the Build menu.

2   After the .dll is built, copy it to the Bin folder of the Web site.

3   The ASCX file should be in the WebParts CustomGallery folder and the Help file should be placed in the Help folder of the Web site. By placing the .dll in the bin folder, your Web part should now appear in the Custom Content 2 section when Add Content is selected.

Using other selectors

The other selectors available for the Sage 500 ERP Web parts are for warehouse, budget type, and purchase product line. The code for these selectors is very similar to the Company Selector and is implemented in much the same manner. If you are using more than one selector, verify that the company selector is used first, because other selectors are usually dependent on the CompanyID being set. Refer to the following code for implementing the Warehouse Selector in the Top Items Web part. In the PreRender event of the Web part, this code follows the company selector code.

```
Try

    WhseSelector1.Initialize(UseCompanyID)

Catch ex As Exception
```

```
            lblErrorMessage.Text =
wpc.ShowError(ex.Message, DailySalesUserControl)

        Exit Sub

    End Try

    If _Warehouse.Trim.Length = 0 Then

        _Warehouse = WhseSelector1.WhseID.Trim

    Else

        WhseSelector1.WhseID = _Warehouse.Trim

    End If

    Dim WhseKey As Int32

    Try

       WhseKey = wpc.GetWhseKey(_Warehouse.Trim,
UseCompanyID)

    Catch ex As Exception

        lblErrorMessage.Text =
wpc.ShowError(ex.Message, DailySalesUserControl)

        Exit Sub

    End Try
```

The following property is used to store the warehouse value at runtime:

```
Private _Warehouse As String = ""

 <Personalizable()> _

 Public Property Warehouse() As String

    Get

        Return _Warehouse

    End Get

    Set(ByVal value As String)

        _Warehouse = value

    End Set
```

```
    End Property
```

The following event handler is used for the warehouse selector control:

```
Protected Sub WhseSelector1_ValueChanged() Handles
WhseSelector1.ValueChanged

        _Warehouse = WhseSelector1.WhseID.Trim

    End Sub
```

The event handler for the company selector control must be changed to the following to set the warehouse to "All" when the company is changed:

```
    Protected Sub CompanySelector1_ValueChanged()
Handles CompanySelector1.ValueChanged

        _CompanyID = CompanySelector1.CompanyID

        _Warehouse = "All"

        WhseSelector1.WhseID = "All"

    End Sub
```

This is the general methodology used for filtering data sets in DataViews and passing user values into GetData and Fill methods in data adapters. This same methodology can be used to create any necessary filters or user selected parameters.

Upgrading user-created Web parts from Business Insights Dashboard version 7.05

The custom user control Web parts created for Sage 500 ERP version 7.05 Business Insights Dashboard are still supported in this version with only minor changes to the ASCX files.

The path in the Web site is changed from BusinessInsightsPortal to BusinessInsightsDashboard. All of the files you have created in the version 7.05 Business Insights Dashboard must be moved to the same folder in the current Dashboard. Be sure to include the CustomGallery1.ascx file that contains the list of your Web parts.

Because the common controls for the Sage 500 ERP Web parts are now contained in a .dll, the reference to the controls must be changed in the user controls .ascx file. Remove any references to Sage 500 ERP common controls, for example:

```
<%@ Register Src="../CompanySelector.ascx"
TagName="CompanySelector" TagPrefix="uc1" %>
```

Add a reference to the Common Controls dll:

```
<%@ Register
Assembly="Accounting.Web.Framework.BIControls"
Namespace="Accounting.Web.Framework"
TagPrefix="uc1" %>
```

This reference only needs to be added once for all the selector controls. Your selector controls now should use the same prefix that is specified in the register statement. If your Web parts use the common error messaging routine in WebPartCommon, make sure all controls are contained inside Panel controls as described in the sample Web part instructions above. The Web parts should now be compatible with the Business Insights Dashboard.

Other Techniques and Methods

The sample Web part created for this document serves as a guide to creating compatible Web parts for the Business Insights Dashboard. You can find examples of topics not covered in this document, including various filtering methods, database access, and working with data sets and data views, throughout the Sage 500 ERP Web parts.

You can find additional information about the development of ASP.NET compatible Web parts at the following Web sites by typing the keywords "Web parts":

http://www.asp.net

http://www.theserverside.net

http://www.codeproject.com

The MSDN Library also contains useful information. At the following Web site, select the .NET Development menu > ASP.NET > Building ASP.NET Applications > Web Parts.

http://msdn2.microsoft.com/en-us/library/

Source Code Listings for Sample Web Part

Source code for
DailySalesTotal.vb

The following is the source code for the DailySalesTotal.vb class.

```
Imports System

Imports System.Web

Imports System.Web.UI

Imports System.Web.UI.WebControls

Imports System.Web.UI.WebControls.WebParts

Imports System.Data

Imports System.Data.SqlClient

Imports System.ComponentModel

Imports System.Text

Imports Accounting.Web.Framework

Public Class DailySalesTotal

    Inherits WebPart

Private _Title As String = "Daily Sales Totals"

    Private _CompanyID As String = ""

Private Const DEFAULT_TEXT As String = "[Default]"

Private DailySalesUserControl As UserControl

    Private WithEvents CompanySelector1 As
CompanySelector

    Private pnlSelector As Panel

    Private pnlContent As Panel

    Private pnlErrorMsg As Panel

    Private gvSalesTotals As GridView

    Private lblErrorMessage As Label

    Private wpc As New WebPartCommon
```

```vbnet
    <Personalizable(), DefaultValue("Daily Sales
Totals")> Public Overrides Property Title() As
String

        Get

            Return _Title

        End Get

        Set(ByVal value As String)

            _Title = value

        End Set

    End Property

    <Personalizable(), DefaultValue(DEFAULT_TEXT)> _

    Public Property CompanyID() As String

        Get

            Return _CompanyID

        End Get

        Set(ByVal value As String)

            _CompanyID = value

        End Set

    End Property

    Private Sub DailySalesTotal_Init(ByVal sender As
Object, ByVal e As System.EventArgs) Handles Me.Init

        Try

            ' call the load control method to load
the Daily Sales User Control

            DailySalesUserControl =
DirectCast(Page.LoadControl("~/WebPart/
CustomGallery/DailySalesTotal.ascx"), UserControl)

            DailySalesUserControl.ID = "DailySales"

            Me.Controls.Add(DailySalesUserControl)
```

```vbnet
' set the control variables for the controls
contained in the user control
            pnlSelector =
DirectCast(DailySalesUserControl.FindControl
("pnlSelector"), Panel)

            pnlContent =
DirectCast(DailySalesUserControl.FindControl
("pnlContent"), Panel)

            pnlErrorMsg =
DirectCast(DailySalesUserControl.FindControl
("pnlErrorMsg"), Panel)

            CompanySelector1 =
DirectCast(pnlSelector.FindControl
("CompanySelector1"), CompanySelector)

            gvSalesTotals =
DirectCast(pnlContent.FindControl("gvSalesTotals"),
GridView)

            lblErrorMessage =
DirectCast(pnlErrorMsg.FindControl
("lblErrorMessage"), Label)


        Catch ex As Exception

            ' display the error message if necessary

            Dim errLabel As Label = New Label

            errLabel.ID = "DailySalesTotalError"

            errLabel.Text = "DailySalesTotal_Init:"
& ex.Message

            Me.Controls.Add(errLabel)

End Try

    End Sub

Private Sub DailySalesTotal_PreRender(ByVal sender
As Object, ByVal e As System.EventArgs) Handles
Me.PreRender
```

```
' call to reset the visible state of all controls
        ResetControls()
        'set the help path
        Me.HelpMode = WebPartHelpMode.Modeless
        Me.HelpUrl = "~/Help/DailySalesHelp.htm"
        Dim userProfileCompanyID As String = ""
        Try
          ' get the default company from the Profile
object
            userProfileCompanyID =
HttpContext.Current.Profile.Item("CompanyID").
ToString.Trim
        Catch ex As Exception
            lblErrorMessage.Text =
wpc.ShowError(ex.Message, DailySalesUserControl)
        End Try
        ' variable to hold CompanyID that is used
        ' _CompanyID could be [Default]
        Dim UseCompanyID As String
      ' if there is no value for _CompanyID use the
default company from Profile
        If _CompanyID.Trim.Length = 0 Or _CompanyID
= DEFAULT_TEXT Then
            If userProfileCompanyID.Trim.Length = 0
Then 'if no default set then show error message
lblErrorMessage.Text =
wpc.ShowError(wpc.NoDfltCmpnyMsg,
DailySalesUserControl)
                Exit Sub
            End If
            ' set the CompanyID property to save
```

```
            _CompanyID = DEFAULT_TEXT
            ' use the default company
            UseCompanyID = userProfileCompanyID.Trim
        Else ' if there is a saved value, use it
            UseCompanyID = _CompanyID
        End If
        Try ' set the company selector to match the
companyID property
            CompanySelector1.CompanyID = _CompanyID
        Catch ex As Exception
            lblErrorMessage.Text =
wpc.ShowError(ex.Message, DailySalesUserControl)
            Exit Sub
        End Try
        ' set up the sales totals grid
        Dim SqlSelect As String
        Dim _dtSales As New System.Data.DataTable
        ' Sql statement for the grid data adapter
        Try
            Dim str As New StringBuilder(277)
            str.AppendLine("SELECT
S.SperID[Salesperson], SUM(SO.TranAmt)[SOTotal]
FROM tsoSalesOrder SO")
str.AppendLine("INNER JOIN tarSalesperson S")
            str.AppendLine("ON SO.PrimarySperKey =
S.SperKey")
            str.AppendLine("GROUP BY S.SperID,
SO.TranDate, SO.CompanyID")
            str.AppendLine("HAVING SO.TranDate =
'2008-06-05'")
```

```vb
              str.AppendLine("AND SO.CompanyID =
@CompanyID")

           str.AppendLine("ORDER BY SUM(SO.TranAmt)
DESC")

           SqlSelect = str.ToString

          Dim sqlCmd As SqlCommand = New SqlCommand

           ' setup the sql Command object

           With sqlCmd

               .CommandType = CommandType.Text

               .CommandText = SqlSelect

               .CommandTimeout =
Accounting.Web.Framework.Session.QueryTimeout


.Parameters.AddWithValue("@CompanyID",
UseCompanyID)

               Dim con As SqlConnection =
Accounting.Web.Framework.Session.OpenConnection

               .Connection = con

           End With

           Dim _daSales As SqlDataAdapter = New
SqlDataAdapter(sqlCmd)

           _daSales.Fill(_dtSales)

       Catch ex As Exception

           lblErrorMessage.Text =
wpc.ShowError(ex.Message, DailySalesUserControl)

           Exit Sub

       End Try

       ' if no rows exist then clear and show no
data message

       If _dtSales.Rows.Count = 0 Then
```

```
            lblErrorMessage.Text =
wpc.ShowError(wpc.NoDataMsg, DailySalesUserControl)

            lblErrorMessage.Text =
lblErrorMessage.Text & " For " & UseCompanyID

            Exit Sub

        End If

        'bind the grid control

        gvSalesTotals.DataSource = _dtSales

        gvSalesTotals.DataBind()

End Sub

    Private Sub ResetControls()

        lblErrorMessage.Text = ""

        pnlErrorMsg.Visible = False

        pnlContent.Visible = True

        pnlSelector.Visible = True

    End Sub

    Private Sub CompanySelector1_ValueChanged(ByVal
sender As Object, ByVal e As System.EventArgs)
Handles CompanySelector1.ValueChanged

        _CompanyID = CompanySelector1.CompanyID

    End Sub

End Class
```

Source code for DailySalesTotal.aspx

The following is the source code for the DailySalesTotal.aspx class.

```
<%@ Control Language="VB" AutoEventWireup="false"
EnableViewState="true" %>

<%@ Register
Assembly="Accounting.Web.Framework.BIControls"
Namespace="Accounting.Web.Framework"
TagPrefix="cc1" %>
```

```
<asp:Panel ID="pnlSelector" runat="server"
Visible="true">

    <cc1:CompanySelector ID="CompanySelector1"
runat="server" />

</asp:Panel>

<asp:Panel ID="pnlContent" runat="server"
Visible="true">

    <asp:GridView ID="gvSalesTotals" runat="server"
SkinID="WebPart" AutoGenerateColumns="False">

        <Columns>

            <asp:BoundField DataField="Salesperson"
HeaderText="Salesperson" HtmlEncode="False" >

                <HeaderStyle HorizontalAlign="Left"
Wrap="false" VerticalAlign="Bottom"></HeaderStyle >

                <ItemStyle HorizontalAlign="Left"
Wrap="False" Width="80px" CssClass="DefaultCell1" >
</ItemStyle>

            </asp:BoundField >

            <asp:BoundField DataField="SOTotal"
HeaderText="SalesTotal" HtmlEncode="False" >

                <HeaderStyle HorizontalAlign="Left"
Wrap="false" VerticalAlign="Bottom"></HeaderStyle >

                <ItemStyle HorizontalAlign="Left"
Wrap="False" Width="80px" CssClass="DefaultCell"
></ItemStyle>
</asp:BoundField>

        </Columns>
</asp:GridView>
</asp:Panel>

<asp:Panel ID="pnlErrorMsg" runat="server"
Visible="false"> <asp:Label ID="lblErrorMessage"
runat="server" CssClass="ErrorLabel" Visible="true"
></asp:Label>

</asp:Panel>
```

Index

Numerics

128-bit encryption 10

A

Anonymous login
    configuring 34
Authenticated access 35

B

Business Insights Dashboard
    client requirements 14
    configuring 50
    configuring ASP .NET 55
    custom errors 81
    customizing Dashboard Web pages 54
    description of 2
    displaying a specific number of rows
       71
    enabling and disabling custom errors
       76
    handling large sets of data 63
    modifying the execution timeout value
       85
    modifying the Filter On Numerics
       setting 73
    modifying the Immediate Results for
       Filter setting 78

Business Insights Dashboard (continued)
    modifying the Immediate Results
       setting 76
    requirement for 3, 14
    scheduling updates 41
    securing the connection 63
    security 3
    setting connection strings 55
    setting security for Web Parts 53
    setting up 53
    setting up users 51
    updating data 52
    Web browser configuration 14
Business Insights Dashboard Web site
    customizing 103
    Windows authentication 7

C

Cascading Stylesheets (CSS). *See* Stylesheets
Case sensitivity 14
Checklist, installation 26
Client workstation
    Business Insights Dashboard settings
       14
Co-location hosting 6
Common Information, options to set in 40