Microsoft Dynamics®

Microsoft Dynamics® GP

Integration Manager User's Guide

Publication date

March 2013

Contents

INTRODUCTION

Introduction

Integration Manager for Microsoft Dynamics® GP is a data integration software tool that you can use to extract, transform, validate, and transfer data among your business applications, so that you won't need to re-key data. Integration Manager provides a cost-effective way to integrate data without a need for specialized knowledge of databases or programming interfaces.

Check for current instructions

This information was current as of March 26, 2013. The documentation may be updated as new information becomes available. Check the Documentation and resources (http://go.microsoft.com/fwlink/?LinkId=249465) for the most current documentation.

.This introduction is divided into the following sections:

What's in this manual

This manual is designed to give you an in-depth understanding of how to use Integration Manager.

The manual is divided into the following parts:

- Part 1, **Getting started**, provides information on how to install and start Integration Manager, and how to verify the installation. It also describes the different parts of the Integration Manager workspace.
- Part 2, **Building and running integrations**, explains how to create and run integrations. It also contains information about troubleshooting integration problems.
- Part 3, **Managing integrations**, explains how to import and export integrations from other databases, how to use the Object Browser, and how to manage logs, including viewing and purging logs.
- Part 4, **Adapter reference**, describes the adapters that can be used with Integration Manager and contains a destination reference for each adapter.
- Part 5, **Using VBScript**, describes how to attach scripts to integrations.

Some features described in this documentation are optional and can be purchased through your Microsoft Dynamics GP partner.

To view information about the release of Integration Manager that you're using and which adapters are installed, choose **Help** > **About Integration Manager**.

Prerequisites

This manual assumes that you are familiar with Microsoft Dynamics financial applications. Experience working with data in tabular format is also helpful.

If you intend to use the advanced capabilities of Integration Manager, then you should know how to set up an ODBC data source and issue queries using SQL statements.

Symbols and conventions

For definitions of unfamiliar terms, see the glossary in the manual or refer to the glossary in Help.

Symbol	Description
	The light bulb symbol indicates helpful tips, shortcuts and suggestions.
	The warning symbol indicates situations you should be especially aware of when completing tasks.

This manual uses the following conventions to refer to sections, navigation and other information.

Convention	Description
Creating a batch	Italicized type indicates the name of a section or procedure.
File >> Print or **File > Print**	The (>>) or (**>**) symbol indicates a sequence of actions, such as selecting items from a menu or toolbar, or pressing buttons in a window. This example directs you to go to the File menu and choose Print.
TAB or ENTER	All capital letters indicate a key or a key sequence.

Resources available from the Help menu

The Microsoft Dynamics GP Help menu gives you access to user assistance resources on your computer, as well as on the Web.

Contents

Opens the Help file for the active Microsoft Dynamics GP component, and displays the main "contents" topic. To browse a more detailed table of contents, click the Contents tab above the Help navigation pane. Items in the contents topic and tab are arranged by module. If the contents for the active component includes an "Additional Help files" topic, click the links to view separate Help files that describe additional components.

To find information in Help by using the index or full-text search, click the appropriate tab above the navigation pane, and type the keyword to find.

To save the link to a topic in the Help, select a topic and then select the Favorites tab. Click Add.

Index

Opens the Help file for the active Microsoft Dynamics GP component, with the Index tab active. To find information about a window that's not currently displayed, type the name of the window, and click Display.

About this window

Displays overview information about the current window. To view related topics and descriptions of the fields, buttons, and menus for the window, choose the appropriate link in the topic. You also can press F1 to display Help about the current window.

Lookup

Opens a lookup window, if a window that you are viewing has a lookup window. For example, if the Checkbook Maintenance window is open, you can choose this item to open the Checkbooks lookup window.

Show Required Fields

Highlights fields that are required to have entries. Required fields must contain information before you can save the record and close the window. You can change the font color and style used to highlight required fields. In the navigation pane for the Microsoft Dynamics GP home page, choose User Preferences and then choose Display.

Printable Manuals

Displays a list of manuals in Adobe Acrobat .pdf format, which you can print or view.

What's New

Provides information about enhancements that were added to Microsoft Dynamics GP since the last major release.

Documentation and resources on the web

Opens a Web page that provides links to a variety of Web-based user assistance resources. Access to some items requires registration for a paid support plan.

Customer Feedback Options

Provides information about how you can join the Customer Experience Improvement Program to improve the quality, reliability, and performance of Microsoft® software and services.

Send us your documentation comments

We welcome comments regarding the usefulness of the Microsoft Dynamics GP documentation. If you have specific suggestions or find any errors in this manual, send your comments by e-mail to the following address: bizdoc@microsoft.com.

To send comments about specific topics from within Help, click the Documentation Feedback link, which is located at the bottom of each Help topic.

Note: By offering any suggestions to Microsoft, you give Microsoft full permission to use them freely.

Part 1: Getting started

Use this part of the documentation to learn how to install and start Integration Manager. If you are new to Integration Manager, you might want to follow the steps outlined in the Integration Manager Quick Start guide to familiarize yourself with Integration Manager.

This part of the documentation includes the following information.

- Chapter 1, "Integration Manager overview," contains overview information about Integration Manager and includes common terminology used in Integration Manager.
- Chapter 2, "Installing Integration Manager," describes how to set up Integration Manager, and includes security information.
- Chapter 3, "Using Integration Manager," describes the Integration Manager workspace and includes information about starting Integration Manager.

Chapter 1: Integration Manager overview

This part of the documentation includes some of the common terminology used in Integration Manager. Before you begin using Integration Manager, you should familiarize yourself with the terminology used throughout the product to have a better understanding of the integration process.

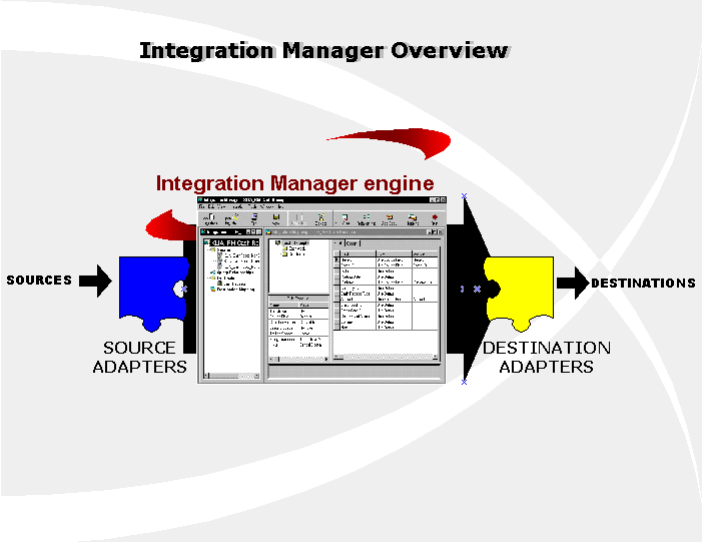
This part of the documentation includes the following information.

Integration Manager

Integration Manager provides you with a safe and easy way of integrating data between business applications. Integrating data involves extracting data from source applications or databases and bringing the data into a destination. You can integrate data from external business databases, e-commerce solutions, or other data file types into the Microsoft Dynamics GP application.

When you create an integration, you'll specify sources, destinations, and destination mappings. Sources include the data to integrate into a destination, such as text files or ODBC databases. A destination indicates where to integrate your source data within Microsoft Dynamics GP, such as General Journal or Payables Transaction. The destination mapping indicates where each item from your source data goes in the destination, such as which fields the source values will be integrated into.

The following figure shows an overview of how source adapters, destination adapters, and the Integration Manager engine work to move data between applications.



Source

A source indicates where the information to be integrated comes from. In Integration Manager, a source can be anything from a comma- or tab-delimited file, or a database such as an Open Database Connectivity source (ODBC). Sources exist independently of the source adapters. Refer to Chapter 5, “Adding sources,” for more information.

Source adapters

Source adapters connect to sources, and filter and extract data. The data is passed on to the Integration Manager engine for processing.

Integration Manager engine

Working with the source adapter and destination adapter, the Integration Manager engine helps you map and integrate the source data into the destination.

Destination

A destination indicates where to integrate the processed information. A destination can be an application or database. Destinations exist independently of the destination adapters. Microsoft Dynamics GP is an example of a destination.

Destination adapters

Destination adapters validate data before integrating it to the destination application, database, or file. If you do not have a destination adapter installed, you won't be able to select a destination.

Destination mappings

Destination mappings define how source data is mapped to the destination. Typically, the information comes from the source you specified, but it also can come from a constant value or default value in the destination. The integration mapping includes several rules you can use when creating a destination mapping.

Query

A query is a request for information. In Integration Manager, queries are used to refer specifically to requests for information from a text file or ODBC source. You can create several queries when using ODBC/text as your source.

Query relationship

When you specify more than one ODBC/text source, you create several queries, as well. You need to create a query relationship between these queries. This relationship tells Integration Manager how the queries work together during the integration. For more information about creating query relationships, refer to Chapter 6, "Creating query relationships."

Chapter 2: Installing Integration Manager

This information will guide you through the steps you need to complete to install Integration Manager, and includes the information you need to know before you begin using Integration Manager.

This part of the documentation includes the following information.

System requirements

System requirements, including which operating systems are supported, are available at the following site. The requirements for Integration Manager will be the same as the requirements listed for Microsoft Dynamics GP, unless otherwise noted.

http://go.microsoft.com/fwlink/?LinkId=249465

You should verify that all system requirements are met before installing Integration Manager.

Integration Manager installation requirements

Administrative rights

You must log in to your computer or network as a user with full, local administrative rights before installing and running Integration Manager.

Microsoft Dynamics GP

If you are using the Microsoft Dynamics GP destination adapter, you must install Microsoft Dynamics GP before you install Integration Manager. If you select to install the Microsoft Dynamics GP destination adapter when Microsoft Dynamics GP isn't installed, a message appears.

eConnect

You must install eConnect before you can install the Microsoft Dynamics GP eConnect destination adapter.

Microsoft Dynamics GP in other countries

If you use Microsoft Dynamics GP in any country other than the United States, contact the sales/support office in your country for information about configurations that have been tested with Integration Manager. In some countries, Integration Manager must be used on an "admin" installation of the client software that uses the U.S. dictionary and runs on a U.S. version of the operating system.

Windows Internet Explorer®

Integration Manager includes a Script Library for Integration Manager, a collection of commonly used scripts you might find useful in your integrations.

Microsoft Office 2013

Integration Manager can create a log file that contains detailed information about the results of an integration. The Integration Log shows a list of logs; one log for each time you run the integration. To view the logs if you have installed Office 2013, the following components must be installed.

- 2007 Office System Driver: Data Connectivity Components
- Microsoft Report Viewer 2008 Redistributable Package

You can download these components from http://www.microsoft.com/en-us/download.

Microsoft Data Access Components

The latest version of Microsoft Data Access Components (MDAC) must be installed on your computer before you install Integration Manager. You also should install the latest Windows Service Packs.

Integration Manager user information

Sample data

When Integration Manager installs sample data, it installs a sample Integration Manager database (IntegrationManager.imd) files that store integrations and some sample text. Sample data files and the sample Integration Manager database files are typically installed in the Samples folder.

```
C:\Program Files\Microsoft Dynamics\Integration Manager\Samples
```

Readme

Refer to the Readme file (IMReadme.rtf) for last minute changes.

Uninstalling previous versions of Integration Manager

If you have any previous versions of Integration Manager (IM) installed on your computer, we strongly encourage you to remove them before you install Integration Manager. Be sure to back up your database before removing any previous versions.

Before you remove a previous version of Integration Manager, we recommend that you make a backup copy of your existing Integration Manager database. After you back up your Integration Manager database file, you'll be able to use the database in Integration Manager for Microsoft Dynamics GP 2013.

To uninstall a previous version of Integration Manager:

To remove Integration Manager, use your system's Add/Remove Programs utility.

1. If you're using Windows Vista® or Windows Server 2008, choose **Start > Control Panel > Programs > Programs and Features**.

   If you're using Windows 7 or Windows Server 2008 R2, choose **Start** > **Control Panel** > **Uninstall a program**.

2. Select **Integration Manager**.

3. Click **Remove**.

Installing Integration Manager

Use the following procedure to install Integration Manager. You need to place the Integration Manager database file into a writable folder. We recommend putting your Integration Manager database files in C:/Programs/Microsoft Dynamics/Integration Manager/Samples or C:/Programs/Microsoft Dynamics/Integration Manager/Data.

To install Integration Manager:

1. Be sure that you're logged in to Windows as a user with system administrator privileges.

2. From the Microsoft Dynamics GP media, double-click the Setup.exe file to open the installation window.

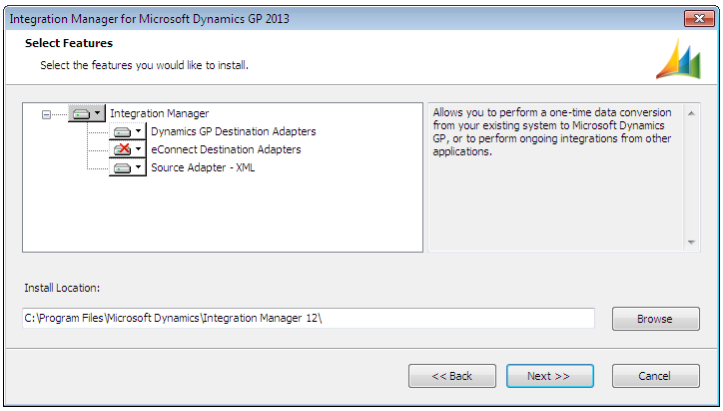
3. Under Additional Products, click **Integration Manager** > **Install**.

   The installation program verifies that your system has the minimum operating system required to run Integration Manager for Microsoft Dynamics GP 2013. If your system does not meet requirements, the installation will not continue.

4. If the current version of Integration Manager is already installed on this computer, the Modify Installation window will open. In the Modify Installation window, you can choose to add or remove features, repair installation files, or remove this version of Integration Manager from your computer.

5. Select to accept the license agreement. Click **Next**.

6. The Select Features window opens. Select the adapters to install.



7. In the Select Features window, indicate where to install the Integration Manager engine and adapter files and click **Next** to continue.

   We recommend that you install these Integration Manager components in the default folder (*Program Files\Microsoft Dynamics\Integration Manager*). The sample Integration Manager database file (IntegrationManager.imd) is installed to this location.

   If you saved an existing Integration Manager database file from a previous version of Integration Manager, you can move it into this folder after you complete the installation.

*To use a shared Integration Manager database file from multiple workstations in a network environment, you must copy the Integration Manager database file to a shared network location after the install is complete. Locate and copy the Integration Manager database file to your shared network location. To access the shared Integration Manager database file, from each workstation that needs to access this file, open Integration Manager. From the **Tools** menu, choose **Options**. Browse to the location of the shared Integration Manager database file, and click **OK**. If you save the file with the .imd file extension, you can double click the .imd file to open your database. For example, double clicking IntegrationManager.imd will open the IntegrationManager.imd database.*

8. The Ready to Install window opens. Click **Install** to start the installation. Click **Back** to review your settings and make corrections as necessary.

   During the copying and installation of files, an Installation Progress window opens. Several messages appear about the progress of the install. When the installation is complete, the Installation Complete window opens.

9. In the Installation Complete window, click **Finish**.

10. The next time someone launches Microsoft Dynamics GP, a message appears. The message may indicate that new code must be included or that the dictionary needs to be un-chunked. Click **Yes**.

11. If you backed up your Integration Manager database file before the installation, move that file into the destination folder you specified during installation.

*You also can change the location of your default Integration Manager database file by choosing **Tools** > **Options** in Integration Manager. The Options window opens, and you can enter the default Integration Manager database path on the **General** tab. If you do not have a default Integration Manager database, the Select Database window opens, where you can select a database.*

Registering Integration Manager

When you purchased Integration Manager, you were provided with valid registration keys. You can enter Integration Manager registration keys in the Integration Manager Registration window. You also can include registration keys in the Microsoft.Dynamics.GP.IntegrationManager.ini file. Refer to *Storing registration keys in the Microsoft.Dynamics.GP.IntegrationManager.ini file* on page 16 for more information. After you register Integration Manager, you can set up and run integrations.

Use the following procedure to register Integration Manager using the Integration Manager Registration window.

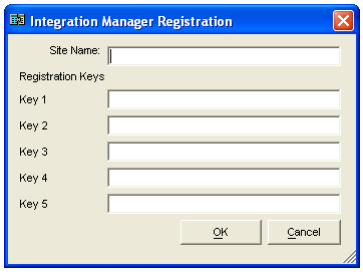
To register Integration Manager:

1. Start Integration Manager, and from the Integration Manager Not Registered window, click **Register Now**.

   – Or –

   From Integration Manager, choose **Tools** > **Registration**.

   The Integration Manager Registration window opens.



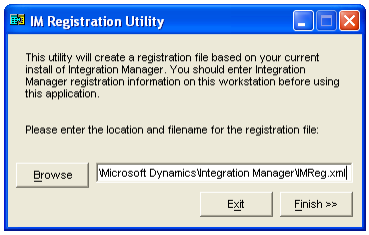
2. In the Integration Manager Registration window, enter the site name exactly as it appears in your registration keys.

3. Enter the unique set of registration keys you were given when you purchased Integration Manager and click **OK**.

   Integration Manager is now registered on your workstation. If you need to register Integration Manager on multiple workstations in a network environment, continue with the following steps to use the IM Registration Key Utility.

4. From the computer where you registered a single installation of Integration Manager, use Windows Explorer to browse to where you installed Integration Manager—usually C:\Program Files\Microsoft Dynamics\Integration Manager—and double-click **IMRegistrationUtility.exe**.

The IM Registration Utility window opens. Note that the default location points to where Integration Manager is installed and the default file name is IMReg.xml.



5. From the IM Registration Utility window, click **Finish** to create a registration entries file. You can accept the default location and file name for the file.

6. On the message that appears, which explains that the registration entries file has been created, click **OK**.

7. Copy the file that you just created to each workstation that has Integration Manager installed and put it in the Integration Manager application directory. Registration settings will be imported the next time you start Integration Manager.

   When you start Integration Manager, Integration Manager verifies if the Registration Keys are valid. If the keys are not valid on any of the workstations, Integration Manager notifies you that required information is missing or not valid, and you'll need to fix it before you can use Integration Manager.

Storing registration keys in the Microsoft.Dynamics.GP.IntegrationManager.ini file

Administrators and advanced users of Integration Manager can store registration keys in the Microsoft.Dynamics.GP.IntegrationManager.ini file, which, if it exists, overrides the Registry's license key information. This allows administrators to set licensing information at the user level, and it is helpful when more than one customer is sharing a Terminal Services server or Microsoft SQL Server® in a data center environment.

The first time a user starts Integration Manager, the software looks for a file named Microsoft.Dynamics.GP.IntegrationManager.ini in the folder where Integration Manager is installed. If it finds an Microsoft.Dynamics.GP.IntegrationManager.ini file, then it uses the settings that are stored there to override some of the default behaviors in Integration Manager.

To use the Microsoft.Dynamics.GP.IntegrationManager.ini in a Terminal Services environment, Terminal Services users need to use separate Microsoft.Dynamics.GP.IntegrationManager.ini files. These separate files store different Integration Manager registration keys that are unique to the user. To do this, Terminal Services users need to modify their Integration Manager shortcut on their desktop with a switch that designates the location of the Microsoft.Dynamics.GP.IntegrationManager.ini file. Append the Integration Manager switch with "/ini=C:\home\TSUser1\Microsoft.Dynamics.GP.IntegrationManager.ini" on the end of the shortcuts target string.

For example:

```
C:\Program Files\Microsoft Dynamics\Integration
Manager\Microsoft.Dynamics.GP.IntegrationManager.exe /
ini=C:\home\TSUser1\Microsoft.Dynamics.GP.IntegrationManager.ini
```

Registry information in the Microsoft.Dynamics.GP.IntegrationManager.ini file would appear as follows:

```
[IMRegistration]
SiteName=Fabrikam, Inc.
Key1=H94KDFJ9009SDF
Key2=J90DF75KDLK0
Key3=LSDKJAF92348UL
Key4=F98
```

Microsoft.Dynamics.GP.IntegrationManager.exe first looks in the Microsoft.Dynamics.GP.IntegrationManager.ini file and checks for the existence of the value SiteName. If SiteName exists, it assumes registration data to be contained in the Microsoft.Dynamics.GP.IntegrationManager.ini file and reads all the keys from the Microsoft.Dynamics.GP.IntegrationManager.ini file. Any missing keys are defaulted to a blank string (as if there were no value for that key). In the above example, Key 5 would be read in as a blank value.

If no registration information exists in the Microsoft.Dynamics.GP.IntegrationManager.ini file, then Microsoft.Dynamics.GP.IntegrationManager.exe checks for licensing information in the Registry.

If you use the Microsoft.Dynamics.GP.IntegrationManager.ini file to store registration keys, the registration dialog box continues to open. To hide this window, under the [IMReg] setting in the Microsoft.Dynamics.GP.IntegrationManager.ini file, add HideMsgBox=True.

If you are using only the Microsoft.Dynamics.GP.IntegrationManager.ini file and no valid registration keys exist in the Registry, the Integration Manager Not Registered window still opens every time you launch Integration Manager. To hide it, select **Do not show this window again**.

Security requirements

There are certain items that only the system administrator has access to. If you are not the administrator of the workstation, you can run Process Monitor to see what items you do not have security access to. Process Monitor is a Sysinternals product. Refer to the Sysinternals Web site (http://technet.microsoft.com/en-us/sysinternals/default.aspx) for additional information.

Database permissions when using eConnect destination adapters

If you are using eConnect destination adapters, the Microsoft Dynamics GP databases are accessed using the eConnect service account. The eConnect service account has access to every Microsoft Dynamics GP database that is created. You may want a more secure environment by providing user access to only select databases. For example, you may have a shared hosting environment with multiple customers or a single customer deployment where Integration Manager users should only integrate with specific companies.

Instead of giving the eConnect service account permissions to the Microsoft Dynamics GP databases, you can give the user accounts running Integration Manager permissions to the Microsoft Dynamics GP databases that the users will be integrating data with. You can then change Integration Manager to access the databases using the account running Integration Manager instead of the eConnect service account.

After updating the database permissions, change the RequireProxyService line to False in the Microsoft.Dynamics.GP.IntegrationManager.ini file. The file is typically located in the Integration Manager installation folder.

Change from	Change to
`RequireProxyService=True`	`RequireProxyService=False`

Database permissions when using the Dynamics GP destination adapters

You'll need to have security access to the set up windows for the module that you are importing into. For example, to import data into the SOP transaction destination, users must have security to the Sales Order Processing Setup window.

You must have security access to all of the windows in the destination that you are importing into. All of the buttons and other windows that can be opened from the destination window need to have security granted to the user that is running the integration.

Data can only be imported into the standard Microsoft Dynamics GP windows. Integration Manager cannot import data into any modified windows or third party products.

User security

If the Integration Manager option is not available in Microsoft Dynamics GP (**Microsoft Dynamics GP** menu > **Tools** > **Integrate** > **Integration Manager**), either you have not registered Integration Manager or you do not have the proper user security settings. For more information about registering Integration Manager, refer to *Registering Integration Manager* on page 15. For more information about user security settings, refer to your System Setup Guide (**Help** > **Contents** > **select Setting Up the System**).

The default security task for Integration Manager is ADMIN_SYSTEM_012*. You can refer to KB 939542 for instructions on creating a security task specific for Integration Manager. If you are only registered for Integration Manager - Conversions, you will not be able to access Integration Manager from the menu in Microsoft Dynamics GP.

Security considerations for integrations

Integration Manager provides powerful integration and import capabilities. Be aware, however, that some functionality that is designed to improve performance may have security implications. Use special care when working with the following features of Integration Manager.

VBScript If you use VBScript to open a connection to a destination, be sure to close it at the end of the script to help prevent unauthorized use of that connection. For more information, see *Integration problems* on page 86.

Command line integrations If you run integrations from the command line and create a login macro, note that the macro will need to store a User ID and Password to launch Microsoft Dynamics GP. Be extremely cautious in who has access to the directory location where the macro is stored. For more information, see *Recording the login macro* on page 107.

Rule properties for override fields If you set up rules that allow a value to be overridden, and if your company's business practices require that a password be entered to override that value, Integration Manager can store the appropriate password. Be extremely cautious in who has access to the integrations that have this capability. For more information, see Chapter 19, "Adapters and Destination Mappings."

Uninstalling the current version of Integration Manager

Use the following procedure to uninstall the current version of Integration Manager.

To uninstall the current version of Integration Manager:

1. Be sure that you're logged in to Windows as a user with system administrator privileges.

2. Close all programs that are running, including those in the system tray.

3. From the Microsoft Dynamics GP media, double–click the Setup.exe file to open the installation window. Click **Integration Manager** > **Install**.

   — Or —

   If you're using Windows Vista or Windows Server 2008, choose **Start** > **Control Panel** > **Programs** > **Programs and Features**.

   If you're using Windows 7 or Windows Server 2008 R2, choose **Start** > **Control Panel** > **Uninstall a program**.

4. In the Program Maintenance window, click **Remove** to uninstall this version of Integration Manager.

5. The Remove Program window opens. Click **Remove** to continue.

   The Remove Progress window opens.

6. Click **Finish** when the uninstall process is complete.

Repairing your Integration Manager installation

Use the following procedure to repair your installation of Integration Manager.

Repairing your installation of Integration Manager overwrites the IntegrationManagerSamples.IMD file so it only contains the original sample integrations. If you are using the IntegrationManagerSamples.IMD file to store your integrations, back up the IntegrationManagerSamples.IMD file and restore the backkup after you repair the installation.

When you repair your installation, the IntegrationManagerSamples.IMD file will be overwritten to only contain the original sample integrations. If this was the file that you are using to store your integrations, you must backup this file and restore it after you run Repair.

To repair your Integration Manager installation:

1. Close all programs that are running, including those in the system tray.

2. From the Microsoft Dynamics GP media, double–click the Setup.exe file to open the installation window. Click **Integration Manager** > **Install**.

   — Or —

   If you're using Windows Vista or Windows Server 2008, choose **Start** > **Control Panel** > **Programs** > **Programs and Features**.

   If you're using Windows 7 or Windows Server 2008 R2, choose **Start** > **Control Panel** > **Uninstall a program**.

3. In the Program Maintenance window, click **Repair** to repair files, registry entries, and configuration information for this installation of Integration Manager.

4. The Repair Program window opens. Click **Repair** to continue.

   The Repair Progress window opens.

5. Click **Finish** when the repair is complete.

Chapter 3: Using Integration Manager

This information describes how to start Integration Manager and how to use the elements of the Integration Manager workspace. It includes the following information.

Starting Integration Manager

You can start Integration Manager from the Start menu (**Start** > **Programs** > **Microsoft Dynamics** > **Integration Manager** > **Integration Manager**), or you can start Integration manager from the Microsoft Dynamics GP application.

In order to run integrations, you must have administrative privileges on the workstation. If you are not the administrator of the workstation, you can run the RegMon or FileMon applications to see what items you do not have security access to.

To start Integration Manager:

1. Start Microsoft Dynamics GP.

2. Choose **Tools** > **Integrate** > **Integration Manager**.

   You also can use the shortcut command ALT+F12 to launch Integration Manager from Microsoft Dynamics GP.

3. If you do not have a default Integration Manager database, the Select Database window opens. Use this window to select the database you want to open. You also can mark the **Set as default database** option.

4. If Integration Manager is not registered, a message appears. Click **Register Now** and enter the correct product registration keys.

*If the **Integration Manager** menu item is not available on the **Tools** menu in Microsoft Dynamics GP, check with your system administrator to be sure you have access to Integration Manager.*

Viewing version and adapter information

Use the following procedure to view version and adapter information for Integration Manager.

To view version and adapter information:

1. Start Integration Manager.

2. Choose **Help** > **About Integration Manager**.

Information about the current version of Integration Manager is displayed in the About IM window.

3. In the **Adapter Data** pane, expand **Destinations** and **Sources**. All the adapters you installed are listed. Expand each adapter that you installed. If an adapter is not installed properly, a message appears.

4. Click **OK** to close the window.

Displaying message details of non-integration errors

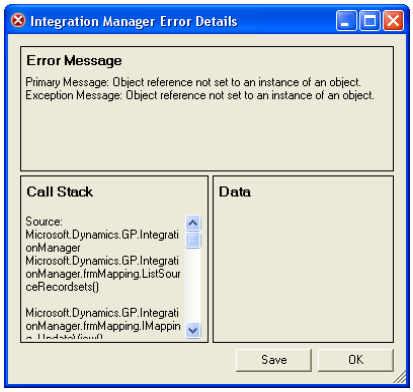
An error outside of the integration might occur when previewing data, mapping an object, or saving an integration. Use the following procedure to display additional information in a message dialog for an error that occurs outside of an integration. In the message dialog that appears, you can choose the Save button to save the error message detail to a text file.

The following areas are displayed in the message dialog if you decide to show details of the error.

Error Message Displays a message or messages that occurred.

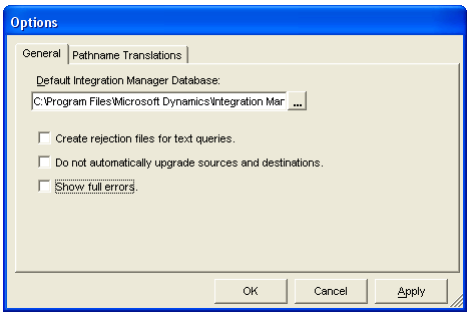
Call Stack Displays a trace of the method calls. You can use this trace to follow the execution path to the line number in the method where the exception occurred.

Data Displays parameters, key values, and field names if there are issues with data assignments or conversion issues.



To display message details of non-integration errors:

1. From the **Tools** menu, choose **Options**.



2. In the **Options** window, choose **Show full errors**.

3. Click **OK** to close the window.

Creating a new database

Use the Create New Database window to create a new Integration Manager database.

To create a new database:

1. Open the Create New Database window.
(**Tools > Create Database**)

2. Browse to the location where you want to save your new database and enter a name for the new database.

We recommend creating new databases outside of the root directory. For example, if your root directory is located at C:/Programs/Microsoft Dynamics/Integration Manager, you could save your database to C:/Programs/Microsoft Dynamics/Integration Manager/Data.

3. Click **Create**.

4. Click **OK** to the message that appears telling you that the database has been created.

Converting a database

Use the Database Conversion window to convert a database from a previous version of Integration Manager to an Integration Manager 2013 database.

To convert a database:

1. Open the Database Conversion window.
(**Tools > Convert Database**)

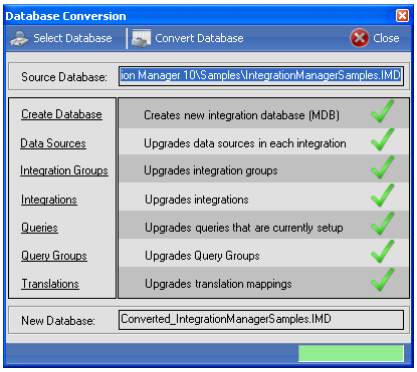
2. Click **Select Database**. Browse to the location of the database to convert and select the database.

3. Click **Open.**

4. Click **Convert Database.** The database is converted and a check mark is added to each step when the conversion for that step is completed successfully. The new file (Converted_<Name of database>) and a log file (Converted__<Name of database>.log) are saved to the same folder location as the original database.

If any of the steps in the list were not completed or converted successfully, a warning icon is displayed by that step. Click the warning icon to find out why the step was not converted successfully. You can view the log file (Converted__<Name of database>.log) to find out more information about a conversion that was not completed successfully. Contact Microsoft Dynamics GP support for more information.

It should look like the following.



5. Click **OK** when the conversion complete message is displayed.

6. When you are finished, click **Close** to close the Database Conversion window. You can click on the converted database file to open Integration Manager.

   You can complete the following optional tasks.

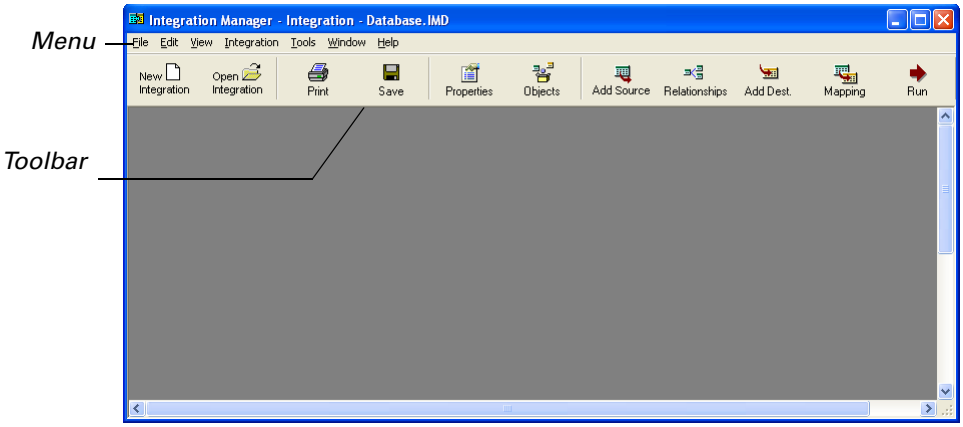
   - Rename the database file.
   - Change the file extension. You can change the file extension to .mdb.

     If the file extension is .imd, you can double click the file to open your Integration Manager database. If you change the extension to .mdb, you'll need to open the database file from Integration Manager.

   - Change the path of the converted file (**Tools > Options**).

Integration Manager workspace

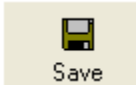
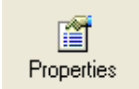
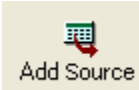
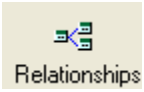
The Integration Manager workspace is where you'll begin all your tasks in Integration Manager. When you start Integration Manager, this workspace is displayed.



You'll use the menu commands and toolbar buttons to complete your tasks. See *Integration Manager toolbar* on page 25 and *Integration Manager menus* on page 26 for more information.

Integration Manager toolbar

The Integration Manager toolbar appears across the top of the main workspace. Each button that appears on the Integration Manager toolbar replaces a commonly used menu command. The following table lists the buttons in the Integration Manager toolbar.

Button	Description
	Create a new integration.
	Open an existing integration.
	Print a report of the integration. You can print reports of other items in the integration using the Object Browser window.
	Save the current integration.
	Display the properties of an integration, source, or destination.
	Open the Object Browser window.
	Add a source to the current integration. Which sources you can add depends on which adapters are installed. If you do not have any adapters installed, you can add only text or ODBC source queries.
	Open the Relationships window, where you create relationships between text or ODBC source queries.
	Add a destination to the current integration. Which destinations you can add depends on which adapters are installed.
	Open the Integration Mapping window.
	Run the current integration.

Integration Manager menus

The Integration Manager menus contain menu commands for creating and managing integrations.

The following table provides the menu commands with their corresponding descriptions and access keys (if available). Access keys are keyboard shortcuts. They are underlined and used in conjunction with the ALT key.

Menu command		Description	Access key
File menu	New Integration	Creates a new integration.	ALT+F+N
	Open Integration	Opens an existing integration.	ALT+F+O
	Close Integration	Closes the current integration.	ALT+F+C
	New Integration Group	Creates a new integration group.	ALT+F+G
	Open Integration Group	Opens an existing integration group.	ALT+F+R
	Save Integration	Saves the current integration.	ALT+F+S
	Import Integrations	Imports integrations from other Integration Manager databases.	ALT+F+I
	Export Integrations	Exports integrations to existing Integration Manager database files. You also can create a new Integration Manager database file for the integrations you export.	ALT+F+E
	Print	Prints a report of the integration.	ALT+F+P
	Exit	Exits the current session of Integration Manager.	ALT+F+X
Edit menu	Remove Script	In the Integration Mapping window, removes the script associated with a field.	ALT+E+S
	Remove Translation	In the Integration Mapping window, removes the translation associated with a field.	ALT+E+T
View menu	Relationships	Opens the Relationships window, where you create the relationship between the source queries used for the current integration.	
	Properties	Displays the properties of the selected integration, source, or destination.	

Menu command		Description	Access key
Integration menu	Add Source	Adds a source to the current integration.	ALT+I+A
	Add Destination	Adds a destination to the current integration.	ALT+I+D
	Remove	Removes the selected source or destination from the current integration.	
	Mapping	Opens the Integration Mapping window for the current integration.	ALT+I+M
	Source Settings	Opens the Source Settings window, where you specify connection information for the source adapters that require these settings.	ALT+I+U
	Destination Settings	Opens the Destination Settings window, where you specify connection information for destination adapters that require these settings.	ALT+I+S
	Run	Runs the current integration.	ALT+I+R
	Properties	Displays the properties of the integration.	ALT+I+E
Tools menu	Object Browser	Opens the Object Browser window, where you can create new data sources (DSNs) and global translations, and view and edit integration groups, integrations, sources, and destinations.	ALT+T+O
	Registration	Opens the IM Registration window, where you can register Integration Manager.	ALT+T+R
	Options	Opens the window, where you can change the location of the Integration Manager database or substitute pathname translations.	ALT+T+P
	Create database	Allows you to create a new database to use in Integration Manager.	
	Convert database	Allows you to convert your databases from previous versions of Integration Manager to use with the current version of Integration Manager.	
Window menu	Tile Horizontally	Lists all open windows and arranges them horizontally.	ALT+W+H
	Tile Vertically	Lists all open windows and arranges them vertically.	ALT+W+V
	Cascade	Displays all open windows in cascading order.	ALT+W+C

Menu command		Description	Access key
Help menu	Contents	Displays the table of contents for the online help for Integration Manager.	ALT+H+C
	Index	Displays the index topics for the online help for Integration Manager.	ALT+H+I
	About This Window	Contains help information about the window displayed on the screen.	ALT+H+W
	About Integration Manager	Provides the version, location, and description of the installed version of Integration Manager and its adapters.	ALT+H+A

The following table contains the access keys assigned to common buttons in the Integration Manager windows.

Button name	Access key
OK	ENTER
Cancel	ESC
Apply	ALT+A
Open	ENTER in some windows; ALT+O in others
Close	ENTER
Help	ALT+H

When a window for Integration Manager is open, you also can press the F1 key to access the online help.

Using the shortcut keys

Shortcut keys, also known as accelerator keys, are used with the CTRL key for menu commands that are used often. Not all menu commands have a shortcut key.

The following table contains a list of the shortcut keys for commonly used menu commands.

Menu command		Shortcut key
File menu	Open Database	Ctrl+D
	New Integration	CTRL+N
	Open Integration	CTRL+O
	Save Integration	CTRL+S
	Print	CTRL+P
View menu	Relationships	CTRL+L
Integration menu	Add Source	CTRL+A
	Add Destination	CTRL+D
	Mapping	CTRL+M
	Run	CTRL+R

Part 2: Building and running integrations

This part of the documentation describes the components of an integration and explains how to create them. The information is presented in the order in which you typically build an integration. Step-by-step procedures are included. Refer to these procedures when you build your own integrations.

This part of the documentation includes the following information.

- Chapter 4, "Creating integrations," explains how to begin building all integrations—by creating the integration object.
- Chapter 5, "Adding sources," explains how to add source data to the integration.
- Chapter 6, "Creating query relationships," explains why you may need query relationships, and how to create them.
- Chapter 7, "Data types," describes how to select the most appropriate data types for source queries.
- Chapter 8, "Adding a destination," describes how to select and set up a destination for your integration.
- Chapter 9, "Creating mappings," explains how to use the Integration Mapping window to map your source data to its destination.
- Chapter 10, "Running integrations," describes how to run an integration after you have created it.
- Chapter 11, "Troubleshooting integrations," provides information about resolving problems in your integrations.

Chapter 4: Creating integrations

The first step in the integration process is to create the integration that contains the source and destination information. You can either define a new integration or use an existing integration.

Step	Description
Step 1	**Create the integration**
Step 2	Add a source
Step 3	Create query relationships (if necessary)
Step 4	Add a destination
Step 5	Create destination mappings
Step 6	Save the integration
Step 7	Run the integration
Step 8	Examine the integration results

This part of the documentation includes the following information.

Parts of an integration

The process of extracting data from source applications or databases and bringing the data into a destination is called an integration. Integration Manager provides you with a safe and easy way of integrating data between business applications. You can integrate data from external business databases, an e-commerce solution, or other data file types into the Microsoft Dynamics GP application.

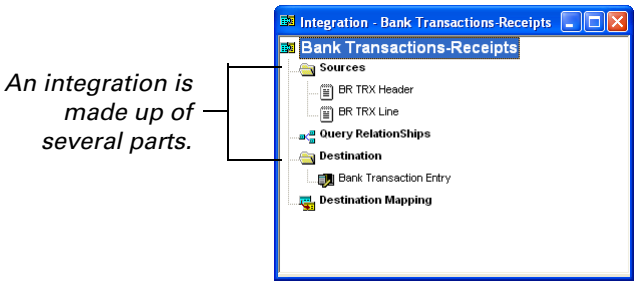
An integration contains sources, destinations, and destination mappings.

- Sources include the data to integrate into a destination. Source data can originate from text files, ODBC databases, and XML files. With Integration Manager, you can integrate text and ODBC source data. You also may integrate source data from XML files.

- A destination indicates where to integrate your source data. Integration Manager allows you to integrate data to destinations in Microsoft Dynamics GP.

- The destination mapping indicates where each item from your source data goes in the destination.

To guide you through the process, the Integration window shows the parts of an integration that you need to set up.



Overview of building and running an integration

The following table describes the basic steps to follow when building and running an integration.

Step	Description
Step 1	Create an integration that contains the source and destination information. You can either define a new integration or use an existing integration. For more information, refer to Chapter 4, "Creating integrations."
Step 2	Add a source, which indicates where information for the integration originates. For more information, refer to Chapter 5, "Adding sources."
Step 3	Create query relationships (if necessary). If you are using more than one text or ODBC source to provide information for the integration, you need to create relationships between the source queries. For more information, refer to Chapter 6, "Creating query relationships."
Step 4	Select the destination for the extracted source information. For more information, refer to Chapter 8, "Adding a destination."
Step 5	Create destination mappings which indicate where each item in the integration destination originates. For more information, refer to Chapter 9, "Creating mappings."
Step 6	Save the integration. You should save the integration periodically as you build it.
Step 7	Run the integration to bring data into your selected destination, For more information, refer to Chapter 10, "Running integrations."
Step 8	Examine the integration results after the integration has finished running. For more information, refer to *Evaluating integration results* on page 82.

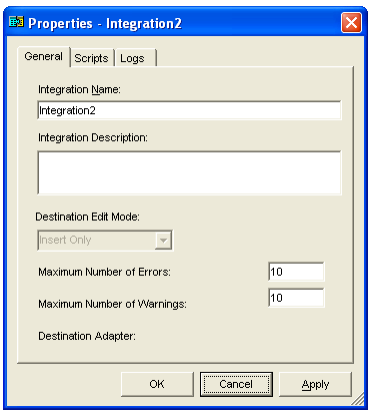
Creating a new integration

The first step in building an integration is to create the integration, which contains information about the sources, destination, and mapping.

To create a new integration:

1. From the Integration Manager toolbar, click **New Integration**.

   The Integration window opens, and the Properties window for the integration opens.



2. Enter an integration name and an integration description.

   Use any integration name that helps you and others easily identify the integration.

   The integration description can provide information about the integration, such as a short description of the source and the destination, as well as a date or a time interval such as "monthly" or "daily."

   If you plan to add a text or an ODBC source, don't use ODBC reserved words in the column names for the ODBC source or destination. These words can prevent the integration from running properly. For a list of ODBC reserved words, go to http://msdn.microsoft.com and search for "ODBC Reserved Words". You'll find a list of reserved keywords in Appendix C of the ODBC Programmer's Reference.

*You do not need to define the other properties on the **General** tab until later. In fact, you can't define some of them, such as the destination edit mode, until you add a destination to the integration. From the Properties window, you also can add scripts to the integration (see Chapter 20, "Using scripts") and set properties for the integration log (see Chapter 14, "Managing logs").*

3. Click **OK** to close the Properties window.

   The name of your integration is displayed in the Integration window.

*If you are using a master level destination, select a destination edit mode in the Integration Properties window (**Integration** > **<integration name> Properties**) before you save the integration. See Chapter 10, "Running integrations," for information about setting the destination edit mode.*

4. Choose **File** > **Save Integration** to save the integration. You also may want to make a backup of the Integration Manager database file to protect against losing the integrations you created.

Integrations are stored in a database file. You can specify the location of the Integration Manager database file using the Options window. Choose **Tools** > **Options** to open the Options window.

See Chapter 5, "Adding sources," to add a source to this integration.

Using an existing integration

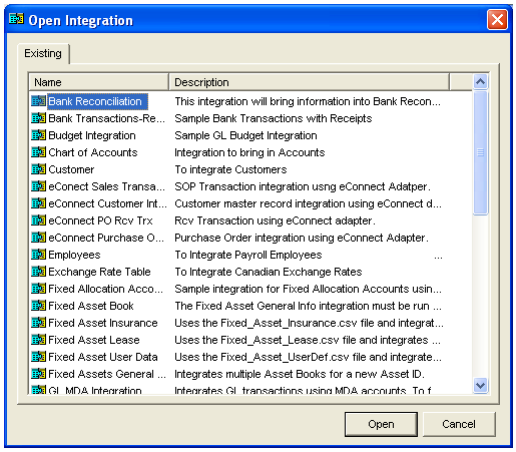
Use an existing integration if you periodically run an integration or if you need to finish or edit an integration you already started to build.

If you are creating an integration that is similar to an existing integration, you may want to start with a copy of the existing integration rather than creating the integration from scratch.

To use an existing integration:

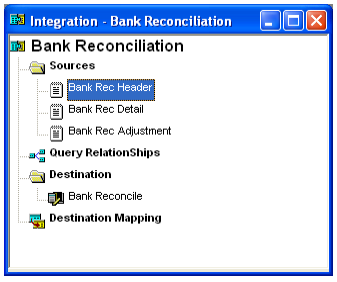
1. From the Integration Manager toolbar, click **Open Integration**.

   The Open Integration window opens.



2. Select the name of the integration to open, and click **Open**.

The Integration window opens.



If the integration you opened is complete and you are ready to run it, refer to Chapter 10, "Running integrations."

3. To make a copy of this integration, choose **File** > **Import Integrations** or **Export Integrations**.

   If you are importing the integration, refer to *Importing integrations* on page 91.

   If you are exporting the integration, refer to *Exporting integrations* on page 91.

   This creates another integration that contains the same source and destination. You will be prompted to rename the integration name and integration source names if you are importing into the same Integration Manager database file.

4. Edit the integration, as necessary.

   *If you are using a master level destination, select a destination edit mode in the Integration Properties window (**Integration** > **<integration name> Properties**) before you save the integration. See Chapter 10, "Running integrations," for information about setting the destination edit mode.*

5. Choose **File** > **Save Integration** to save the integration. You also may want to make a backup of the Integration Manager database file to protect against losing the integrations you created.

   Integrations are stored in a database file. You can specify the location of the Integration Manager database file using the Options window. Choose **Tools** > **Options** to open the Options window.

If you are ready to run your integration, refer to Chapter 10, "Running integrations."

Chapter 5: Adding sources

The next step in setting up your integration is to add your source.

Step	Description
Step 1	Create the integration
Step 2	**Add a source**
Step 3	Create query relationships (if necessary)
Step 4	Add a destination
Step 5	Create destination mappings
Step 6	Save the integration
Step 7	Run the integration
Step 8	Examine the integration results

When you add a source to an integration, you are defining which data from your source to later integrate into a destination. With Integration Manager, you can integrate source data from text files, XML files, or ODBC sources.

This part of the documentation includes the following information.

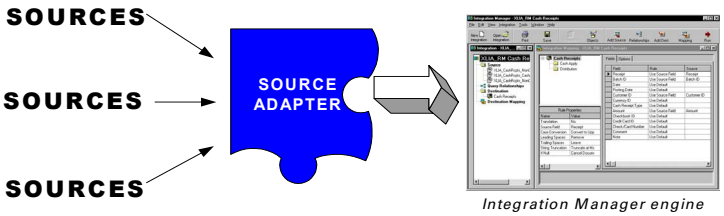
Understanding sources

A source indicates where the requested information you are integrating originates. In Integration Manager, a source can be a comma- or tab-delimited file, XML file, or a database such as an ODBC source. Sources exist independently of the source adapters; however, the sources you can add to your integration depend on the source adapters you have installed.

Understanding source adapters

Source adapters connect to a source (such as delimited text files, XML files, or ODBC databases), and extract data before passing it to the Integration Manager engine.

The ODBC/text adapter extracts data from ODBC and text sources. Integration Manager supports three types of ODBC/text sources: Text, Simple ODBC, and Advanced ODBC. Text sources retrieve data from text files. Simple ODBC sources retrieve data from an ODBC data source. Advanced ODBC sources issue SQL statements from SQL Server to retrieve information from an ODBC data source.



Integration Manager engine

Adding sources to an integration

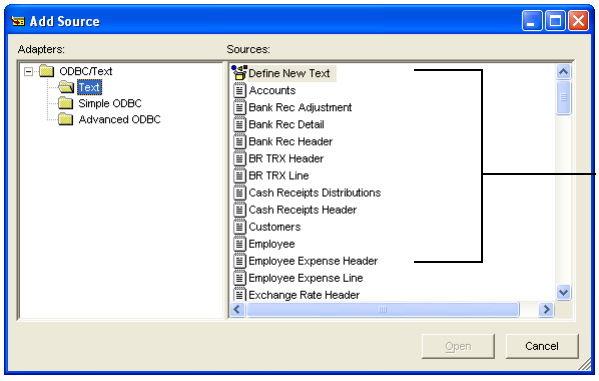
Depending on which source adapter you are using, you can add new or existing sources to an integration. For ODBC and text sources, you may be able to add more than one source to an integration. For the XML source adapter, however, you can add only one source at a time. For this adapter, you need to remove the existing source before adding another source to the integration. See *Removing a source from an integration* on page 53 for more information. If you try to add more than one source to an integration that contains an existing source, a message appears.

To add sources to an integration:

1. Open the integration to add a source to.
2. From the Integration Manager toolbar, click **Add Source**.

You also can right-click on the Sources item in the Integration window, and from the menu that appears, click ***Add Source***.

The Add Source window opens.



Define a new source or select from the existing sources to add to your integration.

3. From the **Adapters** pane, select the source adapter to use.

4. From the **Sources** pane, select the name of the source or, if available, click **Define New Source**, and click **Open**.

   The window that opens next—a Properties window, a Source Settings window, or no window—depends on which source adapter and source you are using.

   - A Properties window opens if you select to define a new source.
   - A Source Settings window opens if you select one of several types of sources requiring source settings. Source settings connect the actual source to the source adapter in Integration Manager. These settings include, but are not limited to, File Path, Filter, Use Filter, User Name, and Password.
   - No window opens if you select a source with properties that are already defined.

5. If you are defining a new source, set up the source definitions.

   For more information about adding text and ODBC sources, see *Using ODBC and text sources* on page 39.

   For more information about setting up XML source definitions, see *Setting up XML source definitions* on page 49.

6. If you are adding a source that uses source settings, define them on a Source Settings window. Source settings connect the actual source to the source adapter in Integration Manager. These settings include, but are not limited to, File Path, Filter, Use Filter, User Name, and Password. For more information about specifying XML source settings, refer to *XML source settings* on page 51.

   After you select a source and, if necessary, set up source definitions and source settings, the source appears in the Integration window, and you can continue to build your integration.

*You can drag a source file from the desktop or Windows Explorer onto the Integration Manager workspace and the Integration Source Properties window automatically opens. For the new source, the default **Delimiter** setting is **Comma**. Change the **Delimiter** and **First Row Contains Column Names** settings to match the file information.*

7. Choose **File > Save**.

Using ODBC and text sources

Integration Manager comes with a source adapter that reads source data from text and ODBC sources. To retrieve data from these sources, it issues queries, which you set up when you add a text or ODBC source.

Integration Manager can issue queries for text sources (see *Adding text sources* on page 40), simple ODBC sources (see *Adding simple ODBC sources* on page 40), and advanced ODBC sources (see Chapter 18, “Using advanced ODBC source queries”).

Several of the following topics in this part of the documentation describe how to create text and simple ODBC sources. Chapter 18, “Using advanced ODBC source queries,” describes advanced ODBC sources.

Adding text sources

If your source data is in a text file, add and set up a text source.

To add a text source:

1. Open the integration to add a source to, or create a new integration.
2. From the Integration Manager toolbar, click **Add Source**.

   The Add Source window opens.
3. In the **Adapters** pane of the Add Source window, expand the **ODBC/Text** folder and select **Text**.

   The available text sources appear in the **Sources** pane.
4. From the Sources pane, select an existing text source or select **Define New Text** and click **Open**.

   If you select an existing source, the source is added to your integration. You don't need to set up the source properties.

   If you are creating a new source, the Properties window opens, allowing you to set the properties of the text query. See *Setting up general properties for text sources* on page 41.

If you add a text source to an integration that already contains ODBC/text sources, a message appears reminding you to create a relationship between the sources.

Adding simple ODBC sources

If your source data is in a single ODBC table or view, which is the case for most SQL/ODBC databases, then add a Simple ODBC source to your integrations.

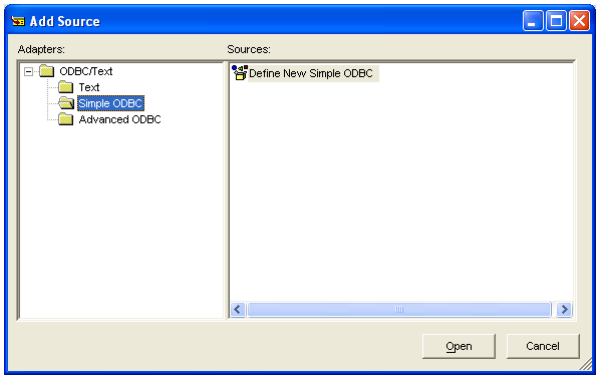
To add a simple ODBC source:

1. Open the integration to add a source to.
2. From the Integration Manager toolbar, click **Add Source**.

   The Add Source window opens.
3. In the **Adapters** pane of the Add Source window, expand the **ODBC/Text** folder, and select **Simple ODBC**.

   The available Simple ODBC sources appear in the **Sources** pane.

4. From the **Sources** pane, select an existing Simple ODBC source or select **Define New Simple ODBC** and click **Open**.



If you select an existing Simple ODBC source, the source is added to your integration. You don't need to set up the source properties.

If you are creating a new Simple ODBC source, the Properties window opens, allowing you to set the different properties of the Simple ODBC query. See the topic *Setting up general properties for ODBC sources* on page 42.

If you try to add an existing simple ODBC source to an integration that already contains ODBC/text queries, you are asked to create a relationship between the new and existing queries.

Setting up general properties for text sources

When you add a new text source to an integration, the Properties window opens. Use the following procedure to set the options on the **General** tab of the Properties window.

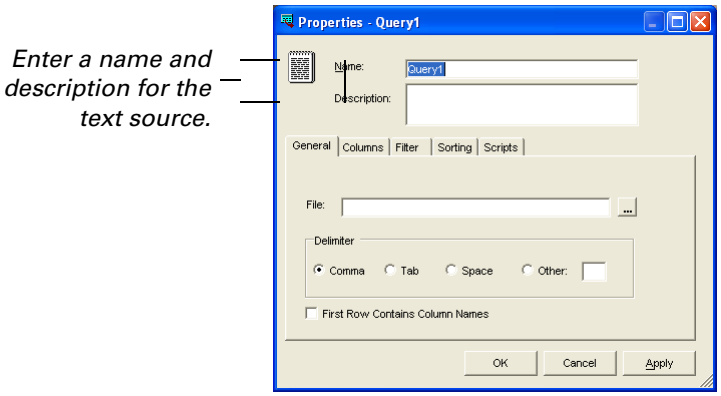
To set up general properties for a text source:

1. Open the properties window for a new text source
   Open an integration > Add Source > Text > Define New Text

2. Specify the **Name** and **Description** of the text source.

   The name should describe the type of information retrieved by the text query. An optional description should provide information about the source, such as

what type of data is retrieved by the query or what data is contained in the source.



A source can be used by multiple integrations, so be sure to provide enough information so the source can be used easily by another Integration Manager user.

Don't use ODBC reserved words in the source name. They can prevent the integration from running properly. For a list of ODBC reserved words, go to the MSDN online library (http://msdn.microsoft.com) and search for "ODBC reserved words". You'll find a list of reserved keywords in Appendix C of the ODBC Programmer's Reference.

3. In the **File** field, enter or select the path to the text file that is being used as the source.

4. Specify the delimiter used for the text file, and whether the first row of the text file contains column names.

   Delimiter The delimiter indicates which character or characters separate the individual data items in the text file. For example, a comma-delimited text file has a comma between each data item.

   First Row Contains Column Names Select this check box to use the names in the text file when referring to columns. Otherwise, Integration Manager provides a set of default names.

After setting up the general properties of the text source, you need to set up the other source properties. Refer to *Setting additional properties for ODBC or text sources* on page 43 for more information. Be sure to save your source when you are finished.

Setting up general properties for ODBC sources

When you add a new simple ODBC source to an integration, the Properties window opens. Use the following procedure to set the options on the **General** tab of the Properties window.

To set up general properties for an ODBC source:

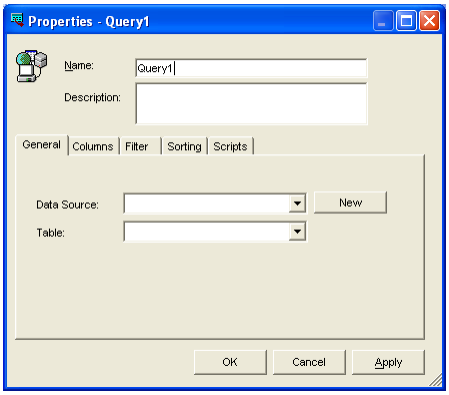
1. Open the properties window for a new simple ODBC source.
   Open an integration > Add Source > Simple ODBC > Define New Simple ODBC

2. Specify the name and description of the simple ODBC source.

   The name should describe the type of information retrieved by the ODBC query. The description should provide information about the source, such as what type of data is retrieved by the query or what data is contained in the source.

Don't use ODBC reserved words in the source name. They can prevent the integration from running properly. For a list of ODBC reserved words, go to the MSDN online library (http://msdn.microsoft.com) and search for "ODBC reserved words". You'll find a list of reserved keywords in Appendix C of the ODBC Programmer's Reference.

3. On the **General** tab of the Properties window, select the data source and table from which the information will be retrieved.



Data Source This is the ODBC data source to retrieve data.

Table This is the table to use from the ODBC data source.

4. If you're using an .xls file type, enter a named range. Select the data range, including column headings, and choose **Insert** > **Name** > **Define**. Type the new name for the reference and click **Add**.

5. After setting up the general properties of the simple ODBC source, you need to set up the other source properties. See *Setting additional properties for ODBC or text sources* on page 43. Be sure to save your integration when you are finished.

Setting additional properties for ODBC or text sources

After creating and setting up the general properties of the ODBC or text sources, you need to set up the columns, rows, sorting, and script properties of the source.

To set up additional properties for an ODBC or text source:

1. Click the **Columns** tab to specify the column properties for the source.

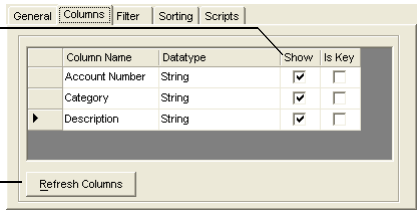
   If you are creating a text source and the text file does not contain column names, you may want to edit the column names to make them more meaningful.

If you rename any of the column names, you must rename all of the column names in order for the integration to be successful.

Column Name These are the names of the data items that are available for the source. If you marked the option indicating that the first row of the text file contained column names, those names appear here. If the text file does not contain column names, Integration Manager supplies its own names. You can edit the column names to make them more meaningful.

*Select the **Show** check box to include the column in the source results.*

*If the characteristics of the text file or ODBC data source have changed since you created the source definition, click **Refresh Columns** to update the column list. The list will be updated with any changes that have been made to the text file.*



Don't use ODBC reserved words in the source name. They can prevent the integration from running properly. For a list of ODBC reserved words, go to the MSDN online library (http://msdn.microsoft.com) and search for "ODBC reserved words". You'll find a list of reserved keywords in Appendix C of the ODBC Programmer's Reference.

Datatype This value indicates what type of data is contained in the column. If you are creating a text source, Integration Manager examines the data in the text file and provides a default value. If you're using a simple ODBC data source, the data type value is automatically retrieved from the ODBC data source and can't be changed.

Column names and the data types you have chosen are stored in the Schema.ini file, which is in the same location as the text file.

You can change the value in the **Datatype** column, based on the type of data in the text file and how the information is used in the destination of the integration. Refer to Chapter 7, "Data types," to help determine the most appropriate data type for each column in the text file.

Size If you are creating a simple ODBC source, the **Size** column indicates the data size of each column, in bytes. You can't edit this value.

Show Select the **Show** check box to include the data in this column in your integration. If you are using a large source file and you do not want all the data to integrate, clear this check box for the columns to exclude.

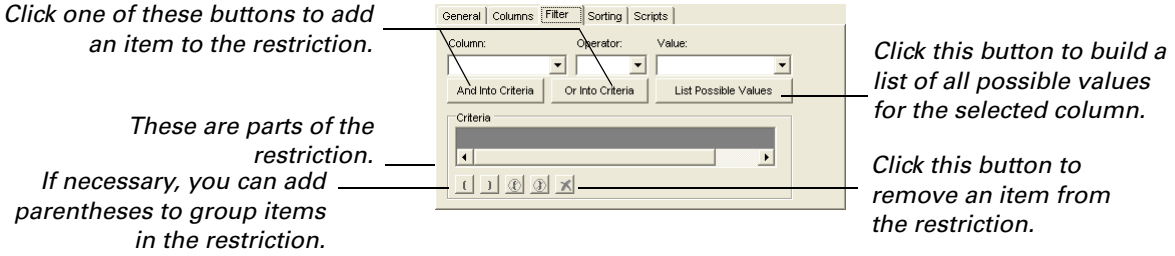
Is Key Select the **Is Key** check box to indicate that the data items within a column are unique identifiers. For example, you might select a column called Customer ID as Is Key to indicate that the values within a column are unique.

Integration Manager uses the values selected as Is Key to identify specific rows that cause errors in the integration.

You don't need to select this check box for any columns.

*If the properties of the text file or the ODBC data source used for the source have changed, click the **Refresh Columns** button to update the columns list.*

2. Choose the **Filter** tab to display the restriction criteria for the source (optional). Restrictions allow you to specify the rows that will be included in the source. All rows that do not fit the criteria are excluded from the source results.



A restriction is composed of several individual criteria. You can use logical AND and logical OR operators to apply several criteria. If necessary, you can include parentheses to group items in the criteria.

The LIKE operator allows you to perform basic pattern matching with string data types. You can use the percent sign (%) as a wildcard character, representing any sequence of characters. The percent sign can be used multiple times within a single string. For example, to include all customers whose names include the word "ACME", the expression would be similar to the following:

CustomerName Like %ACME%

To exclude all rows where a particular field is blank, use a single wildcard character. For example, to exclude all customers where the "CustomerName" field is blank, the expression would be similar to the following:

CustomerName <>0.

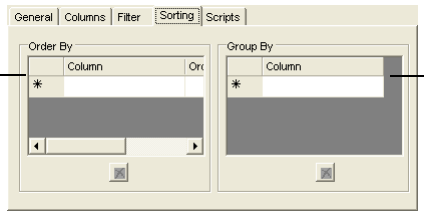
3. Choose the **Sorting** tab to specify sorting and grouping information for the source.

By adding columns to the **Order By** list, you indicate the order rows should appear in the source. Individual columns can be sorted in ascending or descending order. Specifying a sort order is needed when the data in the source text file is not in the desired order. This is useful for integrations such as General Ledger transactions, where the order in which items are added is important.

For example, you typically want General Ledger transactions created in transaction date order. If the transactions in the source text file are not in this order, you can use the sorting capability to import them in the desired order.

*To remove a column from the list, click the line selection button (to the left of the item) to select the entire line, and press **Delete**.*

To sort items in the query, add columns to this list. Individual columns can be sorted in ascending or descending order.



Grouping allows you to "collapse" the rows in a query.

Sorting also can improve the performance of integrations. If multiple sources are used for an integration, sort the source results by the columns that are used to define the relationship. For example, if a relationship between two sources is based on the DocNumber column, sort both sources by that column.

The **Group By** list allows you to "collapse" the rows in the source that have identical values for corresponding columns. To compare values, specify the columns containing the values by adding them to the **Group By** list. These columns must also have the **Show** option marked in the **Columns** tab. When the rows are collapsed, only one row is included in the source results for any group of rows for which all of the selected columns match.

If you group the items in a source, only columns included in the group are included in the source results. No other columns are included in the source results.

Grouping is typically used when you have header information and line item information in a single source file. For example, General Ledger transactions are composed of a transaction header and line items. The transaction header contains the transaction date and reference. The line items contain information for each item in the transaction.

The following text file contains General Ledger transaction information.

These groups make up individual transactions.

Trx Date	Reference	Account	Amount
12-27-98	Xfer to cash	000-1100-00	-1000.00
12-27-98	Xfer to cash	000-1130-00	800.00
12-27-98	Xfer to cash	000-1140-00	200.00
12-27-98	Xfer from warehouse	000-1300-01	500.00
12-27-98	Xfer from warehouse	000-1310-01	-500.00
12-28-98	Xfer from warehouse	000-1300-01	1200.00
12-28-98	Xfer from warehouse	000-1310-01	-1200.00

To get the transaction header information from the text file, the items in the file need to be "collapsed." Each transaction is uniquely identified by the transaction date and reference, but several linesin the text file have the same transaction date and reference. To collapse the file, the Trx Date and Reference columns are added to the **Group By** list. The results of the source are shown in the following illustration.

Note that only the columns in the Group By list appear in the results of the query.

Trx Date	Reference
12-27-98	Xfer to cash
12-27-98	Xfer from warehouse
12-28-98	Xfer from warehouse

The results contain only one row for each unique transaction date and reference combination. The only columns included in the source results are those in the **Group By** list.

4. To attach a script to the integration (optional), choose the **Scripts** tab.

   Scripts are written in VBScript, a subset of the Microsoft Visual Basic® programming language. Refer to Chapter 20, "Using scripts." for more information about using scripts.

5. After you set up properties for the text or simple ODBC source query, click **OK**. The source you created is added to the Integration window.

*Be sure to save your source after setting the source properties. From the **File** menu, click **Save Integration**.*

Understanding the XML source adapter

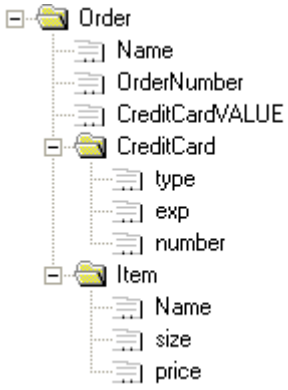
When Integration Manager extracts data from your XML source, it uses the properties in the XML Source Definition window to convert the XML data into a document definition it can use. For example, while XML uses terms like "element", Integration Manager uses the terms recordsets and fields. The document definition serves as a structure that not only holds the XML data but also allows you to map the XML structure of the source to another type of structure in the destination.

You will see how Integration Manager uses this document definition later when you map the source to a destination. In the meantime, the following example shows how the XML source adapter converts XML for Integration Manager. You can use the table at the end of the example as a reference when you specify properties in the XML Source Definition window.

If you were to use the following XML as source data:

```
<Orders>
    <Orders>
        <Name>My Order</Name>
        <OrderNumber>123</OrderNumber>
        <CreditCard>
            <Type>americharge</Type>
            <exp>05/01/07</exp>
            <Number>1234112233224455</Number>
        </CreditCard>
        <Item>
            <Name>T-Shirt</Name>
            <size>Medium</size>
            <price>12.99</price>
        </Item>
    </Orders>
</Orders>
```

It would appear as the following tree view in Integration Manager:



The following table shows how Integration Manager interprets XML. The second and fourth columns include information from the previous example.

This component in XML...		**Becomes this Integration Manager component...**	
Root element	<Orders>	Root recordset	Order
An element containing data	<OrderNumber>123</ OrderNumber>	A field (All fields lie within a recordset.)	OrderNumber
An element containing child elements. **Note:** Attributes are not supported.	<Item>	A recordset	Item

The root recordset is not necessarily the root element in the XML source file. Instead, the XML source adapter looks for the first XML element that matches the value of the root recordset's XML Node as defined in the XML Source Definition window. In the previous table, the display name of "Order" (in the fourth column) matches the XML root element name of "Order." You can change the display name of the root recordset, other recordsets, and fields in the XML Source Properties window.

Setting up XML source definitions

You can set up the source definitions in the XML Source Definition window using the **Template** button or by making direct changes in the window. The **Template** button approximates the structure of the data you are importing into Integration Manager. Be sure to verify the definition. For example, Sales Order Processing line items may be assigned serial numbers or lot numbers. If the definition for the Sales Order Processing template uses serial numbers but not lot numbers, a recordset for lot numbers is not added to the definition.

To set up XML source definitions:

Integration Manager uses recordsets and fields to define an XML source. To generate XML from this hierarchy of recordsets and fields, the XML source adapter uses the properties in the XML Source Definition window. Collectively, these properties make up the source document definition. See *Understanding the XML source adapter* on page 47 to learn more about how the XML source adapter converts XML into a structure it can use.

1. Open the XML Source Definition window.

   In the Object Browser window, expand **Source Adapter**. Select **Microsoft Dynamics GP XML Source Adapter** and click **New**.

   – Or –

   In the Object Browser window, expand **Source Adapter**. Select **Microsoft Dynamics GP XML Source Adapter**, and then select an existing source. Click **Properties**.

2. Create or define the root recordset.

   The root recordset represents the object that contains the entire source document definition, including the root recordset and its properties, all child recordsets of the root and their properties, and all fields within all recordsets and their properties. The following table describes the components and properties of a root recordset.

Properties	Description
Display Name	The name of the Root Recordset. Use any name that is appropriate for the source you are defining. Typically, this is the name of the object that this Root Recordset represents.
XML Name	The name of the source document definition that you are setting up. This name appears in the Add Source window. Use any name that helps you easily identify this source document definition.

3. Create or define the recordset.

A recordset is represented as a folder in the tree view. A recordset is a child of the root recordset or another recordset and represents either an XML element that contains child recordsets. A recordset also represents part of a mixed XML element.

Properties	Description
Display Name	The name of the recordset. Use any name that is appropriate for the source you are defining.
XML Name	The name of the XML node in the source document that this recordset represents. This name must match the one in the source.

4. Create or define the field.

   A field represents an XML element containing only data.

Properties	Description
Display Name	The name of the field.
XML Name	The name of the XML name in the source document that this field represents. This name must match the one in the source.
Data Type	The type of data this field contains, such as String, Boolean, Currency, Date, Double, or Long Integer. Click the drop-down arrow in this field to select an appropriate data type.
Key Field	Click **Yes** to indicate that the field contains unique identifiers.

5. When you are finished, you can save or close the window.

Using a template for XML source definitions

Before you integrate XML source data through Integration Manager, you must set up a document definition that describes the structure of the source in terms Integration Manager can use. While XML uses "elements" to describe the structure of data, Integration Manager uses the terms recordsets and fields. You define the source document definition by setting up properties in the XML Source Definition window.

To set the XML source definitions, use the Template button on the XML Source Definition window. The template approximates the structure of an XML source that you select and converts it into recordsets and fields—a structure Integration Manager can use.

When using the template, you select an XML data file. To quickly approximate the source structure, the XML source adapter scans only the first few records of the XML source you selected, making the template a convenient tool for starting the document definition process.

To use a template for XML source definitions:

1. In the XML Source Definition window, click **Template**.

2. From the Open window, select the XML data file that uses the structure you want to use in the XML Source Definition window and then click **Open**.

The file you select does not need to be the file you use as your source. It simply needs to represent the same data structure as your source. Later, when you specify source settings, you set the path and file name of the actual source.

After you click **Open**, the XML Source adapter approximates the structure of the data, converts it into a hierarchy of recordsets and fields, and then displays this new structure in the XML Source Definition window.

3. Review and, if necessary, edit the properties.

The default value for **Data Type** is "String," and the default value for **Key Field** is "No." You may want to change these default values, especially for **Data Type**. See *Setting up XML source definitions* on page 49 for more details.

4. Save, and then close the window.

The Source Settings window opens if you are assigning the XML source to an integration. You can use this window to connect this document definition to the source containing the data to integrate. At this point, you are no longer setting properties that are attached to the source. Source Settings are properties of the integration. See *XML source settings* on page 51 for more information.

Editing XML source definitions

You can make changes to any of the XML source definitions after you have closed the XML Source Definition window.

To edit XML source definitions:

1. In the Object Browser window, expand **Source Adapter**. Select **Microsoft Dynamics GP XML Source Adapter**.

2. Double-click **<name of source>** or select an existing source and click **Properties**.

The XML Source Definition window opens so that you can make the necessary changes.

XML source settings

When you add a source to an integration, you are attaching a description of the source data structure and content to the integration. That description, known as the *source document definition*, exists independently of the integration you are working on. You can use this source again in other integrations.

Be careful when making changes to a source document definition. If it is attached to other integrations, those integrations might not run correctly.

There are additional properties relating to a source that are associated with the individual integration. These properties are not automatically inherited by other

integrations that use the same source document definition. These additional properties are called source settings.

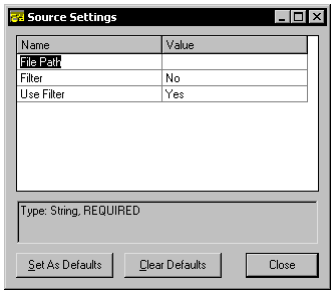
Defining XML source settings

Source settings connect the source document definition to an actual source by having you specify a file name. The Source Settings window also provides a filter that you can use to define precisely which documents from the source to integrate.

To define XML source settings:

1. Open the Source Settings window.

   The Source Settings window automatically opens when you first add a new or existing XML source definition to an integration.You can use this window to specify the physical data file to be used for the integration. If you already added a source to the integration, you can double-click **Source Settings** in the Integration window.



2. Click in the **Value** column next to File Path, and then click the **Lookup (...)** button. The Open dialog box opens.

3. Navigate to the source file and select it. Then, click **Open**. You also can type or paste the path and file name.

   By specifying a file path in the Source Settings window, you are connecting the document definition of the source to the actual source.

Previewing sources

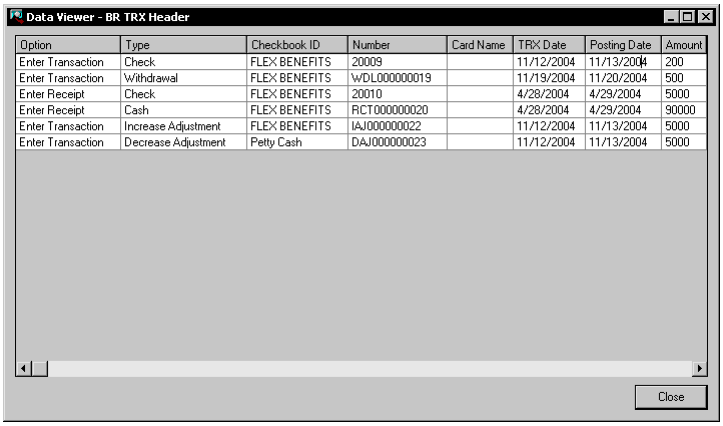
Before you run an integration, it's useful to know that the source you have created is returning the data you expect.

To preview a source:

1. From the Integration window, right-click the source to preview.

2. On the menu that appears, choose **Preview <name of source>**.

Depending on which type of source you are previewing, either the Data Viewer or the Source Preview window opens. For those source adapters that use source settings, you can preview the source only if you have specified accurate settings on the Source Settings window.



3. Examine the data returned by the source to verify that the source is working properly. If you are using a text or ODBC source, search for empty fields or incorrectly formatted data, such as account numbers that have characters missing, and check that the data types are appropriate. See Chapter 7, "Data types," for more information about selecting the appropriate data types.

4. Close the Data Viewer or the Source Preview window.

Removing a source from an integration

Use the following procedure to remove a source from an integration.

To remove a source from an integration:

1. From the Integration window, right-click the source.

2. Choose **Remove <name of source>**.

*You also can select the source to remove and press the **Delete** key.*

3. In the confirmation window, click **Yes**.

Query relationships

When you use ODBC and text sources, Integration Manager issues queries to the sources to retrieve the source data. When you use more than one ODBC or text source in an integration, you need to create relationships among the sources. The relationships tell Integration Manager how the various queries to the sources should work.

For more information about how to set up query relationships, refer to *Creating query relationships* on page 55.

Chapter 6: Creating query relationships

When you use ODBC and text sources in your integration, Integration Manager issues queries to the sources to retrieve the source data. When you use more than one ODBC or text source in an integration, you need to create relationships among the sources. The relationships tell Integration Manager how the various queries to the sources should work.

Step	Description
Step 1	Create the integration
Step 2	Add a source
Step 3	**Create query relationships (if necessary)**
Step 4	Add a destination
Step 5	Create destination mappings
Step 6	Save the integration
Step 7	Run the integration
Step 8	Examine the integration results

This part of the documentation includes the following information.

Relationship guidelines

When creating query relationships, keep the following guidelines in mind.

- There must be only one "root" query that has no arrows pointing into it. This is the main query that is executed. All other queries must be related to the "root" query in some way.
- Circular relationships are not allowed. For example, if Query A has a relationship to Query B, Query B can't have a relationship back to Query A.
- You can link one query to another using more than one column.

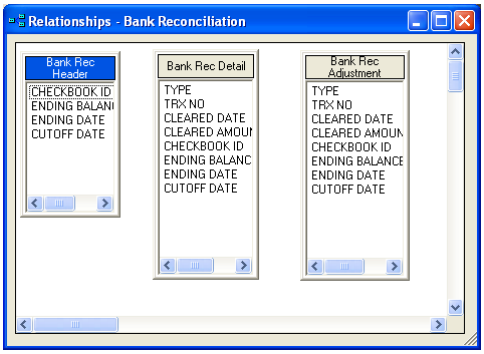
Creating query relationships

A relationship defines the dependency each source has on another. Typically, there is a master source and its related child sources. You can specify the type of relationship between the sources.

To create a query relationship:

1. Open the integration that contains the ODBC or text sources you're creating a relationship for.
2. In the Integration window, double-click **Query Relationships**.

The Relationships window opens.

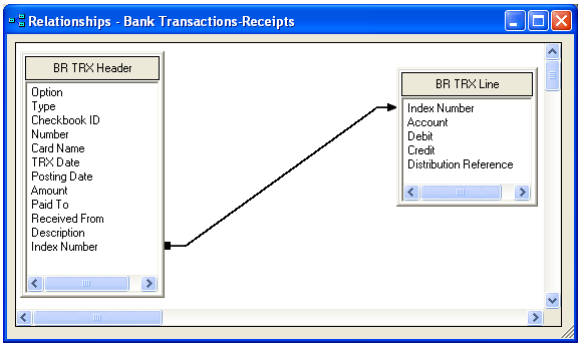


3. Examine the queries you are using for the integration. Decide which query is considered the "root" and which queries are considered child queries. Remember that each query must be directly or indirectly connected to the root. No query or group of queries can remain unconnected.

4. Integration Manager uses a graphical representation to show the relationships among the queries used for an integration. To add the query relationship, draw a line by dragging your mouse pointer between the corresponding column or columns in the queries. The query you draw the line from is called the *master*. The query you draw the line to is called the *child*. When Integration Manager reads a record from the master query, it also reads the appropriate number of records from the child queries.

   For example, two queries are used to retrieve bank transaction information. One query retrieves basic information, including checkbook ID, number, and amount. The other retrieves line detail information. To allow Integration Manager to work with both queries, a relationship must be created between them. In this case, the BR TRX Header query would be considered the master. Each time Integration Manager reads a record from this query, the corresponding record should be read from the BR TRX Line query.

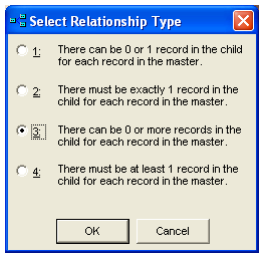
   To set up this relationship, you need to draw a line between the corresponding columns in the two queries. In this case, the corresponding column is **Index Number**. You can resize the window to show more of the query information.



To improve the performance of your integration, we recommend that you sort the queries based on the columns used for the query relationship. To sort the queries, open the Properties window for the source, and choose the ***Sorting*** *tab.*

5. To set the relationship type, right-click the line connecting the two queries and choose **Properties**.

   The Select Relationship Type window opens.



   The relationship type indicates how many records exist in the child query for each record in the master. The four types of query relationships are summarized in the following table.

Type	Description
1	There are zero records or one record in the child for each record in the master.
2	There is exactly one record in the child for each record in the master.
3	There can be zero or more records in the child for each record in the master. This is the default relationship type.
4	There must be at least one record in the child for each record in the master.

The default relationship type is 3, indicating there are zero or more records in the child query for each record in the master.

6. Select the appropriate relationship type and click **OK**.

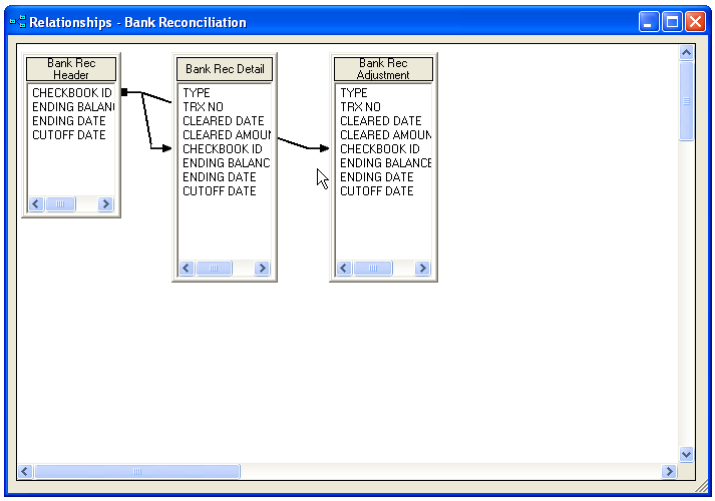
7. Click **Close** to close the Relationships window. When you close the Relationships window, all of the relationships you created are verified. If any problems exist, a message appears. Reopen the Relationships window and correct any problems.

Removing query relationships

Use the following procedure to remove a query relationship.

To remove a query relationship:

1. Open the Relationships window.
   Create or open an integration > click Relationships



2. Select the relevant fields and right-click the line between the two queries.

3. Click **Remove**.

Chapter 7: Data types

When adding source queries to your integrations, use the following information to help you select the most appropriate data type.

Data type		
Boolean	Destination Object Type	Boolean
	Acceptable Data types	Boolean, Byte, Integer, String
	Preferred Data type	Boolean
	Additional information	If using Boolean as the data type, the following values are Acceptable as true or false: **TrueFalseTrueFalse** TFYN tfyn TrueFalseYesNo TRUEFALSEYESNO 10Non-zero0
Currency	Destination Object Type	Currency
	Acceptable Data types	Currency, Double, Single, String
	Preferred data type	Currency
	Additional information	If the currency data in the text file is formatted, for example, if it has a dollar sign ($), you must use a Currency data type. If the currency data in the text file has more than six significant digits, do not use the Single data type. If the currency data has more than 14 significant digits, do not use the Double data type. Instead, use the Currency or String Data types. If the currency values in the text file have more than four decimal places, you must use the String data type.

Data type		
Date	Destination Object Type	Date
	Acceptable Data types	Date, String
	Preferred data type	Date
	Additional information	If you are using date information to restrict the source, you must use the Date data type. The acceptable formats for date values are based on the date settings in the Windows Regional Settings control panel. The following examples assume that you have specified a standard U.S. date format in the Regional Settings control panel. If you use the Date data type, the data in the column must be in one of the following formats: 9–25–98 9–25–1998 9/25/98 9/25/1998 SEP–25–98 If you use the String data type, the data in the column can be in any of the following formats: 9–25–98 25–9–98 9/25/98 25/9/98 SEP–25–98 25–SEP–98 September 25, 1998 25 September 1998 1998–9–25 1998–SEP–25 Other formats may work as well.
Enumeration	Destination Object Type	Enumeration
	Acceptable Data types	Integer, String
	Preferred data type	Integer
Integer	Destination Object Type	Integer
	Acceptable Data types	Integer, String
	Preferred data type	Integer
Long integer	Destination Object Type	Long integer
	Acceptable Data types	Long integer, String
	Preferred data type	Long integer

Data type		
Numeric	Destination Object Type	Numeric
	Acceptable Data types	Currency, Double, Integer, Long integer, Single, String
	Preferred data type	String
	Additional information	If the numeric data in the text file has more than 6 significant digits, do not use the Single data type. If the data has more than 14 significant digits, do not use the Double data type. Instead, use the Currency or String data types. If the numeric values in the text file have more than four decimal places, you must use the String data type.
String	Destination Object Type	String
	Acceptable Data types	String
	Preferred data type	String
	Additional information	If the string is over 255 characters long, you must use the LongVarChar data type. If the total length of the items in the source is over the allowed limit of 2048 bytes, you can reduce the size by using a LongVarChar data type instead of a String data type.
Time	Destination Object Type	Time
	Acceptable Data types	String
	Preferred data type	String
	Additional information	Time values must have the form HH:MM:SS and be in 24-hour format.

Chapter 8: Adding a destination

After you add a source and, if necessary, specify source settings, you can begin to add your destination.

Step	Description
Step 1	Create the integration
Step 2	Add a source
Step 3	Create query relationships (if necessary)
Step 4	**Add a destination**
Step 5	Create destination mappings
Step 6	Save the integration
Step 7	Run the integration
Step 8	Examine the integration results

A destination indicates where the information should go. Integration Manager has destination adapters that validate data before integrating it to the destination application or database.

A destination can be used by multiple integrations. If you change the properties of a destination, it changes the properties of all instances of the destination.

This part of the documentation includes the following information.

Understanding destination adapters

Destination adapters integrate data to the available destinations that are installed on or connected to your computer. The Integration Manager engine performs high-speed integrations to destination applications or databases, such as Microsoft Dynamics GP.

Integration Manager can support many destinations. See Chapter 19, "Adapters and Destination Mappings," for more information about the specific destination adapters that are available in Integration Manager.

Adding the destination

From the Add Destination window, you can add a destination to an integration. When you add a destination to an integration, you are attaching a description of the destination's data structure and content to the integration. That description is known as the destination document definition.

You can only add one destination to each integration.

To add the destination:

1. Open the integration to add the destination to.
2. From the Integration Manager toolbar, choose **Add Dest.**

*You also can right-click on the Destination item in the Integration window, and choose **Add Destination**.*

The Add Destination window opens.

3. From the **Adapters** pane, select the destination adapter to use.

   The **Adapters** pane lists all destination adapters that are available to you. The destination document definitions for each adapter appear in the **Destinations** pane as you select the different adapters. Most destination adapters include predefined destination document definitions, such as General Journal or Payables Transaction.

4. From the **Destinations** pane, select a destination and click **Open**.

   For some adapters, a Destination Settings window opens, where you specify how the integration connects to the destination application or database. Refer to *Specifying destination settings* on page 64 for more information.

If you use an existing destination, be cautious about changing its properties because it might be attached to other integrations. If you change the properties of a destination in one integration, you are changing the properties of the destination in all integrations in which it is being used.

5. Choose **File > Save**.

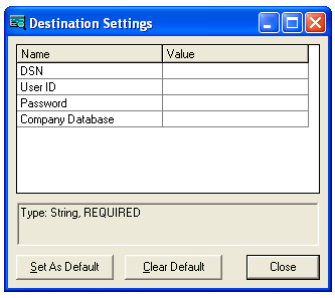
Specifying destination settings

After you add a destination to the integration, the Destination Settings window opens where you can specify how Integration Manager will connect to your Microsoft Dynamics GP database.

To specify destination settings:

1. Open the Destination Settings window.
   (**Integration** > **Destination Settings**)

You must add a destination to the integration before you'll be able to open this window. Refer to Adding the destination on page 63 for more information.



The Destination Settings window for each of the destination adapters looks slightly different, and the settings that you'll define will differ, depending on which adapter you're using.

2. Specify the destination settings by clicking in each field and entering the appropriate information. After you enter the information, you can click **Set As Default** to save these settings. If you choose to do this, the settings you entered are saved and used in each session. To clear these settings, click **Clear Default**.

If you need to run an integration for several companies, you can change the destination settings each time you run that integration, or you can create a separate integration for each company.

3. When you are finished specifying destination settings, click **Close**.

   When you close this window, the destination settings are saved with the integration. To display these settings again, choose **Integration** > **Destination Settings**.

Viewing the destination properties

After you have selected a destination, you can view its properties using the Properties window. Destination properties determine whether an object or field is insert only or if it accepts updates; the data type, such as boolean or string; and the number of characters an item can contain. The properties might vary, depending on which destination you're viewing properties for.

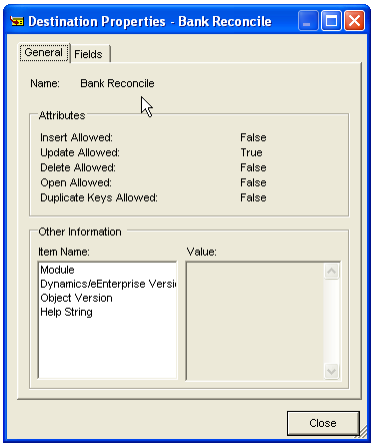
To view the destination properties:

1. From the **Integration** window, select a destination (such as Accounts).

2. From the Integration Manager toolbar, choose **Properties**.

You also can double-click the name of the destination in the Integration window to open the Destination Properties window.

   The Destination Properties window opens.



3. Choose the **General** tab to view the following information.

   Name Indicates the name of the destination.

   Attributes Indicate how information can be imported into the object.

   Each attribute has a **True** or **False** value. The actions listed in the table are allowed when the value for each attribute is **True**.

Attribute	Description
Insert Allowed	New records can be added to this destination.
Open Allowed	Existing records can be opened.
Update Allowed	Existing records can be updated by an integration.
Delete Allowed	Existing records can be deleted.
Duplicate Keys Allowed	Duplicate records can be created in the destination.

*Pay particular attention to the **Update Allowed** attribute. If this value is **True**, you can update existing records for this destination type. If the value is **False**, you can create new records but can't update existing ones.*

   For the **Insert Allowed** attribute, update functionality is limited to adding new child records. For example, you can add a new record, but you cannot update fields in an existing record.

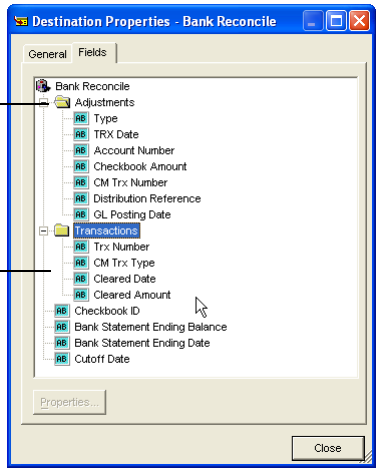
   Other Information Provides version information for the destination. To view this information, select an item in the list.

   See Chapter 19, "Adapters and Destination Mappings," for destination information specific to each adapter.

4. Choose the **Fields** tab to display a list of the fields that are included in the destination.

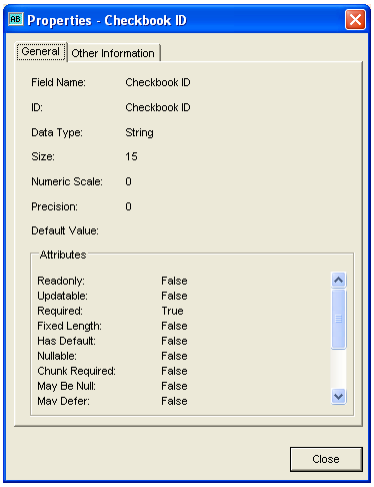
The folder icon represents a recordset of fields.

*To view the characteristics of a field, select it from the list and click **Properties.***



5. Select a field in the field list and click **Properties** to view the properties for the fields in the destination.

The Properties window for the field opens.



Be sure to note the fields that are required. When you create the mapping for your integration, you need to supply a value for each required field.

Field Name The name of the field.

ID The internal ID of the field in the supported adapters.

Data Type The type of data the field contains. The valid types are boolean, currency, date, enumeration, integer, long integer, numeric, string, and time. Which data types are valid depends on the type of data.

Size The number of bytes required to store the field. For string fields, it is the number of characters that can be stored by the field.

Numeric Scale The total number of digits in the numeric field.

Precision The number of digits after the decimal point in numeric fields.

Attributes Indicate how information can be imported into the field. The following is a list of the common attributes that apply to destination fields.

Attributes	Description
Required	Indicates that a value must be supplied for the field.
Readonly	Indicates that data can't be imported into the field.
Updatable	Indicates whether the existing value in the field can be updated.
Fixed Length	Indicates that the value in the field is a fixed length.
Has Default	Indicates that the field has a non-blank default value available.

6. When you are finished viewing destination properties, close the window.

Viewing enumeration items

For enumeration fields, you also can view the items in the field and their corresponding values. When you import integer values into the field, those values should correspond to one of the enumeration items.

To view enumeration items:

1. From the Destination Properties window, choose the **Fields** tab.
2. Select the field to view the enumeration information for.
3. Click **Properties**.
4. In the properties window for the integration, choose the **Other Information** tab.

   Enumeration item properties, if available, are displayed on the **Other Information** tab.

You will learn more about enumerations when you create a mapping for your integration.

Removing a destination from an integration

Use the following procedure to remove a destination from an integration.

To remove a destination from an integration:

1. In the Integration window, select the destination to remove.
2. Right-click the destination and choose **Remove <name of destination>**.

You also can select the desalination and press the DELETE key to remove the destination.

Chapter 9: Creating mappings

After you've added a destination, you must map the source data to your destination.

Step	Description
Step 1	Create the integration
Step 2	Add a source
Step 3	Create query relationships (if necessary)
Step 4	Add a destination
Step 5	**Create destination mappings**
Step 6	Save the integration
Step 7	Run the integration
Step 8	Examine the integration results

After you build an integration, you need to create a destination mapping. A destination mapping describes where each field in the integration's destination will come from. Values for many fields in the destination will come from the sources used for the integration. The values of other fields can be set using default values or constant values.

This part of the documentation includes the following information.

Creating a destination mapping

The mapping describes where each field in the destination will get its data. Use the Integration Mapping window to create a mapping. From this window, you can select rules, specify the source, or set rule properties—all of which define where the information for an item in the destination originates.

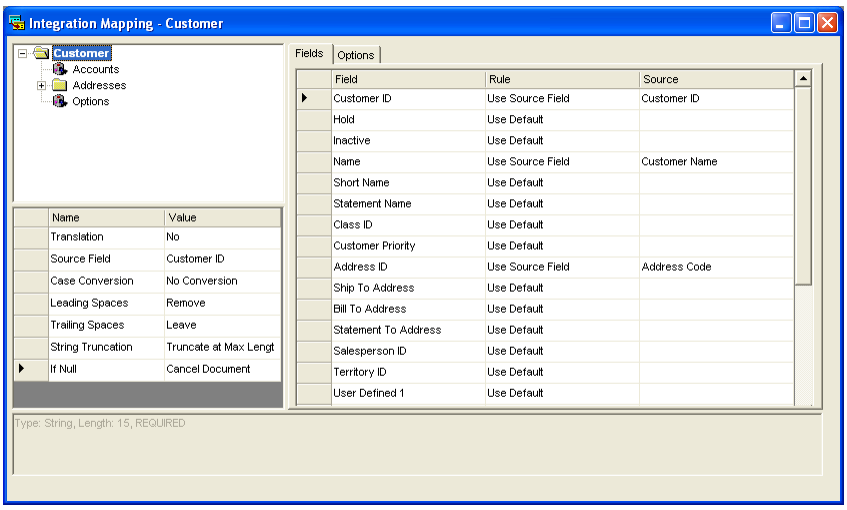
When you create the mapping, you will need to set up a rule for each required field in the root recordset of the destination. You also can set up mapping options. Required fields and mapping options for each adapter are described in Chapter 19, "Adapters and Destination Mappings."

Use the following procedure to map your source data to the destination.

To create a destination mapping:

1. Open the integration and be sure you have added the source or sources, as well as the destination. From the Integration Manager toolbar, choose **Mapping**.

   The Integration Mapping window opens.



   The upper left pane of the Integration Mapping window shows the destination in terms of recordsets. There are three types of recordsets in an Integration Manager destination: root recordsets, one-to-many child recordsets, and one-to-one child recordsets.

   Root recordset The root recordset is the top level of the destination. For example, the root recordset of the Customer destination is called Customer and it contains the **Customer ID** and **Customer Name** fields (plus a number of additional fields). Each destination has only one root recordset.

   One-to-many child recordset A one-to-many child recordset is represented by a folder icon and may contain more than one record for each record in its parent recordset. For example, in the Microsoft Dynamics GP Customer destination, the Addresses recordset is a child of the root recordset. Each customer can have many addresses. Therefore, the Addresses recordset has a one-to-many relationship with its parent.

   One-to-one child recordset A one-to-one child recordset is represented by a cylinder icon. This recordset contains only one record for each record in its parent. For example, in the Microsoft Dynamics GP Customer destination, the Internet Addresses recordset is a child of the Addresses recordset. Although there can be many addresses for each customer, there can only be one Internet address for each address record. Therefore, the Internet Addresses recordset has a one-to-one relationship with its parent recordset (Addresses).

   You need to select a rule for each of the required fields in the root recordset for the selected destination. Required fields are those fields in a destination recordset that need to be mapped for the integration to run successfully. Some of the child recordsets also contain required fields. You do not need to map the required fields for child recordsets if you do not map any of the other fields in that recordset.

For example, in the Customer destination, Customer ID is a required field for the root recordset (Customer). You need to select a mapping rule for the Customer ID field. In addition, if you select mapping rules for any of the fields in the Addresses recordset, you will need to select a mapping rule for the Address ID field, since it is a required field for the Addresses recordset.

The mapping tables in Chapter 19, "Adapters and Destination Mappings," contain lists of required fields by recordset for each destination. You also can view a field's attributes in the bottom pane of the Integration Mapping window to determine which fields are required.

2. From the **Fields** tab, set mapping rules for individual fields.

   A destination mapping is composed of *rules*. A rule defines where the information for an item in the destination originates. There is one rule for each field in the destination. To specify the type of rule to use for a field, choose the **Rule** column and select a value from the list.

   As you click each row, information about that field is displayed at the bottom of the Integration Mapping window. This information includes the type of field (such as String, Numeric, and so on), the field length, and if the field is a required field.

*You also can view the properties for a field in the Integration Mapping window. Right-click a field and choose **Field Properties** to view the properties, which helps you determine the data type, field size, and whether the field is required.*

Integration Manager has the following rules.

Rule	Description
Use Source Field	The information for the field originates from one of the sources that is part of the integration. A Lookup button appears in the **Source** column, allowing you to open the Source Object window and select a field.
Use Constant	You can type a value into the **Source** column. This value is used for every record created or updated when you run the integration.
Use System Date	For date fields, the field in the destination is set to the system date.
Use Positive Source Field	For debit fields, positive values are imported as they are. Negative values are imported as zero.
Use Negative Source Field	For credit fields, negative values are imported as the corresponding positive value. Positive values are imported as zero.
Use Default	For newly inserted documents, the default value for that field is used, as determined by the business logic of the destination application. When updating existing records, the default value is typically the value that already exists for the field.
Blank	For some string fields, the field in the destination is left blank when you run the integration.
Use Script	You'll attach a Visual Basic script that runs to provide the value for the field. Refer to Chapter 20, "Using scripts."

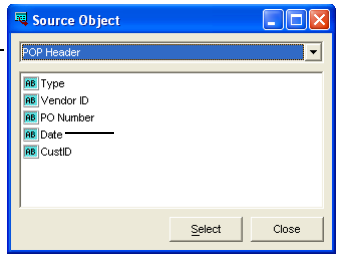
Rule	Description
Use Input	Solicits input from a user when the integration runs. When you use this rule, you can define the prompt in the **Rule Properties** section of the Integration Mapping window. If you do not define the prompt, the default prompt for a string field is, "Please enter a value for <name of field>." If the field uses an enumerated data type, the default prompt is, "Select a value for <name of field>", where you can select a value from a list. For fields that use dates, the prompt will include a Calendar view from which you can select a date.

*The **Use Positive Source Field** and **Use Negative Source Field** rules allow you to use the same source field for both the debit and credit value of a transaction.*

3. Click the lookup button in the **Source** column to select a value for the source. The Source Object window opens.

Select the appropriate source from this list.

*Select an item from the source and click **Select**.*



4. Choose the name of the source that contains the information to be used for the field. Choose the name of the field and click **Select**.

*If you chose the **Use Source Field** rule, you must specify a source for the field.*

The item you selected appears in the **Source** column of the Integration Mapping window.

*You also can drag items from the Source Object window to the **Source** column in the Integration Mapping window. The **Rule** column automatically is set to **Use Source Field** and the **Source** column displays the item you selected.*

5. For fields that are not using default values, set the appropriate rule properties.

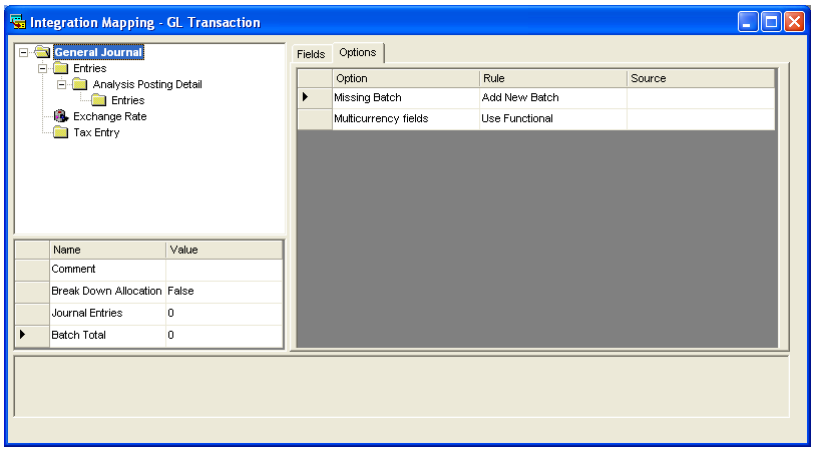
Some rules have additional properties that you specify in the **Rule Properties** section of the Integration Mapping window.

The following table describes the rule properties.

Category	Property	Description
General	Translation	Allows you to convert an item from a source based on the values in the translation list.
	Source Field	The column in a source from which data is read.
	Script Text	Indicates whether VBScript code is associated with the field.
	If Null	Indicates how Integration Manager handles a null (empty) value. A blank value can be used or the current record (document) can be canceled.
	Constant Value	The value to use for the item if the **Use Constant** rule is selected for the field.
String	Case Conversion	Indicates whether characters in a string value are converted to uppercase or lowercase characters.
	Leading Spaces	Indicates whether leading spaces are removed from a string value.
	Trailing Spaces	Indicates whether trailing spaces are removed from a string value.
	String Truncation	Indicates how Integration Manager handles string values whose lengths exceed the value allowed by the destination field. The string can be truncated at the maximum allowable length or the current record (document) can be canceled.
Numerics	Change Sign	Indicates whether to change the sign associated with the value. If set to **True**, all positive values become negative and all negative values become positive.
	Shift Decimal Point	Indicates how the decimal point is shifted for data read from the source. Positive values shift the decimal to the left. Negative values shift the decimal to the right. The value zero leaves the decimal position unchanged.
	Rounding	If the value has too many digits to fit into the destination, it must be rounded or truncated. Set this property to **Round** or **Truncate** to round or truncate the value. Set the property to **Round with Warning** or **Truncate with Warning** to round or truncate the value and generate a warning for the integration. Click **Cancel Document** to cancel the record (document) if rounding or truncation is required.

6. To set the mapping options, choose the destination in the upper-left list in the Integration Mapping window and choose the **Options** tab.

Some destinations have additional options that you specify on the **Options** tab. These options indicate how Integration Manager handles special circumstances when importing data into a specific destination. Additional mapping options are listed in Chapter 19, "Adapters and Destination Mappings."



When you select an option, the properties for that option appear in the **Rule Properties** section of the Integration Mapping window. Use these properties to specify individual options. For more information, refer to Part 4, **Adapter reference**.

The default rule for one-to-many child recordsets is **Use Default**. The first time you map any field in a one-to-many recordset by setting its mapping rule to **Use Source Field**, Integration Manager automatically changes the **Record Source** option to **Use Source Recordset**, and sets the **Source Recordset** to the appropriate value.

The **Record Source** option (which is found on the **Options** tab of the Integration Mapping window) specifies which recordsets in the source will be associated with each recordset in the destination. There are different **Record Source** options for the different types of recordsets. See Chapter 19, "Adapters and Destination Mappings," for more information.

*The root recordset does not have a **Record Source** option. It is always associated with the root recordset of the source. For each record in the root recordset of the source, one record will be created in the destination's root recordset.*

One-to-many child recordsets may have the following options.

Option	Description
Use Source Recordset	Creates one recordset in the destination for each record in the associated source recordset.
Use Default	For newly inserted documents, the default values for the recordset are used, as determined by the business logic of the destination application. When updating existing documents, the default values typically consist of the records that already exist in that recordset.
Empty	No records are integrated to this recordset. If the destination application contains business logic that typically would populate this recordset with default data, the default logic is superseded by the **Empty Record Source** option, and no records are created in this recordset for this destination.
Default Non-Imported	A default set of records is generated for this recordset, but some of them may be overwritten by records mapped to this recordset. For example, if you are integrating to Payables Transaction and you select the **Default Non Imported** rule for the Distributions recordset, a default set of distributions is created based on the business rules in the destination application. Typically this would be a debit to the vendor's default Purchases account and a credit to the default Accounts Payable account. Mapping a Purchases distribution in this situation overwrites the default Purchases distribution, but leaves the default Accounts Payable distribution intact.

7. To set the recordset options, choose the recordset in the upper-left pane in the Integration Mapping window and choose the **Options** tab.

   Recordsets have options that allow you to control how items in the recordset are mapped. (Descriptions for the different recordsets are at the beginning of this procedure.)

   One-to-one child recordsets may have the following options.

Option	Description
Use Field Rules	The mapping rules that appear on the **Fields** tab of the Integration Mapping window will be applied.
Use Default	This is the same as selecting **Use Default** as the mapping rule for all fields within that recordset.

8. When you are finished creating the destination mapping, choose **File** > **Save Integration** and close the window.

Field translations

A field translation allows you to define a relationship between values in the source file and corresponding values that are used for the destination field. For example, assume the source file for an integration contains numeric codes that represent specific salespeople. However, you might want to use names for the Salesperson ID field, rather than the numeric code. You can use a field translation to accomplish this.

Field translations can be local or global. You can create local translations in the Integration Mapping window to use in the current integration, or you can create global translations that can be used with any integration.

Local translations are created directly from the mapping window. A local translation is available only for the specific field and is used in a specific destination mapping. Refer to *Creating a destination mapping* on page 69 to create and use local translations in a destination mapping. Refer to *Creating global translations* on page 76 for more information about creating global translations.

Creating global translations

Global translations can be used in any destination mapping and are available for any integrations you create. You can use the Object Browser window to create a global translation.

To create a global translation:

1. From the Integration Manager toolbar, choose **Objects** to open the Object Browser window.

2. Click **Translations** in the types list, and click **New** to create a new translation.

3. Enter a name and description for the translation, and create the translation.

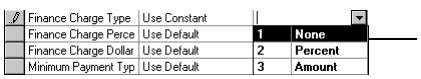
Understanding enumerations

An enumeration is a data type that is restricted to a fixed set of named values. When you set the value of an enumeration field, you supply the integer value or string that corresponds to one of the items in the enumeration.

To view a list of the items in the enumeration, view the properties for the field. The enumeration items are located on the **Other Information** tab. See *Viewing enumeration items* on page 67.

If you choose the **Use Source Field** rule to supply the value of an enumeration, the value from the source should be the integer or string corresponding to the appropriate item in the enumeration. Using the example above, the value of the field can be set to **Percent** either by mapping the value **2** or by mapping the value **Percent** from the source data. You can use a translation to map values from the source file to the appropriate integer value.

If you choose the **Use Constant** rule to supply the value of an enumeration, the items from the enumeration appear in a list in the **Source** column. Select the appropriate value from the list. The value you select will be used for all records that are integrated.



*If you choose the **Use Constant** rule, you can choose enumeration items from a list.*

Adding enumeration values to local translations

Use the Properties window for the translation to add enumeration values to local translations.

To add enumeration values to local translations:

1. In the Integration window, double-click **Destination Mapping**.

   The Integration Mapping window opens.

2. Select a field name that has a translation rule, such as **Finance Charge Type**.

3. Change the rule to **Use Source Field**.

4. Change the source, if necessary.

5. In the **Rule Properties** pane, find the **Translation** row and click the **Value** cell. Click **Yes**.

   The Properties window for the translation opens.

6. Choose the **Translation** tab and set values for each enumeration.

   You can set values using one of the following methods.

   - Set all documents that are to be integrated to use the same enumeration value. For example, if all the customer records you are integrating use percent finance charges, use the **Use Constant** rule.

   - Set all documents that are to be integrated to use the appropriate enumeration value, based on a value for each source document. You may need to translate the value stored in the source (for example, Finance Percentage) to a value that can be recognized by the destination (Percent 2).

7. Click **OK** to save your changes and close the Properties window for the translation.

8. When you are finished, close the Integration Mapping window.

Chapter 10: Running integrations

Now you are ready to run the integration.

Step	Description
Step 1	Create the integration
Step 2	Add a source
Step 3	Create query relationships (if necessary)
Step 4	Add a destination
Step 5	Create destination mappings
Step 6	Save the integration
Step 7	**Run the integration**
Step 8	Examine the integration results

Running the integration imports the source data into your selected destination.

This part of the documentation includes the following information.

Before running integrations

Before you run an integration, do the following:

Save the integration This ensures that if a problem is encountered while performing the integration, you will not lose the integration you have created.

Start Microsoft Dynamics GP If you're integrating data to Microsoft Dynamics GP, be sure that Microsoft Dynamics GP is running and that you have logged on to the appropriate company. Close all windows in Microsoft Dynamics GP.

Verify the integration properties To view the properties for the integration, choose **Integration** > **<Integration Name> Properties**. Verify that the destination edit mode is set appropriately. Also verify that the number of errors and warnings to encounter before the integration is automatically stopped is correct. Refer to *Setting integration properties* on page 79 for more information.

Verify the log settings Verify that the log level is set appropriately. You will learn more about the integration log later.

Setting integration properties

Before running your integration, check the properties that still need to be set. You need to specify a destination edit mode, indicate the maximum number of errors and warnings, attach scripts (if using them), and select a logging level for your integration.

To set integration properties:

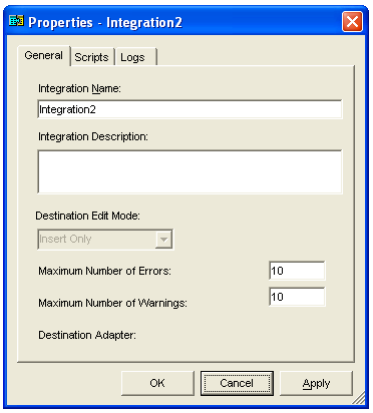
1. Start Integration Manager if it is not already open.

2. Choose **File** > **Open Integration**. Select the integration to run and click **Open**.

   The Integration window opens.

3. Choose **Integration** > **<integration name> Properties**.

   The Properties window opens.



4. Select a destination edit mode, which indicates how records are imported into the destination. You will not be able to specify the destination edit mode when you first create an integration. The destination edit mode only becomes available when you choose the destination for the integration. Some edit modes may not be available for some destinations. The default mode is **Insert Only**.

   The following table describes the available edit modes.

Mode	Description
Insert Only	Only new records can be created by the integration. Existing records can't be updated.
Update Only	Only existing records can be updated by the integration. New records can't be created.
Insert and Update	New records can be created and existing records can be updated by the integration.

5. Specify the maximum number of errors.

   When you run an integration, this value indicates the number of errors that can occur before the integration is automatically stopped. An error occurs when a document fails to be integrated, typically because of an invalid condition in the data. For example, if you try to integrate a Microsoft Dynamics GP receivables invoice where the total sales amount is a negative number, the document is not integrated, because negative invoice amounts are not allowed.

6. Specify the maximum number of warnings.

   When you run an integration, this value indicates the number of warnings that can occur before the integration is automatically stopped. A warning occurs when a document is integrated, but some information needs to be presented to

the user. For example, Microsoft Dynamics GP allows you to enter and save a general journal entry where the total debits do not match the total credits. However, a warning appears, explaining that the transaction can't be posted until the problem has been resolved. Using Integration Manager, the unbalanced journal entry can be integrated, but it results in a warning similar to the one presented by Microsoft Dynamics GP.

It is important to understand the difference between an error and a warning in Integration Manager. When an error occurs, typically because of an invalid condition in the data, the document fails to be integrated. When a warning occurs, the document is integrated, but Integration Manager provides information about the problem so you can resolve it.

7. On the **Scripts** tab, attach a **script** to the integration (optional).

   Use the **Scripts** properties to attach scripts to the integration. An integration can have several scripts attached that are executed at various points during the integration. Scripts are written in VBScript, a subset of the Microsoft Visual Basic programming language. For more information, refer to Chapter 20, "Using scripts."

8. On the **Logs** tab, set the log properties to specify how to view the log files created by Integration Manager. For more information about log properties, refer to *Specifying integration log storage types* on page 97.

9. Click **Apply** to apply the integration properties and click **OK** to close the window.

Running the integration

You can run an integration from within Integration Manager, within Microsoft Dynamics GP, or by using the **Start** menu.

To be sure that users have the correct permissions to run an integration, you must use the appropriate user privileges when you run the integration for the first time. Typically, this means being part of the Administrators group or the Power Users group on Windows XP. On Windows Vista, Windows 7, or Windows Server 2008, it means running the integration with administrative privileges. For example, you can run the integration from the shortcut on the ***Start*** *menu, select* ***Microsoft Dynamics > Integration Manager*** *and right-click* ***Run Integration****, and then select* ***Run as Administrator****. Refer to your operating system's documentation for more information. When you run as an administrator, the registry and other settings are updated so other users can run integrations without permission issues.*

Integration Manager

To run the integration from within Integration Manager, choose **Run** from the Integration Manager toolbar. You also can right-click an item in the Integration window and choose **Run Integration**.

Microsoft Dynamics GP

To run an integration directly from within Microsoft Dynamics GP, start tMicrosoft Dynamics GP and close all the windows. Choose **Tools** > **Integrate** > **Run Integration**. The Run Integration window opens. Use this window to select and run an existing integration.

Start menu

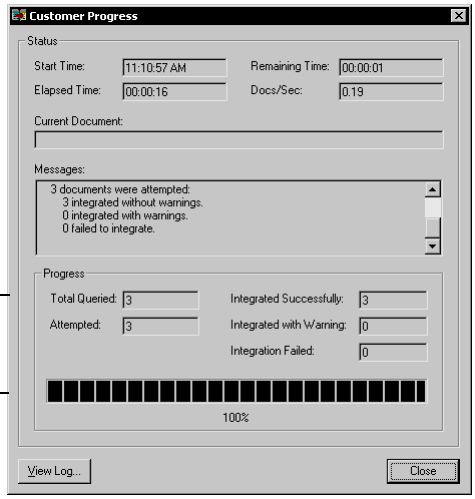
To run the integration from the **Start** menu, choose **Start** > **Programs** > **Microsoft Dynamics** > **Integration Manager** > **Run Integration**. The Run Integration window opens. Select the integration to run, and click **Run**.

Understanding the Progress window

After you have started to run an integration, Microsoft Dynamics GP, if it's open, is minimized and disappears from view. This is done to maximize the speed of the integration. The Progress window opens.

Errors and warnings generated by the integration are listed here.

Progress details are shown here.



As the integration is running, messages appear in the window that describe the integration status. Any errors or warnings are listed in this window. Details of the integration progress are shown at the bottom of the window.

Click **View Log** to view the log results. Refer to *Viewing and printing logs* on page 98 for more information.

Evaluating integration results

After the integration has finished, you can evaluate the integration results. The goal of the integration is for all items to be integrated without errors or warnings. Use the detailed progress information in the Progress window to view the integration status. The following table describes the fields that provide this information.

Field	Description
Total Queried	The number of rows returned from the query for the integration.
Attempted	The number of items Integration Manager attempted to integrate.
Integrated Successfully	The number of items that were integrated without any errors or warnings.
Integrated with Warning	The number of items that were integrated, but encountered a warning.
Integration Failed	The number of items that encountered an error and failed to be integrated.

Creating rejection files

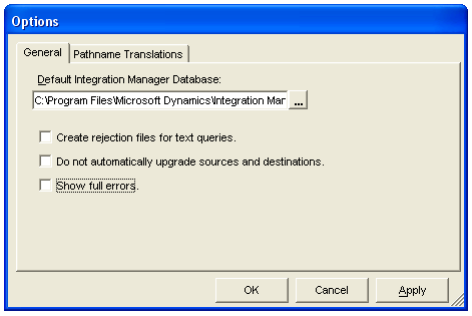
It might be useful to know which records were rejected by Integration Manager. If you are using text queries for your integration, Integration Manager can write any rejected records to special text files called *rejection files*. You can edit the rejection files to fix any problems that prevented the records from being imported, then use them as source files to import the rejected records into Microsoft Dynamics GP.

To create rejection files:

1. From the **Tools** menu, choose **Options**.

2. In the **Options** window, choose **Create rejection files for text queries**.

   Any rejection files that are created are placed in the same location as the text file used for the query. They have the same name as the query, but have an .rjt extension.



   For example, assume the Vendors query retrieves information from C:\My Documents\VendInfo.txt file. If you select the **Create rejection files for text queries** option, any rejected records are written to the following file: C:\My Documents\Vendors.rjt.

Rejection files are not persistent between integrations. If you run an integration again, any existing rejection files are overwritten.

Chapter 11: Troubleshooting integrations

This part of the documentation contains information about troubleshooting integrations if they are not working properly.

This part of the documentation includes the following information.

Source problems

The following information includes common problems that occur with sources.

Missing data for field value

Situation I do not see a value for a certain field in the Data Viewer when I preview the source. When I run an integration, I sometimes get errors about null values.

Solution It is possible that null or empty values for fields exist in the source. If this is the case, the application is behaving as it should.

If you are certain that the source does not contain null values, the field may have an incorrect data type defined for it. If an incorrect data type is set for a field, the value of the field will appear to be null. For information about setting data types for columns, refer to Chapter 7, "Data types."

If the data type is correct, then data for the source field might not be valid. Date fields must be in a proper date format and numeric values must not exceed the range allowed by the specific data type. For more information about proper date formats and numeric ranges, refer to Chapter 7, "Data types."

ODBC errors

Situation When I preview a source query or run an integration, I receive ODBC errors.

Solution Try previewing the individual source queries to determine which query is causing the error. If all source queries generate errors, be sure that the DSN used by the source query is set up correctly in the ODBC control panel. For other ODBC problems, refer to your ODBC driver documentation.

Missing or extra rows

Situation When I preview a source query, it returns more rows than it should, or it doesn't return enough rows.

Solution Check the row restrictions for the source query in the Properties window for the source. If the row restriction is not set up properly, it may be too restrictive and doesn't return enough rows, or it may not be restrictive enough and returns too many rows.

If you are using multiple source queries for the integration, be sure the relationships between the sources are set up correctly. If the relationship type is wrong, the child source query could return an incorrect number of rows.

Be sure there aren't any blank rows in the source files. Blank rows typically appear at the bottom when you preview the source.

Missing or extra fields

Situation When I preview my source, a field is missing, or my source has more fields than I need.

Solution This usually happens when the **Show** property for the field is incorrect. In the Source Query Properties window, choose the **Columns** tab and verify that the **Show** property is set. If you don't want to display the field in the Data Viewer, or use the field in the Destination Mapping, then clear the **Show** check box. Otherwise, be sure it's selected.

If you are missing fields, that may be a result of the grouping set for the source query. When a grouping is set for a source query, the source returns only the columns that are a part of the grouping.

Mapping problems

The following information includes common problems that occur with mappings.

"Field cannot be null" error

Situation When I run an integration, I get an error that a destination field can't be null.

Solution The default rule for destination fields is **If Null Cancel Document**. If this is not the desired behavior, change the destination field rule to **If Null Use Blank** or **If Null Use Default**. To change this rule, open the Destination Mapping window and select the field that is causing the error. In the lower left corner of the mapping window, change the **Rule Property** selection from **If Null Cancel Document** to the appropriate value.

Integration problems

The following information includes common problems that can occur when you run an integration.

Child recordsets not integrating

Situation When I run an integration, the data for a child recordset in the destination is not integrating.

Solution Be sure that the **Record Source** is set to the proper source. In the Destination Mapping window, select the child recordset that is not integrating. Choose the **Options** tab. The **Record Source** option should be set to a valid source query. If it is not, click in the **Source** column and select the proper source query or recordset from the list.

Warnings cause an integration to fail

Situation When I run an integration, no errors occur but the integration fails after 10 warnings.

Solution New integrations have a default setting that causes them to fail after 10 errors or 10 warnings. To change this, open the Integration Properties window and set the number of warnings to a higher number.

Slow integrations

Situation My integration runs slowly.

Solution There are several changes you can make to your integration to increase the performance of the integration.

- If multiple sources are used in an integration, the integration will run faster if you set the sorting options of the source query in Integration Manager. You can set the sorting options in the Source Query Properties window by choosing the **Sorting** tab, and adding columns to the **Order By** list.

  The most appropriate columns to sort by for performance reasons would be those involved in relationships with other source queries in the integration. For example, if a relationship between two sources is based on the DocNumber column, sorting both source queries by that column will increase performance.

  This only works if the sorting options are specified in Integration Manager. The performance of the integration won't increase if the data is sorted by other means, such as specifying an "order by" clause in the SQL text for Advanced ODBC or using a view or stored procedure that sorts the retrieved data.

- Be sure that you have mapped the minimum number of fields required to integrate your data. If a field contains no values, or the value it contains is the default value used in the destination, remove the mapping for that field.

- If you use VBScript in the integration to open an ActiveX™ Data Objects (ADO) connection, be sure the connection is opened only once during the integration. If an ADO connection is opened and closed for every document, the performance of the integration may suffer. It is possible to open an ADO connection once and reuse that connection across script events and documents. You can accomplish this by using the VBScript method SetVariable to store the connection and GetVariable to retrieve it. Be sure to close the connection by using the VBScript method objConn.Close when you are finished.

Script error "Expected Statement"

Situation I have a script in an integration that uses the "Execute" syntax. When I run the integration, I get a script error with the message "Expected Statement."

Solution This is a known issue with VBScript version 5 and Integration Manager. The fix is to fully qualify the Execute method by adding "Integration." before the Execute statement.

Instead of this:

```
Call Execute("C:\Windows\System32\Notepad.exe")
```

Use this:

```
Call Integration.Execute("C:\Windows\System32\Notepad.exe")
```

Errors from Microsoft Dynamics GP

The following list includes common problems that can occur with integrations to Microsoft Dynamics GP.

Incorrect data was integrated

Situation When I run an integration to Microsoft Dynamics GP, no errors are reported but the data that was integrated is not correct.

Solution Common causes for this problem include the following:

- A source field was mapped to the wrong destination field. Check the field mappings in the Destination Mapping window.
- Rule options for a field were set to round or truncate source data. Check the rule properties for this field in the Destination Mapping window.
- Rule options for the field were set to shift the decimal position of the field value. Check the rule properties for this field in the Destination Mapping window.
- A translation was set up incorrectly for a field. Check the translation by selecting the **Destination** field in the Field Mapping window and opening the Translation window.
- The wrong enumeration value was used for a field.

Part 3: Managing integrations

This part of the documentation provides information about managing integrations with Integration Manager.

This part of the documentation includes the following information.

- Chapter 12, "Modifying integrations," explains how to use integrations from other databases.
- Chapter 13, "Pathname translations," describes how to create pathname translations.
- Chapter 14, "Managing logs," explains how to manage the logs that contain information about the results of an integration.
- Chapter 15, "Compacting the Access database," explains how to compact the Microsoft Access database to reclaim space in the database file.
- Chapter 16, "Using integration groups," describes how to create integration groups, which enable you to run several integrations in succession.
- Chapter 17, "Running integrations from the command line," explains how to run integrations from the command line.
- Chapter 18, "Using advanced ODBC source queries," explains how to use a SQL statement to create advanced ODBC queries.

Chapter 12: Modifying integrations

Integration Manager allows you to use integrations from other databases. You can import and export integrations from within Integration Manager. You also can modify the various components that you use to create integrations.

This part of the documentation includes the following information.

Importing integrations

You can import integrations from another database to the current Integration Manager database. Choose **Tools > Options** to determine the current database.

Importing an integration does not remove it from a database. Instead, a copy of the integration is imported to the current database. To remove the integration from the other database, change the Integration Manager database path to point to the other database, and use the Object Browser window to delete the integration.

To import integrations:

1. From the **File** menu, choose **Import Integrations**.

2. From the Open window, select the database to import an integration or integrations from, and click **Open**.

   The Import Integrations window opens.

3. Select the integrations to import and click **Import**.

   As you click integrations to select them, you can press the SHIFT key to select a range of integrations and the CTRL key to select integrations that are not listed next to each other.

   If an integration with the same name already exists in the current database, Integration Manager prompts you to rename the integration you are importing. Integration Manager also prompts you to change the name of the integration's components if components with the same name already exist in the current database.

Exporting integrations

You can export integrations from the current Integration Manager database to another database. Choose **Tools > Options** to determine the current database.

Exporting an integration does not remove it from the current database. Instead, a copy is exported to the other database. To remove the integration from the current database, use the Object Browser window.

To export integrations:

1. From the **File** menu, choose **Export Integrations**.

   The Export Integrations window opens. If the current Integration Manager database contains any integrations that can't be exported, a message appears listing those integrations.

2. Select the integrations to export and click **Export**.

   As you click integrations to select them, you can press the SHIFT key to select a range of integrations and the CTRL key to select integrations that are not listed next to each other.

3. From the Open window, select the database to export the selected integration or integrations to and click **Open**.

Modifying components used in integrations

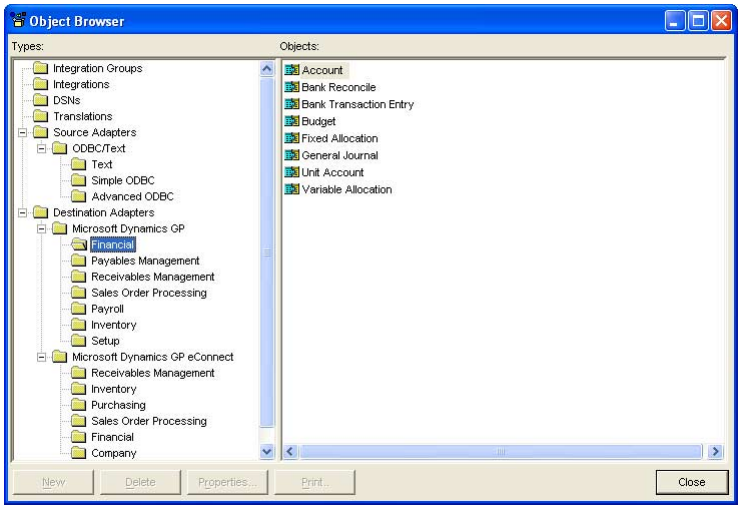
The Object Browser window provides a single location where you can modify the various components used to create an integration.

You can view, edit, print, and delete the following components using the Object Browser window.

- Integration groups
- Integrations
- Data sources
- Translations
- Sources
- Destinations

To modify components used in integrations:

1. From the **Tools** menu, choose **Object Browser**, or choose **Objects** from the Integration Manager toolbar. The Object Browser window opens.



2. To view items, select the item type from the **Types** pane. The **Objects** pane lists the items associated with the selected item type.

3. To create a new item, delete an existing item, view the properties of an item, or print a report for an item, use the buttons at the bottom of the window.

The buttons are active only when that function is available for the selected type.

4. When you are finished, close the Object Browser window.

Chapter 13: Pathname translations

A pathname translation allows you to define substitutions for the pathnames that are defined in the Integration Manager database file. For example, assume that several people access Integration Manager from different workstations. Each user might store the Integration Manager source files to a different drive. You can use a pathname translation to locate the source files.

This part of the documentation includes the following information.

Pathname translation overview

When you create sources, you use pathnames to specify the locations of the source files used by those sources. This pathname information is stored in the Integration Manager database file. If several people access Integration Manager from different workstations, or if you will be distributing the Integration Manager database file to multiple users, the pathnames may not be consistent.

For example, one workstation may use the drive letter H to refer to the network location where the source files are stored. Another workstation may use the drive letter R to refer to the same network location. If you tried to use the same Integration Manager database file on both workstations, one of them would not be able locate the source files.

To solve this problem, Integration Manager has implemented *pathname translations*. Using pathname translations, you can define substitutions that will be used in any pathnames defined in the Integration Manager database file.

Pathname translations are stored on a per user basis, not per workstation. If you create a pathname translation on a computer, and log in as someone else, that translation won't be available.

In the previous example, all the paths for the queries on the first workstation use the drive letter H. If you moved the Integration Manager database file to the second workstation, you would need to create a pathname translation on that workstation that translates from the drive letter H to the drive letter R. Any path that used the drive letter H would then have the drive letter R substituted.

You do not need to use pathname translations if the source files are stored on the server and everyone on the network uses the same drive letter for the server.

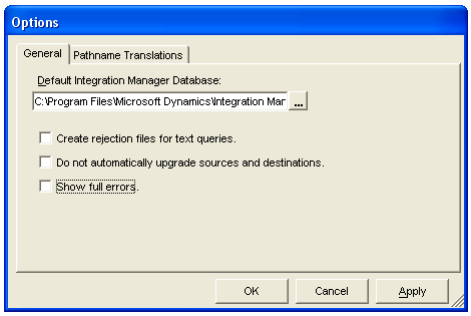
Creating pathname translations

Create pathname translations to allow multiple users to use different pathnames for the Integration Manager database on their workstations.

To create a pathname translation:

1. From the **Tools** menu, choose **Options.**

   The Options window opens.



2. Choose the **Pathname Translations** tab.

3. On each line, enter the **From** and **To** values for the translation. Click **OK** to save the changes.

You are not limited to mapping only drive letters. You can map partial paths, as well. For instance, you could map the path C:\Program Files\Microsoft Dynamics\Integration Manager\ to the path D:\IntegrationManager\.

Chapter 14: Managing logs

Integration Manager can create a log file that contains detailed information about the results of an integration. The Integration Log shows a list of logs; one log for each time you run the integration.

This part of the documentation includes the following information.

Specifying integration log storage types

Logs can be stored in a text file or in the same Access database file that stores integrations. In most cases, you will want to store the log in the Access database file so that any user can view the results of an integration.

If multiple users are using the same Integration Manager Database and these users are saving log information to a database, ODBC and primary key errors may occur. To prevent this from happening, each user should store log information to a text file.

If you choose to store the log in a text file, you must specify the folder that contains the log. By default, text file logs are stored in the Logs folder in the location where you installed Integration Manager.

To specify the integration log storage type:

1. From the Integration window, select the integration and click **Properties**.
2. In the **Properties** window, choose the **Logs** tab.
3. Set the storage type to **Database** or **File**.

Specifying the integration log level of detail

You can specify the level of detail to include in the log. The following table describes the detail levels.

Level	Description
Summary	Basic information about the integration is logged, including starting and ending times, completion status, and record counts. Information about individual records is logged if a warning or error is encountered.
Document	In addition to summary information, information about every integrated record is logged. This can produce a large log.
Trace	Detailed information is logged for all aspects of the integration. This is intended primarily as a debugging tool. Because of the large quantity of information logged, it is best to store this type of log in a text file.
None	No information is logged for the integration.

To specify the integration log level of detail:

1. From the Integration window, select the integration and click **Properties**.
2. In the **Properties** window, choose the **Logs** tab.
3. Change the level to the desired level of detail for this integration.

Viewing and printing logs

Use the Integration Log Viewer window to view and print detailed information about the integration. This information is especially helpful when troubleshooting the integration.

The Integration Log Viewer window includes information such as:

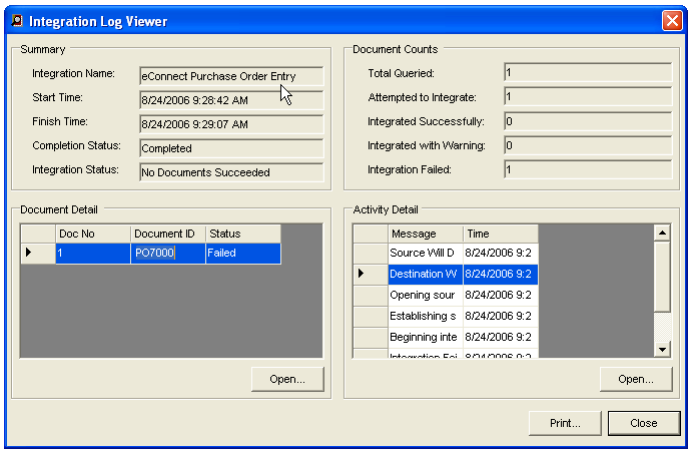
- Integration name
- Start and finish date/time
- Completion status: completed, incomplete, or canceled
- Integration status: all documents succeeded, no documents succeeded, partial success, canceled, or failure
- Number of source documents processed, successfully integrated, integrated with warnings, and number of failed documents
- Document details and activity details

To view and print logs:

1. From the Integration window, select the integration and click **Properties.**
2. In the **Properties** window, choose the **Logs** tab.

*You also can choose **View Log** in the Progress window of the integration.*

3. Select a log for a specific integration and click **Open**. You also can double-click the log item to view the details for that integration.



4. To print a report showing the log information, click **Print.**

5. To view more information about any of the items, select any of the items in the **Document Detail** or **Activity Detail** lists and click **Open**.

*If the information in the **Message** column in the **Activity Detail** section is too lengthy to view, select the message and click **Open**.*

Deleting logs

When you store integration logs in a database file, that file can become quite large. You can delete logs whenever you need to. When you delete a log, you can choose whether to delete only detail information or both summary information and detail information.

To remove a log:

1. From the Integration window, select the integration and click **Properties.**

2. Choose the **Logs** tab.

3. Select a log and click **Purge**.

4. To remove this log, click **Yes**.

5. To remove all log information, click **Yes**. To remove only the detail information, click **No**.

Chapter 15: Compacting the Access database

When you remove information from your integrations, that process removes information from the Access database file, but does not reclaim the space used by the file.

If you have Microsoft Access installed, you can use it to compact the your Integration Manager database, such as IntegrationManager.mdb. Refer to the Access documentation for information about how to do this.

This part of the documentation includes the following information.

Compacting an Access database

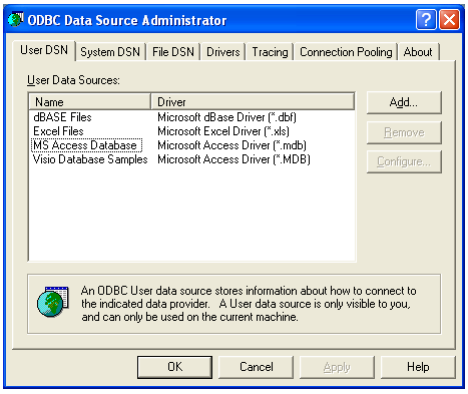
If you do not have Access installed on your system, use the following procedure to compact a file.

To compact an Access database:

1. Choose **Start > Control Panel > Administrative Tools**.

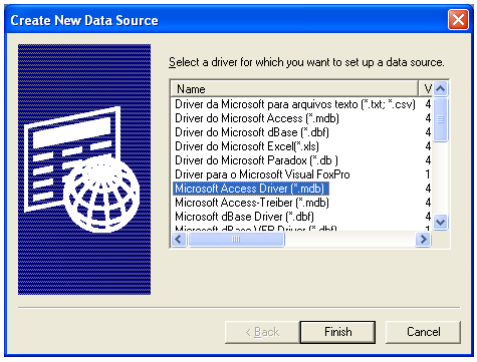
2. Choose **Data Sources (ODBC)**.

   The ODBC Data Source Administrator window opens.

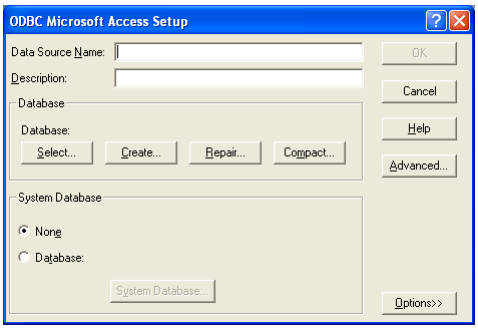


3. Choose **MS Access Database** and click **Add**.

The Create New Data Source window opens.



4. Choose **Microsoft Access Driver (*.mdb)** and click **Finish**. The ODBC Microsoft Access Setup window opens.



5. Click **Compact**.

   A window opens, asking you to locate the Access database file to compact.

6. Select the IntegrationManager.mdb file and click **OK**.

   The compacting process creates a new Access database file. Name this new file and click **OK**.

Using a compacted database

Use the following procedure to select and use a compacted database.

To use a compacted database:

To use the compacted file, do one of the following:

- Rename the compacted Access database file so it has the same name as the original database file. Then replace the original file.
- Set up Integration Manager to use the new file. To do this, start Integration Manager. From the **Tools** menu, choose **Options**. Choose the **General** tab and set the **Default Integration Manager Database** field to use the compacted Access database file.

*If you are running in a multiuser configuration and you change the **Default Integration Manager Database** setting, be sure to change the path for all workstations that use the file.*

Chapter 16: Using integration groups

In some cases, it is useful to perform several integrations in succession. For example, you might want to run several integrations overnight. Integration groups allow you to do this.

This part of the documentation includes the following information.

Creating integration groups

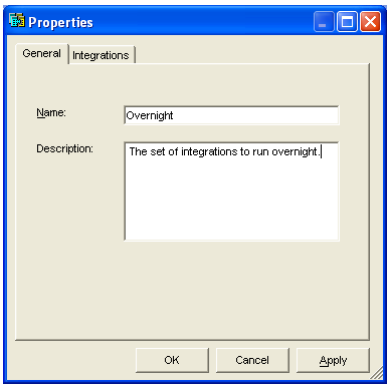
Use the following procedure to create an integration group.

To create an integration group:

1. In the Integration window, select **Objects** from the toolbar to open the Object Browser window.

2. Select **Integration Groups** as the type, and double-click **Define New Integration Group**.

   The Properties window for the integration group opens.



3. Enter the name for the integration group. You also may want to provide a description of the integration group.

   The integration group description should provide information about the group, such as which integrations it contains or when it should be run.

4. Choose the **Integrations** tab to specify which integrations will be part of the group. Click in each line and select an integration from the list that appears. The integrations run in the order you add them to the integrations list.

5. For each integration, specify how errors will be handled. If an integration in the group fails, you can choose to have the integration group stop or have the remaining integrations in the group run.

6. Click **OK** to save your changes and close the window.

Running an integration group

Use the following procedures to run an integration group.

To run an integration group from within Integration Manager:

1. Select **Objects** from the toolbar to open the Object Browser window.

2. Select **Integration Groups** as the type, select the group, and click **Properties**.

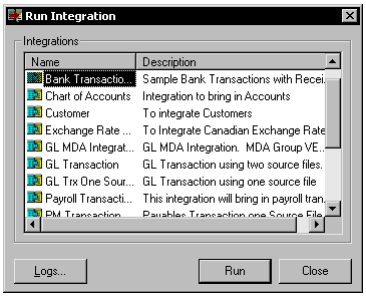
   The Properties window for the integration group opens.

3. Choose the **Integrations** tab, and click **Run**.

To run an integration group from within Microsoft Dynamics GP:

1. Choose **Tools** > **Integrate > Run Integration**.

2. In the Run Integration window, select any integration group and click **Run**.



Chapter 17: Running integrations from the command line

You can start integrations or integration groups directly from the command line. This is useful if you need to schedule integrations to start at predefined times.

Microsoft Dynamics GP must be running before you can start an integration from the command line. Typically, this means you also need to start Microsoft Dynamics GP from the command line. To do this, you first need to record a macro that performs the login process to Microsoft Dynamics GP.

This part of the documentation includes the following information.

Recording the login macro

The login macro logs in to Microsoft Dynamics GP the same way you would as a user. It types the user ID and password, and selects the appropriate company.

To record the login macro:

1. Start Microsoft Dynamics GP. Do not log in.

2. At the Welcome window, press ALT+F8 to begin recording the login macro. Name the macro and save it in a known location, such as the Microsoft Dynamics GP folder.

3. Log in to Microsoft Dynamics GP. Be sure to manually type any user ID and password information so this information is captured by the macro.

Storing the password in a macro could be a security issue.

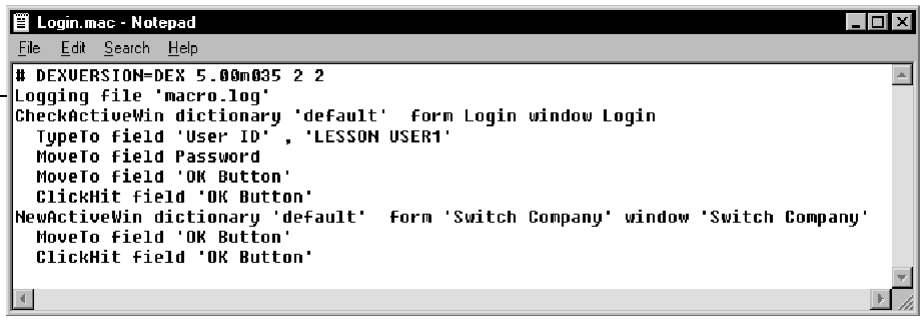
4. After you have logged in to the accounting system, stop recording the macro. To stop recording the macro, from the **Tools** menu, choose **Macro** and choose **Stop Record**. You also can press ALT+F8 to stop recording.

5. Edit the macro using a text editor such as Notepad to add the following line as the second line of the macro:

   ```
   Logging file 'macro.log'
   ```

   Adding this line prevents any message displayed by the macro, such as the total running time, from being displayed on the screen and preventing the login to Microsoft Dynamics GP. Instead, all messages generated by the macro are written to the MACRO.LOG file.

Add this line to the macro.



When you have finished editing the macro, save your changes.

Starting Microsoft Dynamics GP from the command line

To start Microsoft Dynamics GP from the command line, you must supply the location of the Microsoft Dynamics GP runtime engine, the launch file to use, and optionally, the macro to run when Microsoft Dynamics GP starts. For example, if the Dynamics.exe, Dynamics.set, and Login.Mac files are in the C:\Program Files\Microsoft Dynamics\Microsoft Dynamics GP folder, the following command starts Microsoft Dynamics GP and runs the LOGIN.MAC macro.

```
"C:\Program Files\MicrosoftDynamics\Microsoft Dynamics GP\Dynamics.exe"
"C:\Program Files\Microsoft Dynamics\Microsoft Dynamics GP\Dynamics.set"
"C:\Program Files\Microsoft Dynamics\Microsoft Dynamics GP\Login.mac"
```

Notice that a complete path is used to refer to the Microsoft Dynamics GP runtime engine and launch file, and the macro to be run.

Starting integrations from the command line

To start an integration or integration group from the command line, use IMRUN.EXE. The syntax is:

```
"C:\Program Files\Microsoft Dynamics\Integration Manager
11\Microsoft.Dynamics.GP.IntegrationManager.IMRun.exe" [/I Integration] [/G
Integration Group] [/S]
```

- /I *Integration*—Indicates the name of the integration to run. If the name contains spaces, enclose it in quotation marks.
- /G *Integration Group*—Indicates the name of the integration group to run. If the name contains spaces, enclose it in quotation marks.
- /S—Indicates that the integration or integration group is run without displaying the Progress window.

Note that you can run an integration or an integration group, but not both at the same time.

As an example, the following command runs the Vendor Information integration without displaying the progress window.

```
"C:\Program Files\Microsoft Dynamics\Integration Manager
11\Microsoft.Dynamics.GP.IntegrationManager.IMRun.exe" /I "Vendor Information" /S
```

Chapter 18: Using advanced ODBC source queries

Advanced ODBC source queries are those that use a SQL statement to retrieve information from an ODBC data source. We recommend that you have at least a basic understanding of SQL to use advanced queries for your integration.

This part of the documentation includes the following information.

Setting advanced ODBC query properties

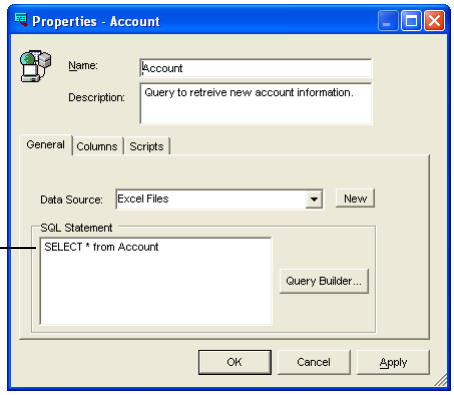
Each advanced ODBC source query has several properties, which are set using the Properties window for the advanced ODBC source.

To set advanced ODBC query properties:

1. Open the integration.

2. Open the Properties window for the Advanced ODBC source.
   Select a source and choose View > <name of source> Properties or
   Right-click on the source and choose <name of source> Properties or
   Add an advanced ODBC source to the integration

The SQL statement specifies what information is retrieved by the query.



3. Enter a name and description for the advanced ODBC source query.

   Name Each advanced ODBC source query must have a name. The name should describe the type of information retrieved by the query.

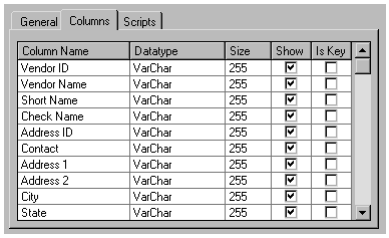
   Description The description should provide information about the advanced ODBC source query, such as what type of data is retrieved by the query or what the source of data is for the query. A query can be used by multiple integrations, so be sure to provide enough information so the query can easily be used by another Integration Manager user.

4. Enter or select a data source and SQL statement to use for the advanced ODBC query.

   Data Source This is the ODBC data source or data source created in Integration Manager from which you retrieve data. Refer to the documentation for the ODBC driver for more information about setting up ODBC data sources.

SQL Statement The SQL statement specifies what information is retrieved by the query. You can type in a SQL statement directly, or you can use the Query Builder in Integration Manager to create a SQL statement. See *Using the Query Builder window* on page 113.

5. Choose the **Columns** tab to display the columns returned by the query and change the datatype and size, as necessary.



Column Name Displays the names of the data items available for the query.

Datatype Indicates what type of data is contained in the column. The data type value is automatically retrieved from the ODBC data source. In most cases, the data type is appropriate. In other cases, you may need to change the data type to better reflect how the data is used for the integration. To change the data type, click in the **Datatype** column and choose an item from the drop-down list. Refer to Chapter 7, "Data types," for more information about selecting appropriate data types.

Size This value indicates the data size of each column, in bytes. In most cases, the default size is appropriate. In special cases, you may need to adjust the size to better reflect the size of the data value returned by the query.

6. Determine which columns should be included in the integration and which items in a column are unique identifiers.

Show Select the **Show** check box to include the data in this column in your integration. If you are using a large source file and you do not want all the data to be integrated, clear this check box for the columns to exclude.

Is Key Select the **Is Key** check box to indicate that the data items within a column are unique identifiers. For example, you might select a column called Customer ID as Is Key to indicate that the values within a column are unique.

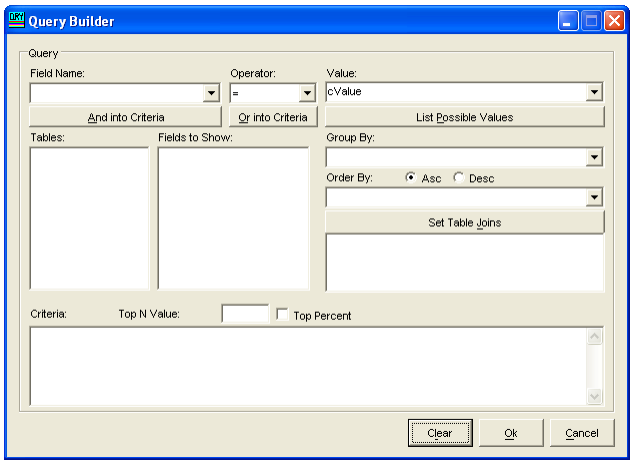
Integration Manager uses values marked as Is Key to identify specific rows that cause errors in the integration.

7. Choose the **Scripts** tab to attach scripts that are executed before and after the query is performed. Scripts are written in VBScript, a subset of the Microsoft Visual Basic programming language. For more information, refer to Chapter 20, "Using scripts."

Using the Query Builder window

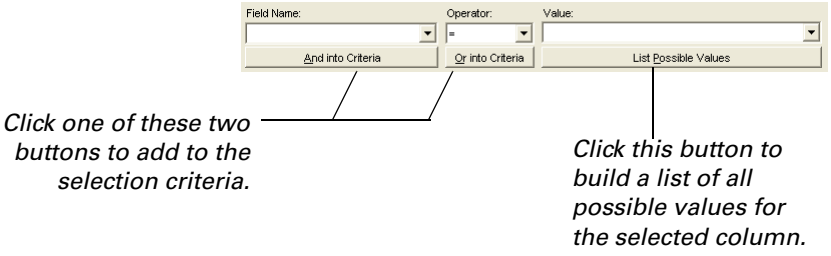
On the **General** tab of the Properties window for the advanced ODBC source, you can use the Query Builder window in Integration Manager to enter SQL information instead of typing in a SQL statement directly.

To view this window, open the properties window for the advanced ODBC source and from the **General** tab, choose **Query Builder**.



Selecting rows

Use the fields at the top of the window to specify which rows are included in the query results.

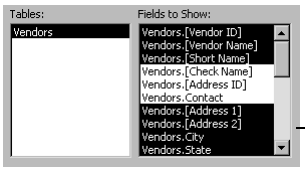


To select specific rows from the table, you create selection criteria based on fields from the tables available for the query. You can use logical AND and logical OR operators to apply several criteria.

The LIKE operator allows you to perform basic pattern matching with string columns. You can use the percent sign (%) as a wildcard character, representing any sequence of characters.

Selecting fields

You can select the fields that are included in the query results.



Select the fields in this list to include in the query results. If no fields are selected, all fields are included.

To specify the fields to include in the query results, select the appropriate tables in the **Tables** list. Then select the individual fields in the **Fields to Show** list. If you do not select any fields in the list, all fields for the selected tables are included.

Grouping

If you select an item in the **Group By** list, the data returned by the query is grouped based on that field. For example, if you group by the City field, the rows that have the same value for City appear as a group.

Select the field to sort. You may sort in ascending or descending order.

Sorting

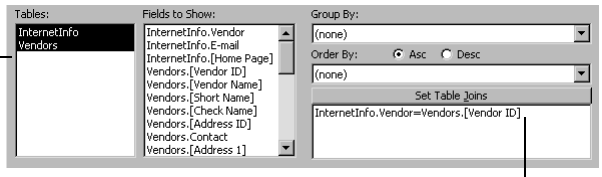
To specify how the rows returned by the query are sorted, select an item in the **Order By** list. The rows returned by the query are sorted based on the field you select. You can sort items in ascending or descending order.

If you are creating a query relationship for this query, you can improve the performance of the integration by sorting the query results by the field used for the relationship.

Joins

If the query returns data from more than one table, you must create joins between the tables to indicate how they are related. To create a join, select the two tables that are related and click **Set Table Joins**.

If the query returns data from multiple tables, you must create a join between them.



*To create a join between two tables, select them in the **Table** list and click **Set Table Joins**.*

In the Join Tables window, select the two tables to join. Select the field or fields in each table that are related. The table join is based on these fields. Click **Add Join to Query**.

Select the field or fields in each table that are related, then click ***Add Join to Query.***

Rather than using joins, you may find it easier to use multiple queries and create query relationships between them in Integration Manager.

Returning top values

To have the query only return a specified number of rows, supply a value indicating the number of rows to return. For example, the value 10 indicates that the first 10 rows that meet the query criteria are returned.

If you select the **Top Percent** check box, the specified percentage of rows is returned. For example, the following selections indicate that the first five percent of the rows that meet the query criteria are returned.

PART 4: ADAPTER REFERENCE

Part 4: Adapter reference

This part of the documentation describes the adapters that are available with Integration Manager and contains adapter specific destination information and mapping options.

This part of the documentation includes the following information.

- Chapter 19, "Adapters and Destination Mappings," describes the adapters that can be used in Integration Manager and contains destination mapping information for each destination adapter.

Chapter 19: Adapters and Destination Mappings

You can choose which adapters to use with Integration Manager. The information provided in this part of the documentation describes the adapters that you can use with Integration Manager.

This part of the documentation includes the following information.

How do I decide which adapter to use?

Use the following table to decide which adapter to use.

Adapter	Supported functionality	Remarks
Microsoft Dynamics GP destination adapter	Integrates data into Microsoft Dynamics GP destinations using Microsoft Dynamics GP.	This adapter includes the following predefined destinations: Financial, Payables Management, Receivables Management, Sales Order Processing, Payroll, Inventory, and Setup.
Microsoft Dynamics GP eConnect destination adapter	Integrates data into Microsoft Dynamics GP destinations using Microsoft Dynamics GP eConnect.	This adapter includes the following predefined destinations: Customer, Inventory Transaction, Inventory Item, Purchasing Order Entry, Receivings Transaction Entry, Sales Transaction, GL Account, GL Transaction, and Shipping Method.
XML source adapter	Integrates XML data into the Integration Manager destination adapters.	The XML source adapter will read an XML document and import data into the Integration Manager destination adapters.

Microsoft Dynamics GP destination mappings

This part of the documentation contains reference information you can use to set up destination mappings for your integrations that use destinations in the Microsoft Dynamics GP adapter. The tables that follow describe the properties and restrictions for each destination in the Microsoft Dynamics GP adapter. The tables also include required fields for each recordset in each destination. You need to select a rule for each of the required fields in the root recordset for the destinations you use. Root recordsets are in bold text.

*The tables also include the mapping options for each destination. These mapping options are on the **Options** tab of the Integration Mapping window. Because each child recordset contains the **Record Source** mapping option, it is not included in the following tables.*

Financial

The Financial module in the Microsoft Dynamics GP adapter contains destinations for Account, Bank Reconcile, Bank Transaction Entry, Budget, Fixed Allocation, General Journal, Unit Allocation, and Variable Allocation.

Account The Account destination has the following properties.

Destination Edit Modes	Insert Only Update Only Insert and Update
Mapping Options	None
Restrictions	The data in the Category field of the source file must match an existing category in Microsoft Dynamics GP (including capitalization and spelling).
Required Fields	
Account	Account Number Description Category

Bank Reconcile The Bank Reconcile destination has the following properties.

Destination Edit Modes	Update Only
Mapping Options	None
Required Fields	
Bank Reconcile	Checkbook ID Bank Statement Ending Date
Adjustments	TRX Date Account Number Checkbook Amount CM Trx Number GL Posting Date
Transactions	Cleared Date Cleared Amount

Bank Transaction Entry The Bank Transaction Entry destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Required Fields	
Bank Transaction Entry	Option Type Checkbook ID Number Trx Date Posting Date
Distributions	Account Number
Note: Posting journals aren't printed as part of an integration. You must print them after the integration has completed.	

Budget The Budget destination has the following properties.

Destination Edit Modes	Update Only
Mapping Options	
Password Required	
	Choose from the following rules to specify the action to take when a password is required. **Cancel Document** **Provide Password**—Enter the password in the **Rule Properties** pane.
Required Fields	
Budget	Account Budget ID Budget Year Period Name Amount

Fixed Allocation The Fixed Allocation destination has the following properties.

Destination Edit Modes	Insert Only Update Only Insert and Update
Mapping Options	None
Required Fields	
Fixed Allocation	Account Description
Distribution Accounts	Distribution Account Percentage

General Journal The General Journal destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	
Missing Batch	
	Choose from the following rules to specify the action to take when the batch is missing. **Cancel Document** **Add New Batch**—Set the following rule properties to specify characteristics of the new batch. **Comment**—Enter a text comment for the new batch. **Break Down Allocation**—Select True or False to indicate whether or not to print distribution accounts for each allocation account on the transaction edit list or posting journal. **Journal Entries**—Enter the number of journal entries for the batch. **Batch Totals**—Enter the actual currency amount in the batch.
Multicurrency fields	
	Choose from the following rules to specify the currency. **Use Functional** **Use Originating**
Override Rate Variance	
	This option is available in the Exchange Rate recordset. Choose from the following rules to specify the action to take if the exchange rate entered is greater than the rate variance. **Cancel Document** **Use Password**—Use the **Rule Properties** pane to enter the password needed to override the rate variance in the **Rule Properties** pane.
Override Exchange Rate	
	This option is available in the Exchange Rate recordset. Choose from the following rules to specify the action to take if an exchange rate is to be overriden by an integration. **Cancel Document** **Use Password**—Use the **Rule Properties** pane to enter the password needed to override the exchange rate.
Use New Exchange Rate	
	This option is available in the Exchange Rate recordset. Choose from the following rules to specify the action to take when a new exchange rate is to be added by an integration. **Cancel Document** **Use Password**—Use the **Rule Properties** pane to enter the password needed to add the exchange rate. **Use Exchange Rate**—Add the new exchange rate without requiring a password.
Required Fields	
General Journal	Journal Entry Intercompany Batch ID Transaction Type Transaction Date Source Document Reference Currency ID
Entries	Account Number
Entries\Analysis Posting Detail	Analysis Group ID
Entries\Analysis Posting Detail\Entries	Analysis Code ID
Exchange Rate	Rate Type ID Exchange Rate
Tax Entry	Account Amount Tax Detail

Unit Allocation The Unit Allocation destination has the following properties.

Destination Edit Modes	Insert Only Update Only Insert and Update
Mapping Options	None
Required Fields	
Unit Account	Account Description

Variable Allocation The Variable Allocation destination has the following properties.

Destination Edit Modes	Insert Only Update Only Insert and Update
Mapping Options	None
Required Fields	
Unit Account	Account Description
Unit Allocation\Distribution Accounts	Distribution Account
Unit Allocation\Distribution Accounts\Breakdown Accounts	Breakdown Account

Payables Management

The Payables Management module in the Microsoft Dynamics GP adapter contains destinations for Payables Transaction and Vendor.

Payables Transaction The Payables Transaction destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	
Missing Batch	
	Choose from the following rules to specify the action to take when the batch is missing. **Cancel Document** **Add New Batch**—Set the following rule properties to specify characteristics of the new batch. **Comment**—Enter a text comment for the new batch. **Transactions**—Enter the number of transactions for the batch. **Batch Total**—Enter the actual currency amount in the batch.
Over Invoice Limit	
	Choose from the following rules to specify the action to take when the invoice total exceeds the invoice limit. **Cancel Document** **Override**—Use the **Rule Properties** pane to enter the password needed to override the invoice limit.
Override Rate Variance	
	This option is available in the Exchange Rate recordset. Choose from the following rules to specify the action to take if the exchange rate entered is greater than the rate variance. **Cancel Document** **Use Password**—Use the **Rule Properties** pane to enter the password needed to override the rate variance.
Override Exchange Rate	
	This option is available in the Exchange Rate recordset. Choose from the following rules to specify the action to take if an exchange rate is to be overridden by an integration. **Cancel Document** **Use Password**—Use the **Rule Properties** pane to enter the password needed to override the exchange rate.
Use New Exchange Rate	
	This option is available in the Exchange Rate recordset. Choose from the following rules to specify the action to take when a new exchange rate is to be added by an integration. **Cancel Document** **Use Password**—Use the **Rule Properties** pane to enter the password needed to add the exchange rate. **Use Exchange Rate**—Add the new exchange rate without requiring a password.
Required Fields	
Payables Transaction	Voucher No Batch ID Doc. Date Vendor ID Currency ID Document Number
Distributions	Intercompany ID Distribution Account Distribution Type
Distributions\Analysis Posting Detail	Analysis Group ID
Distributions\Analysis Posting Detail\Entries	Analysis Code ID
Exchange Rate	Rate Type ID Exchange Rate
Intrastat	Item Number
Tax Details	Tax Detail ID Account Number

Vendor The Vendor destination has the following properties.

Destination Edit Modes	Insert Only Update Only Insert and Update
Mapping Options	
Remove Hold	
	Choose from the following rules to specify the action to take when a vendor hold exists. **Cancel Document** **Provide Password**—Use the **Rule Properties** pane to enter the password needed to remove a vendor hold.
Restrictions	The Vendor ID should be in all capital letters. You can set the Case Conversion rule property to Convert To Upper if the source file has the Vendor ID in lowercase letters.
Required Fields	
Vendor	Vendor ID Name
Accounts\Additional Vendor Accounts	Account
Addresses	Address ID
Withholding	Withholding Tax Rate

Receivables Management

The Receivables Management module in the Microsoft Dynamics GP adapter contains destinations for Cash Receipts, Customer, and Receivables Transaction.

Cash Receipts The Vendor destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	
Missing Batch	
	Choose from the following rules to specifies the action to take when the batch is missing. **Cancel Document** **Add New Batch**—Set the following rule properties to specify characteristics of the new batch. **Comment**—Enter a text comment for the new batch. **Transactions**—Enter the number of transactions for the batch. **Batch Totals**—Enter the actual currency amount in the batch.
Over Writeoff Limit	
	Choose from the following rules to specify the action to take when the writeoff amount exceeds the writeoff limit. **Cancel Document** **Override**—Use the **Rule Properties** pane to enter the password needed to override the writeoff limit.
Required Fields	
Cash Receipts	Receipt Batch ID Date Customer ID Currency ID Cash Receipt Type
Cash Apply	Apply—To Document Number Apply—To Document Type
Distribution	Distribution Account Distribution Type

Customer The Customer destination has the following properties.

Destination Edit Modes	Insert Only Update Only Insert and Update
Mapping Options	None
Restrictions	The Customer ID should be in all capital letters. The rule property for Case Conversion can be set to Convert to Upper if the source file has the Customer ID in lowercase letters.
Required Fields	
Customer	Customer ID
Address	Address ID

Receivables Transaction The Receivables Transaction destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	
Customer Hold	
	Choose from the following rules to specify the action to take when a customer hold exists. **Cancel Document** **Override Hold**—Use the **Rule Properties** pane to enter the password needed to override a customer hold.
Over Credit Limit	
	Choose from the following rules to specify the action to take when the credit amount exceeds the credit limit. **Cancel Document** **Override**—Use the **Rule Properties** pane to enter the password needed to override the credit limit.
Missing Batch	
	Choose from the following rules to specifies the action to take when the batch is missing. **Cancel Document** **Add New Batch**—Set the following rule properties to specify characteristics of the new batch. **Comment**—Enter a text comment for the new batch. **Transactions**—Enter the number of transactions for the batch. **Batch Totals**—Enter the actual currency amount in the batch.
Restrictions	This destination is insert–only. It cannot be used to update existing receivables transactions.
Required Fields	
Receivables Transaction	Document Type Document Number Batch ID Document Date Tax Date Customer ID Currency ID Commissions Applied To
Commissions	Salesperson ID Sales Territory ID
Distribution	Distribution Account Distribution Type
Distribution\Analysis Posting Detail	Analysis Group ID
Distribution\Analysis Posting Detail\Entries	Analysis Code ID
Exchange Rate	Rate Type ID

Intrastat	Item Number
Tax Detail	Tax Detail ID Account Number

Sales Order Processing

The Sales Order Processing module in the Microsoft Dynamics GP adapter contains the Sales Transaction destination.

Sales Transaction The Sales Transaction destination has the following properties. This destination has mapping options which can be set to optimize the performance of the integration.

Destination Edit Modes	Insert Only
Mapping Options	
Quantity Shortage	
	Choose from the following rules to specify how to respond if sufficient quantities are not available to fill the order. **Cancel Document** **Override Shortage**—Ignore the shortage and sell the entire quantity. **Sell Balance**—Sell the quantity available in inventory for the site. **Back Order All**—Back order the entire quantity. **Back Order Balance**—Sell the quantity available and back order the shortage quantity. **Cancel All**—Cancel the entire quantity for the item. **Cancel Balance**—Sell the quantity available at the site and cancel the shortage quantity.
Over Credit Limit	
	Choose from the following rules to specify the action to take when the credit amount exceeds the credit limit. **Cancel Document** **Override**—Use the **Rule Properties** pane to enter the password needed to override the credit limit.
Missing Batch	
	Choose from the following rules to specifies the action to take when the batch is missing. **Cancel Document** **Add New Batch**—Set the following rule properties to specify characteristics of the new batch. **Comment**—Enter a text comment for the new batch. **Transactions**—Enter the number of transactions for the batch. **Batch Totals**—Enter the actual currency amount in the batch.
Customer Hold	
	Choose from the following rules to specify the action to take when a customer hold exists. **Cancel Document** **Override Hold**—Use the **Rule Properties** pane to enter the password needed to remove a customer hold.
Multicurrency Fields	
	Choose from the following rules to specify the currency. **Use Functional** **Use Originating**
Delete with Password	
	Choose from the following rules to specify whether or not a password is required to delete a transaction. **No Password Required** **Provide Password**—Use the **Rule Properties** pane to specify the password to use to delete the transaction.
Missing Comment	

	Choose from the following rules to specify the action to take when the comment ID does not exist. **Cancel Document** **Add New Comment**—Use the following rule properties to add a new comment. **Series**—Select the value for the series. **Comment**—Enter a comment.
Restrictions	This destination is insert–only. It cannot be used to update existing sales order transactions.
Required Fields	
Sales Transaction	Type Type ID Document No Date Batch ID Customer ID Customer Name Currency ID
Commissions	Salesperson ID Territory ID
Distributions	Account Number Distribution Type
Distributions\Analysis Posting Detail	Analysis Group ID
Distributions\Analysis Posting Detail\Entries	Analysis Code ID
Exchange Rate	Rate Type ID Exchange Rate
Items	Item Number U Of M Quantity
Items\Intrastat	Tax Registration Country Code Transport Mode Transaction Nature
Items\Item Detail	Site ID Price Level
Items\Item Detail\Bin	Bin Quantity Selected
Items\Item Detail\Line Item Taxes	Tax Detail ID
Items\Item Detail\Lot Numbers	Lot Number Quantity
Items\Item Detail\Serial Numbers	Serial Number
Items\Returned Quantities	On Hand Returned In Use In Service Damaged
Payments	Date Posting Date Cash Receipt Number Payment Amount Cash Account Deposits
Sales Tax Detail Summary	Tax Detail ID Account Number
User–Defined Fields\Tracking Numbers	Tracking Number

Payroll

The Payroll module in the Microsoft Dynamics GP adapter contains destinations for Payroll Manual Checks, Payroll Master, and Payroll Transaction.

Payroll Manual Checks The Payroll Manual Checks destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	
Missing Batch	
	Choose from the following rules to specifies the action to take when the batch is missing. **Cancel Document** **Add Payroll Batch**—Set the following rule properties to specify characteristics of the new batch. • **Comment**—Enter a text comment for the new batch. • **Transactions**—Enter the number of transactions for the batch. • **Employees**—Enter the number of employees in the batch.
Net Pay Amount Less than Employee Minimum Net Pay	
	Choose from the following rules to specify the behavior when the net employee pay is below the set minimum. **Cancel Document** **Allow with Warning** **Allow without Warning**
FICA Medicare Tax Exceeded Year Maximum	
	This option is available with the Transactions recordset. Choose from the following rules to specify the behavior when the FICA Medicare Tax limit has been exceeded. **Cancel Document** **Allow with Warning** **Allow without Warning**
FICA Social Security Tax Exceeded Year Maximum	
	This option is available with the Transactions recordset. Choose from the following rules to specify the behavior when the FICA Social Security Tax limit has been exceeded. **Cancel Document** **Allow with Warning** **Allow without Warning**
Transaction Exceeded Pay Period Maximum	
	This option is available with the Transactions recordset. Choose from the following rules to specify the behavior when the maximum pay for a period has been exceeded. **Cancel Document** **Allow with Warning** **Allow without Warning**
Transaction Exceeded Year Maximum	
	This option is available with the Transactions recordset. Choose from the following rules to specify the behavior when the maximum pay for a year has been exceeded. **Cancel Document** **Allow with Warning** **Allow without Warning**
Transaction Exceeded Lifetime Maximum	
	This option is available with the Transactions recordset. Choose from the following rules to specify the behavior when the maximum pay for a lifetime has been exceeded. **Cancel Document** **Allow with Warning** **Allow without Warning**
Required Fields	

Payroll Manual Checks	Check Type Payment/Adjustment Number Batch ID Checkbook ID Check Date Employee ID
Transactions	Transaction Code Beginning Date Ending Date Amount

Payroll Master The Payroll Master destination has the following properties.

Destination Edit Modes	Insert Only Update Only Insert and Update
Mapping Options	
Invalid Social Security Number	
	Choose from the following rules to specify the behavior when a Social Security Number does not have a valid format. **Cancel Document** **Allow with Warning** **Allow without Warning**
Default Information From	
	This option is available in the Payroll Master, Benefits, Deductions, Pay Codes, and Local Tax recordsets. Choose from the following rules to specify where any default information will be retrieved from. **No Default** **Employee Class**—For Payroll Master recordset. **Payroll Company Record**—For Benefits, Deductions, and Pay Codes recordsets. **Employee Pay Code**—For Pay Codes recordset. **Company Local Tax Code**—For Local Tax recordset.
Duplicate Social Security Number	
	Choose from the following rules to specify the behavior when a duplicate Social Security Number is encountered. **Cancel Document** **Allow with Warning** **Allow without Warning**
Inactive Employee Class Change	
	Choose from the following rules to specify the action to take when the class of an inactive employee is changed. **Cancel Document** **Allow with Warning** **Allow without Warning**
Employee Reactivation	
	When an employee is reactivated, choose from the following rules to specify which records for that employee are also reactivated. **Do Not Reactivate Any** **Reactivate All Records** **Reactivate Specific Records**—Use the **Rule Properties** pane to specify which of the following records to reactivate. • State Tax Records • Local Tax Records • Pay Records • Deduction Records • Benefit Records

No Deductions Specified	
	This option is available in the Benefits recordset. Choose from the following rules to specify the behavior when no deductions have been specified. **Cancel Document** **Allow with Warning** **Allow without Warning**
No Pay Codes Specified	
	This option is available in the Benefits and Deductions recordsets. Choose from the following rules to specify the behavior when no pay codes have been specified. **Cancel Document** **Allow with Warning** **Allow without Warning**
Pay Codes not Set Up for Employee	
	This option is available in Benefits and Deductions recordsets. Choose from the following rules to specify the behavior when no pay codes have been set up for an employee. **Cancel Document** **Allow with Warning** **Allow without Warning**
Adding Active Records to Inactive Employee	
	This option is available in the Benefits, Deductions, Pay Codes, Local Tax, and State Tax recordsets. Choose from the following rules to specify the behavior when the integration attempts to add records for an employee that is inactive. **Cancel Document** **Make Inactive with Warning** **Make Inactive without Warning**
Deductions not Set Up for Employee	
	This option is available in Benefits recordset. Choose from the following rules to specify the behavior when no deductions have been set up for an employee. **Cancel Document** **Allow with Warning** **Allow without Warning**
Inactive Pay Code	
	This option is available in the Deductions and Pay Codes recordsets. Choose from the following rules to specify the behavior when an inactive pay code is used. **Cancel Document** **Allow with Warning** **Allow without Warning**
Pay Rate Change	
	This option is available in Pay Codes recordset. Choose from the following rules to specify the behavior when a pay rate changes. **Cancel Document** **Roll Down to all Employees Using this Code** **Do Not Roll Down**
Shift Code Change	
	This option is available in Pay Codes recordset. Choose one of the following rules to specify the behavior when a shift code changes. **Cancel Document** **Roll Down to all Employees Using this Code** **Do Not Roll Down**
Data Entry Default for Salary Pay Codes	
	This option is available in the Pay Codes recordset. Choose from the following rules. **Cancel Document** **Allow with Warning** **Allow without Warning**
Inactive Local Tax Code	

	This option is available in the Tax Information\Local Tax recordset. Choose from the following rules to specify the behavior when an inactive local tax code is used. **Cancel Document** **Allow with Warning** **Allow without Warning**
Inactive State Tax Code	
	This option is available in the Tax Information\State Tax recordset. Choose from the following rules to specify the behavior when an inactive state tax code is used. **Cancel Document** **Allow with Warning** **Allow without Warning**
Required Fields	
Payroll Master	Employee ID Last Name First Name Soc Sec Number Department Position
Additional Information	Work Hours Per Year
Addresses	Address ID
Benefits	Benefit Code Start Date
Benefits\Based on Codes	Code
Deductions	Deduction Code Start Date
Deductions\Based on Codes	Code
Pay Codes	Pay Code
Tax Information\Local Tax	Local Code
Tax Information\State Tax	State

Payroll Transaction The Payroll Transaction destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Required Fields	
Payroll Transaction	Batch ID
Entries	Transaction Type Transaction Code Date From Date To Position

Inventory

The Inventory module in the Microsoft Dynamics GP adapter contains destinations for Inventory Item and Inventory Transaction.

Inventory Item The Inventory Item destination has the following properties.

Destination Edit Modes	Insert Only Update Only Insert and Update
Mapping Options	None
Required Fields	
Inventory Item	Item Number Description Sales U of M Schedule ID
Currency	Currency ID
Pricing\Items	Price Level U of M Currency ID
QTY/Sites	Site Display Options
Vendors	Vendor ID Currency ID FOB

Inventory Transaction The Inventory Transaction destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	
Missing Batch	
	Choose from the following rules to specify the action to take when the batch is missing. **Cancel Document** **Add New Batch**—Set the following rule properties to specify characteristics of the new batch. • **Comment**—Enter a text comment for the new batch. • **Transactions**—Enter the number of transactions in the batch. • **Quantity Total**—Enter the total quantity in the batch.
Missing Serial Number	
	Choose from the following rules to specify how to respond if a serial number is missing. **Cancel Document** **Add New Serial Number**
Adjustment Override	
	Use this rule to integrate more quantities than are available in the system. Choose from the following rules to specify how Integration Manager will respond when the adjustment quantity in the source is greater than the quantity available in the system. **Cancel Document** **Override**—Override the available quantities and integrate a negative quantity into Microsoft Dynamics GP.
Variance Override	
	Use this rule to integrate more quantities than are available in the system. Choose from the following rules to specify how Integration Manager will respond when the Variance quantity in the source is greater than the quantity available in the system. **Cancel Document** **Override**—Override the available quantities and go negative in Microsoft Dynamics GP.
Missing Lot Number	
	Choose from the following rules to specify how to respond if a lot number is missing. **Cancel Document** **Add New Lot Number**
Restrictions	This destination is insert–only. It cannot be used to update existing inventory transactions.

Required Fields	
Inventory Transaction	Document Type Number Date Batch ID
Items	Item Number U of M Quantity Unit Cost Site ID
Items\Bins	Bin Quantity Selected
Items\Lot Numbers	Lot Number Quantity Selected
Items\Serial Numbers	Serial Number

Setup

The Setup module in the Microsoft Dynamics GP adapter contains a destination for Exchange Rate.

Exchange Rate The Exchange Rate destination has the following properties.

Destination Edit Modes	Update Only
Mapping Options	
Override Rate Variance	
	Choose from the following rules to specify how to respond If the exchange rate exceeds the rate variance. **Cancel Document** **Use Password**—Use the Rule Properties pane to specify the password needed to override the rate variance.
Modify Exchange Rate	
	Choose from the following rules to specify how to respond If an exchange rate is to be modified by an integration. **Cancel Document** **Use Password**—Use the Rule Properties pane to specify the password needed modify the exchange rate. **Modify Exchange Rate**—Modify the exchange rate without a password.
Required Fields	
Exchange Rate	Exchange Table ID
Exchange Rate\Exchange Rate Maintenance	Date Exchange Rate Expiration Date

Microsoft Dynamics GP eConnect destination mappings

This part of the documentation contains reference information you can use to set up destination mappings for your integrations that use destinations in the Microsoft Dynamics GP eConnect adapter. The tables that follow describe the properties and restrictions for each destination in the Microsoft Dynamics GP eConnect adapter. The tables also include required fields for each recordset in each destination. You need to select a rule for each of the required fields in the root recordset for the destinations you use. Root recordsets are in bold text.

*The tables also include the mapping options for each destination. These mapping options are on the **Options** tab of the Integration Mapping window. Because each child recordset contains the **Record Source** mapping option, it is not included in the following tables.*

Receivables Management

The Receivables Management module in the Microsoft Dynamics GP eConnect adapter contains destinations for Customer.

Customer The Customer destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Restrictions	
	The Customer ID should be in all capital letters. The rule property for Case Conversion can be set to Convert to Upper if the source file has the Customer ID in lowercase letters.
Required Fields	
Customer	Customer ID
Addresses	Address ID

Inventory

The Inventory module in the Microsoft Dynamics GP eConnect adapter contains destinations for Inventory Item and Inventory Transaction.

Inventory Item The Inventory Item destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Restrictions	
	This destination is insert-only. It cannot be used to update existing inventory transactions.
	Adding components to kit items is not supported.
	Entry of Internet information specific to each item is not supported.
Required Fields	
Inventory Item	Item Number Description U of M Schedule ID
Currency	Currency ID
Price List/Detail	Price Level U of M
Vendors	Vendor ID

Inventory Transaction The Inventory Transaction has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Restrictions	
	This destination is insert-only. It cannot be used to update existing inventory transactions.
	Transactions for lot-numbered and serial-numbered items are not supported.
Required Fields	
Inventory Transaction	Document Type Date Batch ID
Items	Item Number Quantity

Purchasing

The Purchasing module in the Microsoft Dynamics GP eConnect adapter contains destinations for Purchasing Order Entry, Receivings Transaction Entry, and Vendor.

Purchasing Order Entry The Purchasing Order Entry destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Restrictions	
	This destination is insert-only. It cannot be used to update existing purchase orders.
	Multicurrency transactions are not supported. The Currency ID for the transaction must always be the ID of the functional currency.
	You must use the Item Vendor Maintenance window to link inventory items to vendors before they can be used in a purchase order and received. Automatic linking is not supported.
	Sales document commitment is not supported.
	Distributions to unit accounts are not supported.
Required Fields	
Purchase Order Entry	Vendor ID

Receivings Transaction Entry The Receivings Transaction Entry destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Restrictions	
	Multicurrency transactions are not supported. The Currency ID for the transaction must always be the ID of the functional currency.
	Tax on Shipment/Invoice receipts will not be automatically calculated based on the vendor, items and locations.
	Receipt of inventory items that are lot–numbered or serial–numbered is not supported.
	You must use the Item Vendor Maintenance window to link inventory items to vendors before they can be used in a purchase order and received. Automatic linking is not supported.
	Sales Document Commitment is not supported.
Required Fields	
Receivings Transaction Entry	Batch ID Vendor ID
Distributions	Account Type Debit Credit

Vendor The Vendor destination has the following properties.

Destination Edit Modes	Insert Only Update Only Insert and Update
Mapping Options	None
Restrictions	None
Required Fields	
Vendor	Vendor ID Name
Addresses	Address ID

Sales Order Processing

The Sales Order Processing module in the Microsoft Dynamics GP eConnect adapter contains destinations for Sales Transaction.

Sales Transaction The Sales Transaction destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Restrictions	
	This destination is insert-only. It cannot be used to update existing sales order transactions.
	Only Order and Invoice document types are supported. Quotes, returns and back orders are not supported.
	Multicurrency, VAT and Intrastat are not supported. The Currency ID for the transaction must always be the ID of the functional currency.
	Entering deposits on Order documents is not supported. Entering payments on Invoices is supported.
	Automatic fulfillment of lot-numbered and serial-numbered items is not supported. These items can be entered on an order or invoice only if the Document ID is set up to use a separate fulfillment process.
	For quantity shortages, only the Override Shortage option is allowed. Users who want to select other options can set up their document types to allocate by document or batch.
	Entering documents as "repeating" is not supported.
	Commissions fields and tables cannot be populated.
	Automatic calculation of sales tax is not supported. Users can specify tax details and tax amounts at the document header level in the integration.
	Line item taxes are not supported.
	Process holds are not supported. If document types have been set up for automatic process holds, those process holds will not automatically be applied.
	Payment terms that have the discount type "EOM" are not supported. When this discount type is used, the document will integrate successfully, but the discount date will be set to 0/0/00.
	No warning will be issued if the transaction date falls within a period that has been closed.
	The following SOP functions are always allowed: Markdown, Allow Prices Below Cost, Override Prices, Override Quantity Shortages, and Override Price Levels.
	Customer credit limits and hold status are not checked during the integration process.
	General ledger account numbers that contain blank spaces cannot be handled by this adapter.
	Unit account distributions are not supported.
Required Fields	
Sales Transaction	Type Type ID Batch ID Customer ID
Distributions	Account Number Distribution Type Debit Amount Credit Amount
Items	Item Number
Sales Tax Detail Summary	Tax Detail ID

Financial

The Financial module in the Microsoft Dynamics GP eConnect adapter contains destinations for GL Account and GL Transaction.

GL Account The GL Account destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Restrictions	
	This destination is insert-only. It cannot be used to update existing General Ledger account records.
	Only posting accounts are supported.
	Account Currencies are not supported.
	Account Analysis Defaults are not supported.
Required Fields	
GL Account	Account Number Category

GL Transaction The GL Transaction destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Restrictions	
	This destination is insert-only. It cannot be used to update existing General Ledger account records.
	Only posting accounts are supported.
	Account Currencies are not supported.
	Account Analysis Defaults are not supported.
Required Fields	
GL Transaction	Batch ID Transaction Date Reference
Entries	Account Number Debit Amount Credit Amount

Company

The company module in the Microsoft Dynamics GP eConnect adapter contains destinations for Shipping Method.

Shipping Method The Shipping Method destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Restrictions	None
Required Fields	
Shipping Method	Shipping Method Shipping Type

Project Accounting

The Project Accounting module in the Microsoft Dynamics GP eConnect adapter contains destinations for Employee Expense and Timesheet Entry.

Employee Expense The Employee Expense destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Restrictions	None
Required Fields	
Employee Expense	Transaction Type Document No. Document Date Batch ID Employee ID Address ID Start Date End Date
Line Entries	Date Project Number Cost Category ID Currency ID Line Seq Num
Tax Details	Line Seq Num Tax Detail ID Tax Type

Timesheet Entry The Timesheet Entry destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Restrictions	None
Required Fields	
Timesheet Entry	Transaction Type Document Number Document Date Batch ID Employee ID Rep Period Period Begin Currency ID
Line Entries	Date Project Number Cost Category ID
Notes	The PA Total Quantity field is required when values are entered into the field. The PA Reporting Suffix field is required and must be incremented if there is more than one timesheet for the same employee in the same reporting period.

Fixed Asset Manager

The Fixed Asset Manager module in the Microsoft Dynamics GP eConnect adapter contains destinations for Asset Book, Asset General Information, Asset Insurance, Asset Lease, and Asset User Data.

Asset Book The Asset Book destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Restrictions	None
Required Fields	
Asset Book	Asset ID Book ID
Notes	There are fields that become required depending on you map to the Asset Book. Depreciate to date Special depreciation allowance: 1=No; 2=Yes Special depreciation allowance percentage Luxury auto: 1=No; 2=Yes Luxury van or truck: 0=No; 1=Yes Luxury electric auto: 0=No; 1=Yes

Asset General Information The Asset General Information destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Restrictions	None
Required Fields	
Asset General Information	Asset ID Description Class ID Acquisition Date
Book	Book ID
Notes	The Fixed Asset Account Group data is not available to be mapped when you run the FA General eConnect Destination in Integration Manager for Microsoft Dynamics GP. You can use the data from the Fixed Asset setup information in Microsoft Dynamics GP by using one of the following methods. Method 1 Set up the Fixed Asset Class ID to have a default Account Group ID field in Microsoft Dynamics GP. 1. Open the Class Setup window (Financial >> Setup >> Fixed Assets >> Class). 2. Set up a class for the new assets or select an existing class. 3. Select an account group ID. 4. Click Save. 5. Map the Class ID field in the Asset General Information destination mapping folder in the Integration Manager Destination Mapping window. The Asset Account Master (FA00400) table will be updated with the account group data that is associated with the class. Method 2 Set up the default accounts for the company to have an account group ID in Microsoft Dynamics GP 1. Open the Fixed Assets Company Setup window (Financial >> Setup >> Fixed Assets >> Company). 2. Choose the Default Accounts expansion button to open the Default Accounts window. 3. Enter an account group ID. 4. Click OK. 5. Choose Save in the Fixed Assets Company Setup window. You do not have to map any specific field within Integration Manager. The Asset Account Master (FA00400) table will be updated with the account group data that is associated with the default account information in the Fixed Asset company setup.

Asset Insurance The Asset Insurance destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Restrictions	None
Required Fields	
Asset Insurance	Asset ID

Asset Lease The Asset Lease destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Restrictions	None
Required Fields	
Asset Lease	Asset ID

Asset User Data The Asset User Data destination has the following properties.

Destination Edit Modes	Insert Only
Mapping Options	None
Restrictions	None
Required Fields	
Asset User Data	Asset ID

Microsoft Dynamics GP eConnect destination mappings for Analytical Accounting

You can set up destination mappings for Analytical Accounting integrations that use destinations in the Microsoft Dynamics GP eConnect adapter. The following mapping tables describe each field in each recordset of the Analytical Accounting destinations. Each table contains the following information.

Field Name The name of the field in the Integration Mapping window.
Data Type The type of data in each field.
Length The number of characters that can be stored in the field.
Required States whether the field is required.

Be sure that you have a transaction/document number when you map your destinations.

Destinations

You can map Analytical Accounting integrations for the following destinations: Inventory Transaction, Receivings Transaction Entry, Sales Transaction, and GL Transaction. Each of these destinations contain the following recordsets.

Analytics recordset			
Field Name	**Data Type**	**Length**	**Required**
Account Number	string	75	No
Assign ID*	long integer	4	No
Amount*	numeric	14	No
Assigned Percent*	numeric	14	No

Analytics recordset			
Field Name	**Data Type**	**Length**	**Required**
Reference	string	30	No
Note	string	32768	No
*Not available in the Inventory Transaction Destination			

Dimensions recordset			
Field Name	**Data Type**	**Length**	**Required**
Transaction Dimension	string	30	No
Transaction Dimension AlphaNumeric Code	string	30	No
Transaction Dimension Numeric	numeric	14	No
Transaction Dimension Boolean	boolean	2	No
Transaction Dimension Date	date	8	No
Transaction Dimension ID	long integer	4	No
Transaction Dimension Code ID	long integer	4	No

XML source adapter

With Integration Manager and its XML adapter, you can quickly and easily move XML data into Microsoft Dynamics GP.

The XML source adapter allows you to create integrations that use XML files as source data. Then, you can map the source data to any destination, depending on which destination adapters you have.

To use the XML source adapter, you use the XML Source Definition window to define your XML source. You can define the source from scratch or simply point to an XML file and let the adapter do the work for you. Integration Manager uses this definition to convert the XML structure into a document definition it can use. A document definition is the metadata that Integration Manager uses to describe the structure and content of the source. It describes recordsets, hierarchical relationships, fields, data types, and more. It is analogous to an XML schema, but it usually contains more information.

After the document definition is set up, you use the Integration Mapping window to map the source data to a destination. In addition to mapping source values to the destination, you can choose from a number of additional rules for each field, including:

- Using a field's default value
- Using a constant value
- Using VBScript to apply custom logic to a field

When you add an XML source to an integration, you are telling Integration Manager the format of the source data to be used. When you add a source, you can choose to define a new source or select an existing source. You complete adding a XML source by defining the properties.

You can add only one XML source to any integration. You cannot add two XML sources and link them together as you can with ODBC and text sources. You must remove the old source before adding the new XML source, or you must create a new integration.

Microsoft Dynamics GP Record Source mapping option

On the **Options** tab of the Integration Mapping window, you can specify additional options that indicate how Integration Manager should handle special circumstances. This part of the documentation contains reference information for some of the additional mapping options that are available.

Each child recordset in the Microsoft Dynamics GP destination adapter contains the **Record Source** mapping option.

The following table describes the rules available with the **Record Source** mapping option.

Rule	Description
Use Default	The default values for the fields in the recordset are used, regardless of the field rule selected for each field.
Use Field Rules	The rules chosen for individual fields in the recordset are used.
Use Source Recordset	This rule is available only for recordsets that allow multiple sets for each record. It allows you to specify a query to associate with the recordset. One set of items will be read into the recordset for each row returned by the selected query.
Empty	All fields in the recordset will be made empty, regardless of any default values or rules you have applied to individual fields.
Default (Non–Imported)	For some recordsets, such as Distributions, Microsoft Dynamics GP provides default values. If you choose to map selected fields in the recordset, you can choose this rule to have Microsoft Dynamics GP provide default values for the remaining fields.

Part 5: Using VBScript

Integration Manager provides scripting capabilities so that you are able to customize integrations. Integration Manager uses the command set, grammar, and syntax of the Microsoft Visual Basic Scripting™ Edition (VBScript), a subset of the Microsoft Visual Basic programming language. It can be used throughout Integration Manager to do the following activities.

- Map and transform data
- Trigger events and commands
- Process user input
- Provide feedback

By adding scripts to fields and integration events, you can significantly extend the functionality of Integration Manager.

This part of the documentation includes the following information.

- Chapter 20, "Using scripts," provides an overview of how VBScript can be used in Integration Manager.

- Chapter 21, "VBScript objects," provides an overview of how objects can be used in Integration Manager.

- Chapter 22, "Functions," provides an overview of how functions can be used in Integration Manager.

Chapter 20: Using scripts

You can use VBScript–which is included in Integration Manager–to attach scripts that automate integrations. VBScript allows you to attach code that runs as various actions are performed by Integration Manager.

This part of the documentation includes the following information.

Overview of VBScript

VBScript is a language that can be embedded into products such as Integration Manager to enhance them with scripting capabilities. Developed by Microsoft, VBScript is based on Visual Basic. If you have used Visual Basic or Visual Basic for Applications (VBA), you are already familiar with the syntax of VBScript.

This part of the documentation assumes that you have basic knowledge of scripting in general, and VBScript in particular. It is not intended to teach VBScript. Rather, it is intended to show you how to use the implementation of VBScript that has been added to Integration Manager.

For more information about VBScript, go to the MSDN online library (http://msdn.microsoft.com) and click on the Library link) and search for VBScript.

Attaching scripts to integrations

Use the following procedure to attach a script to an integration.

To attach a script to an integration:

1. Open an integration.

2. From the Integration window, click **Properties**, and choose the **Scripts** tab.

An icon next to a script indicates that the script is attached to this integration.

3. Select the script type for the integration.

   The type of script you choose depends on when the script should run.

Script event	Description
Before Integration	The script runs once at the start of the integration.
Before Document	The script runs once at the beginning of each document (record). It runs after the appropriate sources for the document have been read, but before any values in the destination have been set.
Before Document Commit	The script runs once for each document, after all of the destination fields have been set based on the selected rules, but before the document is actually inserted or updated.
Document Warning	The script runs each time a warning occurs for a document.
After Document	The script runs once after each document that has been inserted or updated.
Document Error	The script runs each time an error occurs for a document.
After Integration	The script runs once after the entire integration has finished.
Integration Error	The script runs each time an error occurs for the integration process as a whole.

4. To attach a script, select it in the list and click **Open Script**.

   – or –

   Double-click the name of the script.

   The Script Editor window opens, where you can write the script.

*To remove a script, select the script in the list and click **Remove Script**.*

Attaching scripts to ODBC or text sources

ODBC or text sources are accessed using VBScript objects. You can use these objects only in the **Before Query** and **After Query** scripts. For more information, refer to Chapter 21, "VBScript objects." The most common tasks performed with the query object are:

- Supplying row selection criteria for the source.
- Specifying a new location for source files.
- Deleting source files or records after the source has completed.

Use the following procedure to attach a script to an ODBC or text source.

To attach a script to an ODBC or text source:

1. Open an integration.

2. From the Integration window, select the source, choose **Properties** on the toolbar, and choose the **Scripts** tab.

   A list of the scripts for the source appears.

3. Select the script from the list.

   The type of script you choose depends on when the script should run.

Script event	Description
Before Query	The script runs once before any data is retrieved from the source.
After Query	The script runs once after Integration Manager has processed all source records for the source.

4. Click **Open Script**.

   The Script Editor window appears, where you can write the script.

*To remove a script, select the script in the list and choose **Remove Script**.*

Attaching scripts to fields

Use the following procedure to attach scripts to fields.

To attach a script to a field:

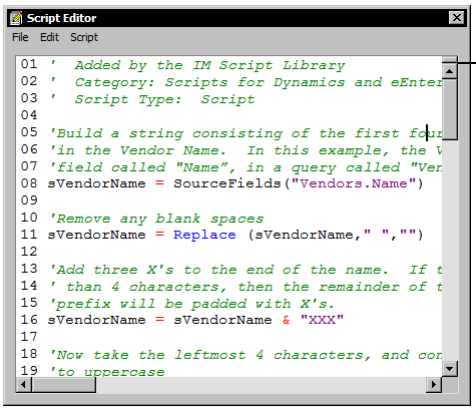
1. Open an integration.
2. From the Integration window, double-click **Destination Mapping**.

   The Integration Mapping window opens.
3. Set the rule for the field to **Use Script**. In the **Rule Properties**, click the lookup button for the **Script Text** rule to open the Script Editor window.

*To remove a script, select the **Script Text** rule property for the field. Choose **Edit** and choose **Remove Script**. After you remove the script, be sure to change the field rule to some value other than **Use Script**.*

Using the Script Editor window

Integration Manager has a built-in Script Editor window that you can use to write scripts for your integration. The Script Editor window automatically opens when you attach a script. (See *Attaching scripts to integrations* on page 149, *Attaching scripts to ODBC or text sources* on page 151, and *Attaching scripts to fields* on page 152.)



Point and drag this rectangle to position the horizontal splitter.

The Script Editor acts like any basic text editor. The commands on the **File** and **Edit** menus allow you to edit, cut, copy, paste, and save your scripts.

You can use the VBScript Library for Integration Manager to check the syntax of your script. In the Script Editor window, choose **Script** > **Script Library** to open the VBScript Library**.** The VBScript Library for Integration Manager is a collection of commonly used scripts that you might find useful in your integrations.

To make composing and editing scripts easier, the Script Editor includes line numbering, which you can turn on and off from the **Edit** menu, and it includes unlimited undo and redo functionality, which is also on the **Edit** menu. If you select multiple lines and press TAB, you can indent several lines at once. Also, if you are working on a long script, you can use a horizontal splitter so you can see two areas of the script at the same time. This horizontal splitter is located just above the vertical scroll bar.

As you compose scripts, you will notice that the Script Editor automatically highlights syntax and maintains the tab positions from the previous line. It also employs keyword case normalization as you type. For example, it changes "dim" to "Dim".

To save the script and close the Script Editor, choose **File** > **Save and Close**.

Working with source fields

Source fields are the individual field values returned from a source. You can access them using the Source Fields object. See *SourceFields object* on page 164 for more information. Source field information is available only after the sources for an integration have been read. This means you can use them only for the **Before Document**, **Before Document Commit**, **After Document**, and **Field** scripts.

Working with destination fields

Destination fields are the individual fields in the integration destination. You access them using the Destination Fields object. See *DestinationFields object* on page 156 for more information. Integration Manager uses the field rule selections to set the value of all destination fields just prior to executing the **Before Document Commit** script. If your script refers to a single destination field, consider attaching it directly to that field and using the *CurrentField object* on page 155. If your script refers to setting other destination fields, use the **Before Document Commit** script.

Order of events

It is helpful to know when scripts are run in relation to the actions performed as an integration runs. The following table describes the events and scripts performed as an integration runs, in the order in which they are performed.

Step	Event	Description
1	Preparation phase	Integration Manager prepares the rules, options, and other information for the integration.
2	**Before Integration** script	The **Before Integration** script runs.
3	**Before Source** script	Each source is started, but no information is retrieved. The **Before Source** script for each source runs.
4	Start the sources	Data from each source is retrieved.
5	**Before Document** script	The **Before Document** script runs. It can access information read by each source.
6	Set field values	To set the destination field values, Integration Manager uses the field rules that you specified in the Integration Mapping window. The scripts attached to the individual fields run.
7	**Before Document Commit** script	When all fields have been set for the destination, the **Before Document Commit** script runs.
8	Write/Update document	Integration Manager writes or updates the current document.
9	**After Document** script	The **After Document** script runs. The process returns to step 5 until all source results have been read.
10	**After Integration** script	After all documents have finished, the **After Integration** script runs.
11	**After Source** script	The **After Source** script for each source runs.
12	Close sources	All the sources close.

Null values

If you are using the SourceFields object to retrieve data from a source, but no value exists for that field in the current row, a NULL value is returned. You can't check for a NULL value directly. Instead, you must use the VBScript **IsNull** function. The following example shows how to use this function.

```
If IsNull(SourceFields("Fax")) Then
   CancelDocument "No fax number", 5000, "Fax"
End If
```

Variables

Variables are scoped at the script level. A variable may be created in a particular script (such as a **Before Integration** script) and assigned a value, but the variable will not be recognized by other scripts in the same integration (such as a **Field** script). The GetVariable and SetVariable methods are designed to store variables from one script so that they may be retrieved by another script.

Sometimes, you need to pass information from one script in the integration to another. The SetVariable statement, GetVariable function, and ClearVariables statement are available in Integration Manager to allow this. With these commands, you can create, set, and retrieve variables from any script in the integration. For more information, refer to Chapter 22, "Functions."

Debugging scripts

Any errors that prevent your scripts from running properly will be listed in the integration's Progress window.

If you are familiar with using a debugger, the Microsoft Script Debugger is available from msdn.microsoft.com. Refer to the documentation included with the debugger for a complete description of its features.

To use this debugger with Integration Manager, install the debugger and include the Stop statement in your VBScript code where debugging should begin. When this statement is encountered, the debugger starts, allowing you to debug from that point.

Microsoft Dynamics GP does not provide technical support for debugging scripts or using the Microsoft Script Debugger.

Chapter 21: VBScript objects

VBScript objects are a combination of code and data that can be treated as a unit. An object can be a piece of an application or an entire application. All objects are created as identical copies of their classes. Once they exist as individual objects, you can change their properties.

This part of the documentation includes the following information.

CurrentField object

The **CurrentField** object is used in scripts that are attached to fields in the Integration Mapping window. This object refers to the destination field that the script is attached to.

Syntax
CurrentField.[*property* | *method*]

Example
The following example sets the current field to the value 19.95.

```
CurrentField.Value = 19.95
```

FullName property
The **FullName** property returns a string containing the full name of a field.

Syntax
object.**FullName**

Parameters
object—A current field object.

Comments
This name also contains the names of any collections the current field may be part of.

Example
The following example sets the CurrentField variable to the full name of the current field.

```
SetVariable "CurrentField", CurrentField.FullName
```

Name property
The **Name** property returns a string containing the name of a field.

Syntax
object.**Name**

Parameters
object—A current field object.

Comments
This name does not contain any information about a collection the current field may be part of.

Example
The following example sets the CurrentField variable to the name of the current field.

```
SetVariable "CurrentField", CurrentField.Name
```

SetToDefault method
The **SetToDefault** method specifies that a destination field will use its default value.

Syntax
object.**SetToDefault**

Parameters
None

Comments
You can use the **HasDefault** property to verify that a destination field has a non-empty default value before you use the **SetToDefault** method.

Examples
The following example is the Before Document Commit script for a customer integration. If the CustomerType value from the Customer query is Preferred, the Finance Charge Percent is set to 8. Otherwise, it is set to use the field's default.

```
If SourceFields("CustomerType") = "Preferred" Then
    DestinationFields("Options.Finance Charge Percent") = 8
Else
    DestinationFields("Options.Finance Charge Percent").SetToDefault
End If
```

DestinationFields object

The **DestinationFields** object refers to any destination field in the Integration Mapping window. It is typically used in scripts attached to fields in the Integration Mapping window, and also can be used in the Before Document Commit script.

Syntax
DestinationFields(*name***)**.[*property* | *method*]

Parameters
name—A string containing the full name of the destination field. If the destination field is part of a collection, the full name includes the names of any collections the field is part of, separated by periods.

Example
The following example is the Before Document Commit script for a vendor integration. It sets the Comment 1 field to the value "Imported by IM".

```
DestinationFields("Comment 1") = "Imported by IM"
```

The following example is the Before Document Commit script for a customer integration. It sets the Credit Limit Amount field, which is part of the Options collection, to the CreditLimit value returned by a query.

```
DestinationFields("Options.Credit Limit Amount") =
SourceFields("CreditLimit")
```

DefaultIsSet property

The **DefaultIsSet** property returns a boolean indicating whether a destination field is set to use the default value. True indicates it uses the default value. False indicates it does not.

Syntax

object.**DefaultIsSet**

Parameters

object—A destination field object.

Example

The following example is the Before Document Commit script for an integration. It examines the Transaction Date destination field to find out whether it is set to the default value. If it is, a message appears and the integration is stopped.

```
If DestinationFields("Transaction Date").DefaultIsSet = True Then
    MsgBox "Have not set the date for the transaction"
    CancelIntegration
End If
```

HasDefault property

The **HasDefault** property returns a boolean indicating whether a destination field has a non-empty default value. True indicates the field has a non-empty default value, while false indicates it does not.

Syntax

object.**HasDefault**

Parameters

object—A destination field object.

Comments

Use the **HasDefault** property to verify that a destination field has a default value before you use the **SetToDefault** method.

Example

The following example verifies that the Post Sales In destination field in the Account object has a non-empty default value. If it does, the default value is used.

```
If DestinationFields("Post Sales In").HasDefault = True Then
    DestinationFields("Post Sales In").SetToDefault
End If
```

SetToDefault method

The **SetToDefault** method specifies that a destination field will use its default value.

Syntax

object.**SetToDefault**

Parameters

None

Comments

You can use the **HasDefault** property to verify that a destination field has a non-empty default value before you use the **SetToDefault** method.

Example

The following example is the Before Document Commit script for a customer integration. If the CustomerType value from the Customer query is Preferred, the Finance Charge Percent is set to 8. Otherwise, it is set to use the field's default.

```
If SourceFields("CustomerType") = "Preferred" Then
    DestinationFields("Options.Finance Charge Percent") = 8
Else
    DestinationFields("Options.Finance Charge Percent").SetToDefault
End If
```

Errors Collection object

The **Errors Collection** object represents a collection of errors generated during an integration. Unlike the other VBScript objects, you need to explicitly retrieve the Errors Collection using the GetVariable function.

Syntax

Set MyErrors = GetVariable("**Errors**")**MyErrors**.[property]

Parameters

None

Example

```
Dim MyErrors, MyError
Set MyErrors = GetVariable("Errors")
Set MyError = MyErrors.LastError
```

Count property

The **Count** property returns the total number of errors in the collection.

Syntax

object.**Count**

Example

```
If MyErrors.Count > 0 then
    MsgBox "An error occurred"
End If
```

LastError property

The **LastError** property contains the Error Object for the last error or warning that occurred.

Syntax

object.**LastError**

Examples

```
Set MyError = MyErrors.LastError
```

Item property

Description

The **Item** property acts as the array indexer into the Errors Collection. You can retrieve a specific Error using this property by passing in the ordinal of the Error you want. The ordinal starts at 1.

Syntax

object.**Item**

Example

This example gets the Error Object at the start of the collection:

```
Set MyError = MyErrors.Item(1)
```

Error object

The **Error** object is used in scripts to get information about a specific error, such as the time it occurred or the text of the error. The only way to access a specific error is through the Errors Collection object. An Error object may contain warnings as well as errors.

Syntax

Set MyError = GetVariable("Errors").**LastError**
MyError.[property]

– Or –

GetVariable("Errors").LastError.[property]

Parameters

None

Example

The following example demonstrates retrieving the last error that occurred in the integration:

```
Dim MyErrors, MyError
Set MyErrors = GetVariable("Errors")
Set MyError = MyErrors.LastError
```

The following example shows how to display the text of the last error or warning that occurred:

```
MsgBox GetVariable("Errors")._
LastError.MessageText
```

DateTime property

The **DateTime** property is a time stamp for when the error or warning occurred.

Syntax

object.**DateTime**

Severity property

Description

The **Severity** property indicates whether the Error Object is an error or a warning.

Syntax

object.**Severity**

Example

This example checks whether the error object is a warning:

```
If MyError.Severity = GetVariable("SeverityWarning") then
  MsgBox "This is a warning " & MyError.MessageText
End If
```

This example checks whether the error object is an error:

```
If MyError.Severity <> GetVariable("SeverityWarning") then
  MsgBox "This is an error " & MyError.MessageText
End If
```

MessageText property

The **MessageText** property contains the description of the error or warning that occurred.

Syntax

object.**MessageText**

Query object

The **Query** object is used to access and set properties of a source query. It can be used only in scripts attached to queries.

Syntax

Query.[*property* | *method*]

Parameters

None

Example

The following example is the Before Query script. It adds additional selection criteria to a query used for a vendor integration.

```
Query.AdditionalCriteria = "State = 'ND'"
```

AdditionalCriteria property

The **AdditionalCriteria** property allows you to add additional selection criteria to a text query or simple ODBC query.

Syntax

Query

Parameters

object—The query object to which additional selection criteria is applied.

criteria—A string containing the additional criteria to apply.

Comments

You must set the **AdditionalCriteria** property in the Before Query script.

If you refer to a column whose name contains spaces or other special characters, be sure to enclose the column name in square brackets [].

You can use parentheses, logical AND, logical OR, Like and wildcard (%) operators in the criteria string, similar to how you use them in the Query Properties window.

Example

object.**AdditionalCriteria** [= *criteria*]

object—The query object to which additional selection criteria is applied.

```
Query.AdditionalCriteria = "[Customer Name] Like 'A%'"
```

OverrideCriteria property

The **OverrideCriteria** property allows you to replace the selection criteria used for a text query or simple ODBC query.

Syntax

object.**OverrideCriteria** [= *criteria*]

Parameters

object—The query object for which the selection criteria will be replaced.

criteria—A string containing the criteria.

Comments

You must set the **OverrideCriteria** property in the Before Query script.

If you refer to a column whose name contains spaces or other special characters, be sure to enclose the column name in square brackets [].

You can use parentheses, logical AND, logical OR, Like and wildcard (%) operators in the criteria string, similar to how you use them in the Query Properties window.

Example

The following example replaces the selection criteria for a query to include only customers from North Dakota.

```
Query.OverrideCriteria = "State = 'ND'"
```

OverrideFileLocation property

The **OverrideFileLocation** property allows you to specify a different file location for the source file used for a text query.

Syntax

object.**OverrideFileLocation** [= *path*]

Parameters

object—A query object for text query.

path—A string containing the new location of the source file. Note that this is the *location* of the source file. The name of the source file must remain the same.

Comments

You must set the **OverrideFileLocation** property in the Before Query script.

Example

The following example is a Before Query script. It reads the value of the Path variable, which was set in the Before Integration script. Then it uses the path information to override the location of the source file for the query.

```
Query.OverrideFileLocation = GetVariable("Path")
```

QueryName property

The **QueryName** property returns a string containing the name associated with a query object.

Syntax

object.**QueryName**

Parameters

object—A query object.

Example

The following example is the Before Query script for a query. It retrieves the name of the query and sets a variable to be used later in the integration.

```
SetVariable "QueryName", CStr(Query.QueryName)
```

DeleteSourceFile method

The **DeleteSourceFile** method allows you to delete the source file used for a text query.

Syntax

object.**DeleteSourceFile(**{*suppress*}**)**

Parameters

object—The query object for which the source file will be deleted.

suppress—An optional boolean indicating whether the default confirmation warning will be suppressed. True indicates the warning will be suppressed.

Comments

You can use the **DeleteSourceFile** method in the After Query script.

Example
The following example is the After Query script for a query. It deletes the source file from the query.

```
query.DeleteSourceFile()
```

DeleteSourceRecords method
The **DeleteSourceRecords** method allows you to delete the source records from a simple ODBC query that was used for an integration.

Syntax
object.**DeleteSourceRecords(**{*suppress*}**)**

Parameters
object—The query object for which the source records used will be deleted.

suppress—An optional boolean indicating whether the default confirmation warning will be suppressed. True indicates the warning will be suppressed.

Comments
You can use the **DeleteSourceRecords** method in the After Query script.

If the integration was completed successfully, you may want to delete the records from the source query if they are no longer required.

Example
The following example is the After Query script for a query. It deletes the source records from the query, without prompting the user.

```
Query.DeleteSourceRecords(True)
```

ExecuteSQL method
The **ExecuteSQL** method allows you to execute a SQL statement for an ODBC query that was used for an integration.

Syntax
object.**ExecuteSQL(***SQL_statement***)**

Parameters
object—The query object for which a SQL statement will be executed.

SQL_statement—A string containing the SQL statement to execute.

Comments
The **ExecuteSQL** method is typically used in the After Query script to complete any post processing, such as updating a status column, necessary for the data source.

The SQL statement you execute should not produce a result set.

Example
The following example is the After Query script for an integration. It executes a SQL statement that updates the IntStatus column of the Vendors table to indicate which items were imported by Integration Manager.

```
query.ExecuteSQL("Update Vendors Set IntStatus = 'Yes' where
IntStatus = '' ")
```

SourceFields object

The **SourceFields** object is used in scripts attached to fields in the Integration Mapping window. It refers to any field in the queries that are part of the integration.

Syntax
SourceFields(*name***)**

Parameters
name—A string containing the name of the source field. The full name includes the name of the query the field is part of, followed by a period. If the source field is part of the root query for the integration, you do not need to include the query name.

Example
The following example is the Before Document Commit script for a vendor integration. It retrieves a value of the VendorID field from a query used by the integration and stores it in a variable to be used later in the integration.

```
SetVariable "Document ID", CStr(SourceFields("VendorID"))
```

The following example is the Before Document Commit script for a General Ledger integration. It retrieves a value of the Doc Num field from the GL Header query used by the integration and stores it in a variable to be used later in the integration.

```
SetVariable "DocNum", CStr(SourceFields("GL Header.Doc Num"))
```

GPConnection object

The **GPConnection** object is used in scripts instead of the RetreiveGlobals dynamic link library that was used in Microsoft Dynamics GP 9.0 and previous versions.

Open method
The **Open** method allows you to open an ADO connection using the current GP user login information.This method uses the data source that is in use when Microsoft Dynamics GP is open. If you want to use a default company database (TWO or GPDAT) for this method, then you must set the **Open** value in the connection string before using the **Open** method. You will not be able to update the connection string after the **Open** method is called. There is no close method for this object. Once the connection is returned to the same way the connection object was initially created in the script, that connection object can be closed normally.

Syntax
object.**Open(**{*suppress*}**)**

Comments
All properties for the **Open** method will return string values.

Examples

The following example is the Open script. It opens the data connection.

```
set MyCon = CreateObject("ADODB.Connection")
MyCon.Connectionstring = "database=GPDAT"
GPConnection.Open(MyCon)
```

The following is an example of creating the ADO record set.

```
set recset = CreateObject("ADODB.Recordset")
```

The following is an example of creating the ADO connection.

```
set MyCon = CreateObject("ADODB.Connection")
```

The following is an example of executing the update command

```
recset = MyCon.Execute(updatecommand )
```

The following is an example of closing the ADO Connection.

```
MyCon.Close
```

The following are examples of retrieving the properties exposed by the new GPConnection object.

```
MsgBox GPConnection.GPConnUserDate
MsgBox GPConnection.GPConnInterCompanyID
MsgBox GPConnection.GPConnUserID
MsgBox GPConnection.GPConnUserName
MsgBox GPConnection.GPConnDataSource
```

UserDate property

The **UserDate** property contains the current user date.

Syntax

object.**UserDate**

CompanyID property

The **CompanyID** property contains the intercompany ID (company database ID).

Syntax

object.**CompanyID**

Example

The following is an example of initializing the connection string to specify a default database. In this case it is set to the current company. This could be set to a constant database, such as GPDAT.

```
MyCon.Connectionstring = "database=" + GPConnection.GPConnInterCompanyID
```

UserID property

The **UserID** property contains the current User ID.

Syntax

object.**UserID**

UserName property

The **UserName** property contains the name of the current user.

Syntax

object.**UserName**

Example

The following is an example of creating a string to update the customer name in the customer master table.

```
updatecommand = "update RM00101 set [CUSTNAME]='IM Customer' where
[CUSTNMBR]='AARONFIT0022'"
```

DataSource property

The **DataSource** property contains the name of the current data source that is being used in Microsoft Dynamics GP.

Syntax

object.**DataSource**

Chapter 22: Functions

VBScript contains many built-in functions. The available functions are a subset of those included in Visual Basic. Most are identical to their Visual Basic counterparts.

This part of the documentation includes the following information.

CancelDocument function

The **CancelDocument** function cancels the current document for an integration and writes an entry to the log file.

Syntax

CancelDocument [*message*, *source*, *status_code*, *field_name*, *field_value*]

Parameters

message—An optional string parameter corresponding to the Message item that is written to the log file for the integration.

source—An optional string parameter corresponding to the source item that is written to the log file for the integration.

status_code—An optional long integer parameter corresponding to the Status Code item that is written to the log file for the integration.

field_name—An optional string parameter corresponding to the Field Name item that is written to the log file for the integration.

field_value—An optional string parameter corresponding to the Field Value item that is written to the log file for the integration.

Comments

All parameters for this function are optional.

Example

The following example is the Before Document script for an integration. It examines the Fax number field and cancels the document if no fax number is available.

```
If IsNull(SourceFields("Fax")) Then
    CancelDocument "No fax number", 5000, "Fax"
End If
```

CancelIntegration function

The **CancelIntegration** function cancels the current integration and writes an entry to the log file.

Syntax

CancelIntegration [*message*, *source*, *status_code*]

Parameters

message—An optional string parameter corresponding to the Message item that is written to the log file for the integration.

source—An optional string parameter corresponding to the source item that is written to the log file for the integration.

status_code—An optional long integer parameter corresponding to the Status Code item that is written to the log file for the integration.

Comments

All parameters for this function are optional.

Example

The following example is the Before Integration script for an integration. It displays a dialog, asking the user to supply a password required to start the integration. If an incorrect password is supplied, the integration is canceled.

```
Dim Input
Input = InputBox("Enter password", "Password")

If Input <> "Access" then
    MsgBox("Incorrect password.")
    CancelIntegration "Incorrect password supplied."
End If
```

ClearVariables function

The **ClearVariables** function clears all variables set by the SetVariable function.

Syntax

ClearVariables

Parameters

None

Example
The following is the After Document script for an integration. It clears all variables that were set earlier in the integration.

```
ClearVariables
```

Related items
GetVariable function
SetVariable function

DocumentIsNew function

The **DocumentIsNew** function returns a value indicating whether the document (record) being written to a destination is new or is being updated.

Syntax
DocumentIsNew

Parameters
None

Return value
A boolean. True indicates a new document is being written to the destination. False indicates an existing document is being updated.

Comments
You can use the **DocumentIsNew** function in the Before Document, Before Document Commit and After Document scripts.

Example
The following example is the Before Document Commit script for a vendor integration. The Comment 2 field is set based on whether a new vendor is being written or an existing vendor is being updated.

```
If DocumentIsNew = True Then
    DestinationFields("Comment 2") = "New vendor from IM"
Else
    DestinationFields("Comment 2") = "Updated by IM"
End If
```

DocumentNo function

The **DocumentNo** function returns a long integer value indicating the sequence of the current document (record) being written to the destination. The value 1 indicates the first document that was written, 2 indicates the second document that was written, and so on.

Syntax
DocumentNo

Parameters
None

Return value
A long integer

Comments

You can use the DocumentNo function in the Before Document, Before Document Commit and After Document scripts.

Example

The following example is the After Document script for a vendor integration. It creates a text file that lists whether each document being imported created a new vendor or updated an existing vendor. The **DocumentNo** function indicates the order in which documents were added or updated. The DocumentID variable used was set in the Before Document script.

```
Const ForAppending = 8
Dim fso, f
Set fso = CreateObject("Scripting.FileSystemObject")
Set f = fso.OpenTextFile("C:\Program Files\Microsoft Dynamics\Integration
Manager\VendRslt.txt",
ForAppending, True)
If DocumentIsNew = True Then
    f.WriteLine "Document " & DocumentNo & " " &
    GetVariable("DocumentID") & " is new."
Else
    f.WriteLine "Document " & DocumentNo & "  " &
    GetVariable("DocumentID") & " was updated."
End If
```

Execute function

The **Execute** function starts another application.

Syntax

Execute *path* [, *timeout*] [, *window_style*]

Parameters

path—A string parameter specifying the complete path to the application to be started.

timeout—An optional long integer specifying the time in seconds to allow before control returns to Integration Manager. If this parameter is not included, the default timeout of 30 seconds is used. If the value –1 is used, the application launches asynchronously without any timeout value.

window_style—An optional integer specifying how the application to be executed appears. Use one of the following values:

Value	Description
0	The window is hidden and focus is passed to the hidden window.
1	The window has focus and is restored to its original size and position.
2	The window is displayed as an icon with focus.
3	The window is maximized with focus.
4	The window is restored to its most recent size and position. The currently active window remains active.
6	The window is displayed as an icon. The currently active window remains active.

Comments
To have an application run silently, use the value 0 (zero) for the *window_style*.

Example
The following example is the After Integration script for an integration. It executes the PostProc.bat file, which performs some post processing actions.

```
Execute "C:\Program Files\Microsoft Dynamics\Integration
Manager\PostProc.bat", 60,0
```

GetVariable function

The **GetVariable** function retrieves the value of a global variable that was set by the SetVariable function.

Syntax
GetVariable(*variable***)**

Parameters
variable—A string parameter specifying to the name of the variable to retrieve.

Return value
A variant containing the value of the variable.

Example
The following example is a Before Query script. It reads the value of the Path variable, which was set in the Before Integration script, and uses the path information to override the location of the source file for the query.

```
Query.OverrideFileLocation = GetVariable("Path")
```

Related items
ClearVariables function
SetVariable function

LogDetail function

The **LogDetail** function writes an activity detail entry to the log file for an integration.

Syntax
LogDetail [*message*, *source*, *status_code*]

Parameters
message—An optional string parameter corresponding to the Log Text item of the activity detail entry written to the log file.

source—An optional string parameter corresponding to the source item of the activity detail entry written to the log file.

status_code—An optional long integer parameter corresponding to the Status Code item of the activity detail entry written to the log file.

Comments
All parameters for this function are optional.

Example
The following example is the Before Integration script for an integration. It prompts the user for his or her name and writes the value to the log.

```
Dim Input
Input = InputBox("Enter your user name.", "User Name")

If Input <> "" then
    LogDetail "User "& Input & " ran the integration."
Else
    CancelIntegration "Didn't supply a user name."
End If
```

LogDocDetail function

The **LogDocDetail** function writes an activity detail entry for the current document to the log file for an integration.

Syntax
LogDocDetail [*message*, *source*, *status_code*, *field_name*, *field_value*]

Parameters
message—An optional string parameter corresponding to the Message item of the document detail entry written to the log file.

source—An optional string parameter corresponding to the source item of the document detail entry written to the log file.

status_code—An optional integer parameter corresponding to the Status item of the document detail entry written to the log file.

field_name—An optional string parameter corresponding to the Field Name item of the document detail entry written to the log file.

field_value—An optional string parameter corresponding to the Field Value item of the document detail entry written to the log file.

Comments
The **LogDocDetail** function will not write information to the log file if the log level for the integration is set to Summary.

All parameters for this function are optional.

Example
The following example is the Before Document Commit script for an integration. It examines the Contact field and logs a message if no contact is supplied.

```
If IsNull(SourceFields("Contact")) Then
    LogDocDetail "No contact supplied."
End If
```

LogDocWarning function

The **LogDocWarning** function writes an activity detail entry for the current document to the log file for an integration and increases the warning count for the integration by one.

Syntax
LogDocWarning [*message*, *source*, *status_code*, *field_name*, *field_value*]

Parameters
message—An optional string parameter corresponding to the Message item of the document detail entry written to the log file.

source—An optional string parameter corresponding to the source item of the document detail entry written to the log file.

status_code—An optional integer parameter corresponding to the Status item of the document detail entry written to the log file.

field_name—An optional string parameter corresponding to the Field Name item of the document detail entry written to the log file.

field_value—An optional string parameter corresponding to the Field Value item of the document detail entry written to the log file.

Comments
All parameters for this function are optional.

Example
The following example is the Before Document Commit script for an integration. It examines the Fax number field and logs a warning if no fax number is available.

```
If IsNull(SourceFields("Fax")) Then
    LogDocWarning "No fax number."
End If
```

LogWarning function

The **LogWarning** function writes an activity detail entry to the log file for an integration and increases the warning count for the integration by one.

Syntax
LogWarning [*message*, *source*, *status_code*]

Parameters

message—An optional string parameter corresponding to the Log Text item of the activity detail entry written to the log file.

source—An optional string parameter corresponding to the source item of the activity detail entry written to the log file.

status_code—An optional long integer parameter corresponding to the Status Code item of the activity detail entry written to the log file.

Comments

All parameters for this function are optional.

Example

The following example is the Before Query script for an integration. It overrides the criteria for the query. The warning indicates that the criteria were overridden.

```
query.OverrideCriteria = "State = 'ND'"
LogWarning "Overriding query criteria", query.QueryName, 1000
```

PlaySound function

The **PlaySound** function plays a Windows .wav file.

Syntax

PlaySound *path*

Parameters

path—A string parameter specifying the complete path to the .wav file to be played.

Comments

All parameters for this function are optional.

Example

The following example is the After Integration script for an integration. It plays the Msremind.wav file to indicate that the integration is complete.

```
PlaySound "c:\windows\system\Msremind.wav"
```

SetVariable function

The **SetVariable** function sets the value of a global variable that can be accessed by any script in the integration.

Syntax

SetVariable *variable*, *value*

Parameters

variable—A string parameter containing the name of the variable to set.

value—A variant containing the value that the variable will be set to.

Example

The following example is the Before Integration script for an integration. It reads a path value from the path.txt file, and uses that value to set the value of the Path variable.

```
'Read path information from the path.txt file.

Const ForReading = 1
Dim fso, f
Set fso = CreateObject("Scripting.FileSystemObject")
Set f = fso.OpenTextFile("C:\Program Files\Microsoft Dynamics\Integration
Manager\path.txt", ForReading, True)
FilePath = f.ReadLine
SetVariable "Path", FilePath
```

Related items

ClearVariables function
GetVariable function

Glossary

Adapter
An Integration Manager component that connects to a destination or source application.

Advanced ODBC query
A query that issues a SQL statement to retrieve information from an ODBC data source. *See also* Simple ODBC query.

Boolean
The logical value of **True** or **False**.

Comma-delimited file
A text file that uses commas to separate the individual data items in the text file.

Collections
See Recordset.

.CSV file
An acronym for Comma-Separated Values. It indicates a text file that uses commas to separate the individual data items.

Currency
A data type that is intended to hold monetary values. It can have up to four decimal places and must be in the range –922,337,203,685,477.5808 to 922,337,203,685,477.5807.

Data source
An ODBC data source from which you retrieve data.

Data type
A data source setting that indicates what type of data is contained in the column of a data source. Common data types include booleans, currencies, integers and strings.

Delimiter
A character or characters that separate the individual data items in a text file. Commas and tab characters are often used as delimiters.

Destination
Where data gathered by Integration Manager is placed in Microsoft Dynamics GP. Integration Manager provides several common destinations, such as customer information or receivables transactions.

Destination adapter
A feature that validates data before integrating it to the destination application or database such as Microsoft Dynamics GP.

Destination mapping
Where information for each item in the integration destination originate. For many items in the destination, the destination mapping indicates that information originates from a query. For other items, the mapping indicates that a constant value or a default value from Microsoft Dynamics GP should be used.

Document definition
The metadata that Integration Manager uses to describe the structure and content of a source or destination. It describes recordsets, hierarchical relationships, fields, data types, field lengths, and more. It is analogous to an XML schema, but it usually contains more information than an XML schema.

Double
A data type that stores a double-precision floating point number. The value can have up to fifteen significant digits.

Negative values must be in the range –1.79769313486232E308 to –4.94065645841247E–324.

Positive values must be in the range 4.94065645841247E–324 to 1.79769313486232E308.

eConnect destination adapter
An Integration Manager destination adapter that enables integration with Microsoft Dynamics GP data using eConnect.

Enumeration
A data type that is restricted to a fixed set of named values. Enumeration fields in a destination correspond to list boxes, drop-down lists and other list controls in Microsoft Dynamics GP. When you set the value of an enumeration field, you supply the integer value that corresponds to one of the items in the enumeration. *See also* Translation.

Filter
Specifies the criteria for determining precisely which documents to extract from the source. You can define filters only for those fields located in the Root Recordset.

Integer
A data type that stores integral numeric values. It must be in the range –32,768 to 32,767. A set of integrations that are performed in succession in a specified order.

Integration Manager engine
An Integration Manager component that receives data from the source adapter, provides mapping and transformation functionality, and passes data to a destination adapter.

Join
A database operation that combines some or all records from two or more tables.

Long integer
A data type that stores integral numeric values. It must be in the range –2,147,483,648 to 2,147,483,647.

LongVarChar
A data type that stores a sequence of up to 65,535 characters.

Mixed element
Those elements that contain child elements to attributes as well as data.

Null
A keyword that indicates a field or variable does not contain valid data.

Numeric
A data type specific to Integration Manager. It stores decimal values that can have up to 19 digits with up to 5 of the digits after the decimal point.

Query
A request for information. In Integration Manager, a query can request information from text files or from ODBC data sources.

Query builder
A tool in Integration Manager to aid writing a SQL statement to use for an advanced ODBC query.

Query relationship
A relationship between two queries that defines how they work together to retrieve information.

Recordset
An element that is used to map items in an integration destination. There are two types of recordsets. One type of recordset simply groups related fields in the destination. The other type of recordset indicates that several sets of fields in the recordset can be associated with a single instance of a record imported into the destination. These recordsets are represented by the folder icon.

Rejection file
A text file that contains records that were rejected from text queries by Integration Manager. Rejection files have the .rjt extension.

Restriction
A set of criteria that allows you to specify the rows that will be included in a query. All rows that do not fit the criteria are excluded.

Rule
Defines where the information for an item in the destination mapping originate.

Simple ODBC query
A query that retrieves data from an ODBC data source. *See also* Advanced ODBC query.

Single
A data type that stores a single-precision floating point number. The value can have up to seven significant digits.

Negative values must be in the range –3.402823E38 to –1.401298E–45.

Positive values must be in the range 1.401298E–45 to 3.402823E38.

Source
Indicates where the data to be integrated comes from. A source can either be a text file, a database, or an application

Source adapter
A feature that connects to a database, text file or application source. It filters and extracts the data from the source before passing the information to the Integration Manager engine.

Source data
Shows data from the source one document at a time and in the structure of the document definition.

Source name
The name of the source document definition that you are setting up. This name appears in the Add Source window. You may use any name that helps you easily identify this source document definition.

Source settings
Source settings connect the source document definition to an actual source by having you specify certain parameters. They are additional properties relating to a source and are associated with an individual integration. They are not, however, automatically inherited by other integrations that use the same source document definition.

String
A data type that stores a sequence of up to 255 characters.

Tab-delimited file
A text file that uses tab characters to separate the individual data items in the text file.

Text query
A query that retrieves data directly from text files.

Translation
Allows you to define a relationship between values in the source file and corresponding values that are used for the destination field.

VBScript
A subset of the Microsoft Visual Basic programming language that is embedded into Integration Manager to provide scripting capabilities.

XML
Extensible Markup Language (XML) is the standard method of viewing data on the Web. Rich, structured data from any application can be easily described in a standard and consistent manner through the use of XML. It also is a complementary format of HTML.

XSLT
eXtensible Stylesheet Language for Transformations (XSLT) is used as a part of XSL, which functions as a stylesheet language for XML. XSL includes an XML vocabulary for specifying formatting; XSL specifies the styling of an XML document by using XSLT to describe how the document is transformed into another XML document that uses the formatting vocabulary.

Index

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

X