Email2DB V3

Advanced Email & Web Automation

# Email2DB Version 3.2

# User Guide

PARKER Software

# Email2DB



Last Updated: February 2014

**Publisher**

*Parker Software*

**Managing Editor**

*Stephen Parker*

**Technical Editors**

*Daniel Tallentire*
*James Horton*

# Table of Contents

# Part

# 1 Welcome To Email2DB

Email2DB V3

Advanced Email & Web Automation

Version 3.2

On-Line Help

PARKER Software

Engaging Applications · Engaging Results

www.email2db.com

# Part

# 2 Company Information

Parker Software is a focused software house with offices in the UK and USA. Established in the UK in 2003, Parker Software develop innovative software aimed at companies that do business on the web. Parker Software produce two flagship products: "WhosOn", a web site visitor monitoring and live chat solution and "Email2DB", a message automation solution. Parker Software's products are used by thousands of businesses, large and small, in all parts of the world.

You can contact us in any of the following ways:

| | |
|---|---|
| Address | **UK**<br>Parker Software Ltd<br>Victoria Business Park<br>Prospect Way<br>Stoke on Trent<br>ST8 7PL<br><br>**USA**<br>Parker Software Inc<br>4767 New Broad Street<br>Baldwin Park<br>Orlando FL<br>32814 |
| Telephone From USA | 1 800 680 7712 |
| Telephone From Outside UK | 00 44 1782 822577 |
| Telephone From Inside UK | 01782 822577 |
| Fax | 00 44 1782 839682 |
| Email Support | Email2DB Support <support@parker-software.com> |
| WhosOn product info & support: | http://www.whoson.com |
| Email2DB product info & support: | http://www.email2db.com |
| Email2DB Professional Services contact: | Sharlene Baker <sharlene@parker-software.com> |

## Product Updates

You can purchase annual updates which gives you access to major product releases and services packs free of charge. Users to purchased Email2DB via subscription are entitled to free minor and major upgrades for the life of the subscription.

## Extended Support

All users have access to email support and use of the support forum. You can purchase extended support which gives you the following additional benefits:

- Access to a named support technician via telephone or live chat from 8:30am to 8:30pm Monday - Friday (GMT).
- Access to live chat support directly from the Email2DB Administrator.
- Ability to submit a support incident via the Email2DB Administrator.
- Priority support incident submission.
- 24 hour response time (weekdays).

## 24/7 Support

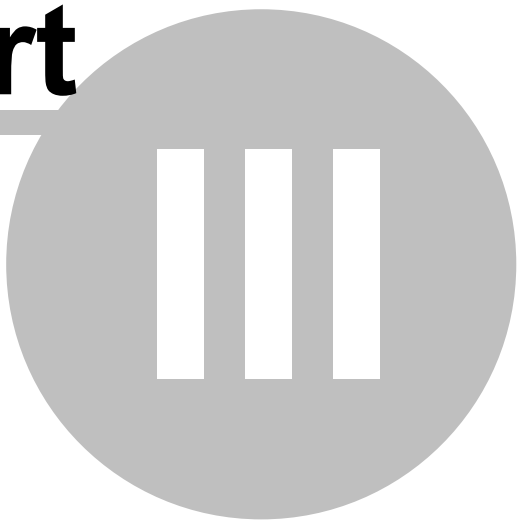
For larger corporations with mission critical Email2DB installations we also offer 24/7 extended support with a 60 minute response time. Please contact us for details & pricing.

## Professional Services

We provide Professional Services for clients who need assistance with the setup, training and configuration of Email2DB. Please contact the professional services team for a quotation or see: http://www.email2db.com/services.aspx

PARKERPROServices
Bespoke Software Solutions for WhosOn & Email2DB

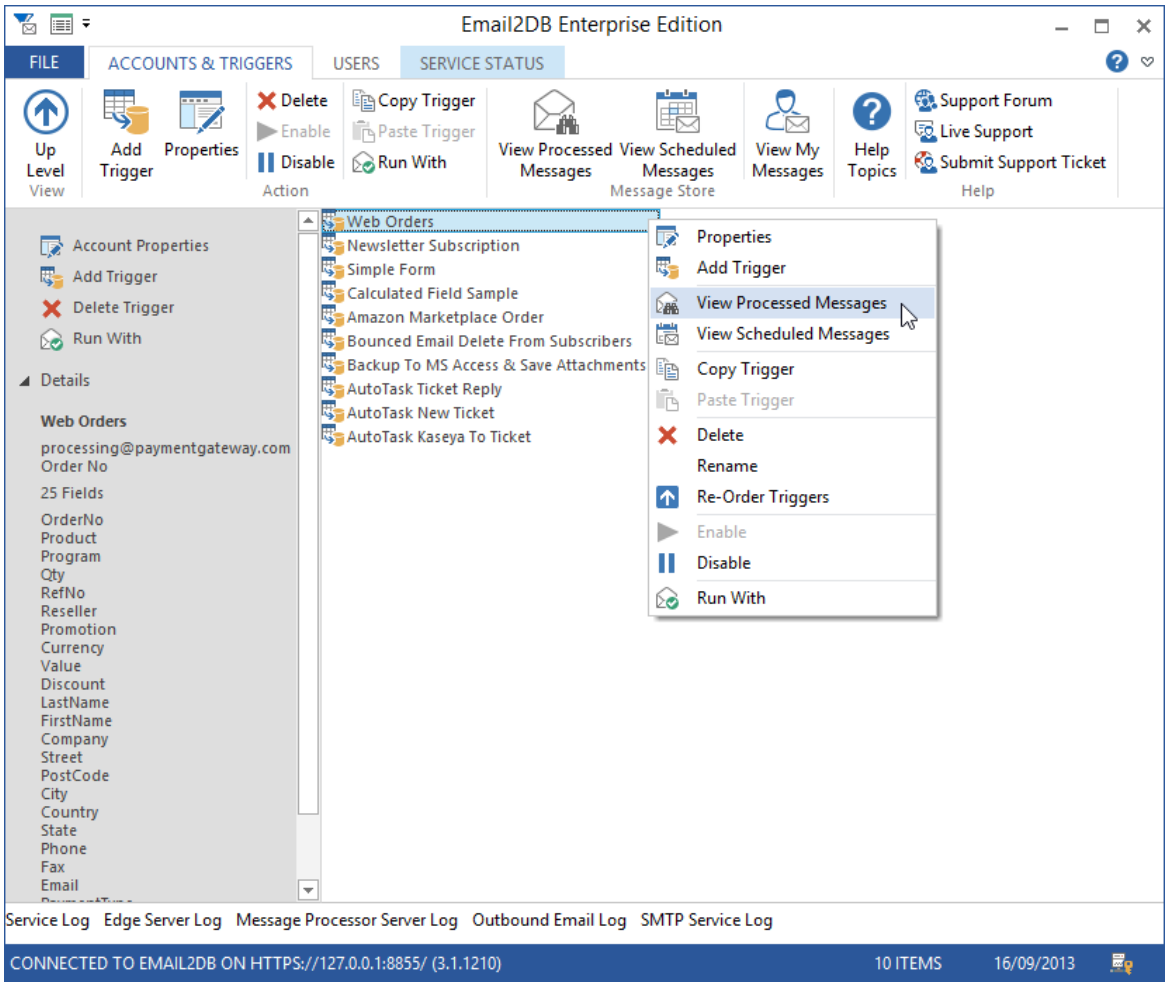
# Part



# 3 Introduction

## Welcome To Email2DB Version 3.2

Email2DB is a message parser & message automation solution. Use Email2DB to read Email Messages, Database Records, Twitter Feeds, Web Pages, RSS Feeds and other sorts of messages from multiple sources - then extract useful information from these messages to update your databases and perform multiple automated actions. Automate the processes that need to occur when your business receives messages and dynamically respond to things happening on the web.




If you receive any sort of data by email or via the Web, such as orders, customer requests, web form results, or any other regular responses you can use Email2DB to convert them into real data, and at the same time perform other tasks, such as forwarding the email, sending new emails, printing a report, posting to web pages, posting to social networks, sending faxes & much more.

Email2DB is a message 'parser'. It scans messages and picks out (parses) data that it matches to fields in your database. After extracting the field data it builds SQL statements to send to your database.

So you could use Email2DB to receive customer orders from your web site, then automatically update your Orders database in your accounting system - then print a report, make a sound (so you know a new order has arrived), forward the email to someone else, post to Twitter, send a confirmation to the customer, schedule an automated email follow up in the future and much more.

Email2DB can process more than just email messages. It can retrieve messages from web pages,

Twitter & RSS feeds, database records and files on your network. Email2DB can even act as a fully featured email server, allowing it to process email as it comes in (or out). It can also receive messages from multiple remote users on your network or across the Internet.

Once setup, Email2DB is designed to run uninterrupted and in the background. It can run on the company server, or on a workstation. The Email2DB Data Center Edition can run on multiple servers (nodes) within your organization with each node handling message retrieval or message processing. You can add more nodes as your processing needs grow.

See Also: Quick Start
See Also: What's New In This Release

## 3.1 Product Editions

Email2DB is available in a number of editions: **Small Business**, **Enterprise**, **Data Center, Hosted & Developer**.

All editions include the same features for parsing and extracting data from emails to update databases. All editions allow any number of Accounts and Triggers to be processed. All editions allow emails to be retrieved and processed from POP3, IMAP, Exchange Server sources.

### Enterprise Edition

The Enterprise Edition provides more Action types. Click here to view the Action types available with each edition. The Enterprise Edition includes a built-in mail server for receiving emails directly. Messages received via the built-in mail server are processed immediately. The Enterprise Edition also enables messages to be retrieved from Database Sources (Database Pull), Web Pages & RSS & Twitter Feeds.

### Data Center Edition

The Data Center (DC) Edition provides the same level of functionality as the Enterprise Edition but with the added ability that the Edge & Message Servers can be run on separate physical computers providing a distributed processing environment. You purchase a number of 'nodes'. Each node equates to an Edge or Message Server - you then deploy these depending on your processing and redundancy requirements.

### Hosted Edition

The Hosted Edition of Email2DB provides most of the functionality of the Enterprise Edition, however the Email2DB Server is run and managed for you on one of Parker Software's Cloud based servers. This provides all the features and benefits of Email2DB without the hassle of setting up, maintaining & backing up of a server.

### Developer Edition

The Developer Edition allows developers to create and test Email2DB Accounts & Triggers before deployment to production servers. It provides the full Enterprise Edition functionality but with a restriction on the number of messages that can be processed in a single day.

# 3.2 What's New In This Release

## Version 3.2

Email2DB Version 3.2 adds the following new features over version 3.1.

**Improved Logging**
Version 3.2 changes the way Email2DB Message Processing logs are generated, stored and viewed. This provides a significant improvement in performance when the Email2DB server is under heavy load.
Logs are now stored in a new MessageLog table in the Message Store database. Previously logs were held in separate files.

**Log Viewing**
A number of improvements have been made to log viewing via the Email2DB Administrator:
- The Node Name will be shown if multiple message processor nodes are in use (when using the Data Center Edition).
- The Account name is shown.
- Search for text whilst viewing logs.
- View logs for previous days.
- Lock the scroll position to prevent new log lines scrolling the display whilst viewing.
- View the Message in the Message Store for the currently selected log line.

**New Actions**
- Exchange Mark Message As Junk.
- Exchange Create Lync Folder.
- Exchange Move Contact To Lync Folder.
- Improved Update CSV Action.
- GeoIP Lookup Action.

## Version 3.1

Email2DB Version 3.1 adds the following new features over version 3.0.

**Action Snippets**
Blocks of Actions can now be saved as 'Snippets'. These can be re-used on other Triggers. You can create a Snippet whilst editing Actions. This creates a new Tab in the Actions list. You can then copy/ paste Actions to it or edit it as normal. When you create a Snippet this automatically as a new action 'Include' to your current trigger. The Include Action is used to call the Snippet. Once a Snippet is saved it will appear as an action in the Toolbox to be used on other Triggers. If you edit a Snippet – any changes will take effect on any Trigger that includes it.

**Trigger Revision Saving**
A new option has been added to main program options 'Keep A Copy Of All Trigger Edits'. If enabled then all changes made to Triggers will be stored. A new 'Revert' menu option on the Trigger ribbon allows you to revert to any previous version of the Trigger.

**Multiple Action Selecting**
When editing Trigger Actions you can now select multiple lines at the same time for Delete, Copy, Cut & Move Up/Down.

**New Actions**
- Execute A Database Command - allows you to execute a command or stored procedure and assign the results to variables.
- Lookup From A Database - allows you to lookup a record from a database and assign multiple database fields to Email2DB fields or variables.
- Post To LinkedIn - allows automation of LinkedIn Shares.

- Create Outlook MSG File - saves the current message or custom message as an Outlook compatible MSG file.
- Send TCP Data - opens any TCP socket, sends custom data and optional wait for a response.
- File Operations - create folders, copy, move, rename & delete files.
- SkyDrive Upload/Download.
- Google Drive Upload/Download.
- Download File - download a file via HTTP from any URL.
- Translate Text - translate text from one language to another and assign the results to variables.
- Speak Text - speak text in a given language. Return the file URL to a variable.
- Detect Language - returns the language code for any text.

**Other Changes**

- Condition builder can now use fields %hour% and %minute% allowing you to create time-based conditions.
- Improved Regular Expression handing - now compatible with full Microsoft RegEx implementation.
- Improved Performance of Message Processor when actions generated large logs.
- New Office 2013 Theme and updated Icons.
- Updated Twitter parsing & posting using the new Twitter API.

## Version 3.0

Email2DB Version 3 adds the following new features over version 2.x.

### Architecture

Email2DB Version 3 has been rewritten from the ground up to provide a more scalable & secure architecture. Two new services have been created:

**1. The Edge Server**

The Edge Server handles the retrieval of messages via IMAP, POP3 ,Exchange, Database Pull, Twitter, RSS & HTTP sources. The Edge Server runs the Trigger Condition checker on each new message and assigns it a Trigger ID. It then places the message in a new database table called the 'Edge Queue'.

**2. The Message Processor Server**

The Message Processor Server reads new messages from the Edge Queue and executes all Trigger Actions for the Trigger that the message has been assigned to. The message is then stored in the Email2DB Message Store.

In Email2DB Version 2 the above processes were handled by a single server. The advantage of this new architecture is that the Edge Server & Message Processor Servers execute concurrently. In addition, with the new Data Center Edition the Edge & Message Processor Servers can be run on separate physical computers and you can run multiple instances to share the processing load.

### Security

Starting with Email2DB Version 3, the Email2DB Administrator no longer updates the Account & Trigger settings (the 'MetaData') directly - even when run on the same computer. It reads/posts MetaData XML via the Email2DB Server using secure HTTPS. This makes editing Accounts & Triggers faster - regardless of the number of Accounts & Triggers you have setup. All passwords are now stored in the MetaData encrypted. The MetaData can now be stored in a SQL Server database in addition to Email2DB's own database (SQLite).

### Triggers

- Trigger actions are now executed in the order you add them - in Version2 Trigger Actions executed in a fixed order.
- Actions can be moved up and down the execution order.
- Multiple actions of the same type can be added.
- Each Action can now have its own Conditional Execution rule.
- If, Else, End If blocks can be created allowing for conditional execution of blocks of Actions.
- Easy creation of Actions via drag/drop interface.
- Visual view of Action execution order eases Trigger development and increases productivity.

### New Action Types (Dependent On Edition)

- Update Azure SQL Databases, Table Entities & Blobs.
- Update Excel Spreadsheet.
- Update/Read/Upload to SharePoint sites.
- Update/Read/Upload to Microsoft Dynamics or Salesforce CRM systems.
- Upload to Amazon S3.
- Upload to Dropbox.
- Send iCalendar Appointments.
- Wait For Manual Validation.
- Update Exchange Server (2007/2010/2013 & Office 365) Contacts.
- Update Exchange Server Appointments.
- Update Exchange Server Tasks & Notes.
- Set & Get Exchange Out Of Office Status for any user.
- Move Message To Different Exchange folder.
- Add Message To Exchange.
- Read A Web Page.
- DNS Lookup.
- Update WebDav Resource.
- Upload To FTP Server.
- Post To A Jabber Server.
- Post To Twitter.
- Render messages as PDF files.
- Set Variable.
- Read/Write Text File.
- Logical If, Else, End If, Go To, Label, For Each Loop, & End.
- Comment.
- SMS Sending can now use SNPP Servers in addition to the Email2DB SMS Gateway.

See Also: Available Action Types

### Accounts

- Can now read messages via Exchange Web Services (EWS).
- Can now accept messages from multiple remote users running the Email2DB Client.
- Atom feeds can be read in addition to RSS.
- Can now read tweets from a Twitter Account using any search term.
- File pickup supports Outlook MSG files.
- CSV & Excel Spreadsheet File Pickup.
- Outlook PST Export Wizard.

### Administrator

- Tabbed editing of multiple Accounts & Triggers at the same time.
- Drag & Drop toolbox interface for Trigger Action creation.
- Visual view of Trigger Action execution sequence.
- Faster loading & editing when large numbers of Accounts & Triggers are being used.
- Secure HTTPS connection to the Email2DB Server.
- Administrator can be used on remote computers on all editions.

## Server

- New 'Edge Server' & 'Message Processor' Servers split the reading & processing of messages into two concurrently running processes - improving overall performance and reliability.
- Edge & Message Processor Servers can run on separate physical computers (Data Center Edition).
- Multiple Edge & Message Processors can run at the same time for maximum scalability.
- Can now use SQL Server to store the Email2DB Accounts & Trigger Settings (MetaData).
- Built-In Message Store Database now uses SQLite.
- Outgoing emails are now sent via the Email2DB Mail Server allowing for better performance, failure retries and non-delivery processing.
- New Web Services included for easy addition of messages to be processed via 3rd party applications.
- Message Store can now use SQL Server 2012.

## New Email2DB Client Application

- A new application that can run on multiple networked workstations or over the Internet.
- Connects to the Email2DB Server via secure HTTPS.
- Allows users to drag and drop messages onto specific Email2DB Accounts & Triggers.
- Outlook messages can be dragged and dropped directly onto the Email2DB Client for processing.
- Messages are then sent to the Email2DB Server for immediate processing.
- Can integrate with the user's local Outlook Client allowing multiple Outlook Clients to send messages to a single Email2DB Server.

## Data Center Edition

- Aimed at larger businesses or businesses with high message volume.
- Allows distributed processing and redundancy.
- Multiple 'nodes' can be configured on separate computers.
- Each node can be configured to read incoming messages, process messages or act as a mail server.
- Multiple message processor servers can be activated at any time to share message processing load.

## Hosted Edition

- Now available as a hosted service.
- Provides all the features of the Email2DB Enterprise edition.
- Secure message storage using Microsoft Azure geo-redundant containers.

The product Edition names have been changed starting with Version 3. The 'Standard' edition in Version 2 has been discontinued. The Version 2 'Professional' Edition is now renamed 'Small Business' Edition.

## 3.3 Email2DB Hosted Edition


In addition to the regular Email2DB Installable Edition we also provide a hosted cloud-based edition.

It provides most of the features of the Email2DB Enterprise edition but with the added benefit that you do not need to setup and maintain the Email2DB Server components. Your Email2DB Cloud server is pre-configured to receive messages directly via your own Email2DB email addresses. It can also read messages from any IMAP/POP3 or Exchange/Office 365 accounts. You can access & configure your Email2DB Accounts using the Email2DB Administrator from any PC with Internet access.

Email2DB Cloud Edition is built on the Microsoft Azure Cloud platform providing a secure, robust and geo-redundant platform for automating & storing your business messages.

The Email2DB Cloud Edition is priced using a subscription model which includes all support & product updates. Plans are available for small & large message volume requirements.

This manual is used for both Email2DB Installable & Email2DB Cloud Administration. For Email2DB Cloud Edition users you do not need to configure the server or mail server options as these will have been pre-configured for you.

See Also: Available Action Types

# Part

# 4 Getting Started - Email2DB Installable

Email2DB is a combined message parser, message automation server and email server. Use Email2DB to read Email Messages, Database Records, Twitter Feeds, Web Pages, RSS Feeds and other sorts of messages from multiple sources - then extract useful information from these messages to update your databases and perform multiple automated actions. Automate the processes that need to occur when your business receives incoming messages and dynamically respond to things happening on the web.

The Email2DB Server is a Windows application. It can run on any Windows computer.

System Requirements:
- Windows 7, 8, Vista, 2003 Server, 2008 Server (32 or 64 bit), 2012 Server.
- .NET Framework 3.5
- IIS with ASP.NET enabled (only required if you will be running the Email2DB Administrator or Client on remote computers).
- 500 MB free disk space.

To install Email2DB run the Email2DB31.exe setup program. You need to be logged on with full Administrator rights during the setup. If using Windows 7/8, ensure you right click the setup exe and select 'Run As Administrator'. If using Windows 8 or 2012 you may need to enable .NET Framework 3.5 in your Control Panel - Add/Remove Windows Features. If you want to run the Email2DB Administrator or Client on remote computers then you will also need to enable IIS and ASP.NET. This allows the Email2DB Web Services to be used.

The setup will create an 'Email2DB Version 3' folder on your start menu.

The setup will also install the Email2DB Server Services. This will be set to start automatically when your PC starts up. It will be removed if you un-install Email2DB.

When you start the Email2DB Administrator for the first time the Registration Wizard will start. Run through this to register your copy. If you are evaluating Email2DB and don't have a serial number, click the 'Get A 30 Day Evaluation Serial Number' button to have a temporary serial number sent to you.

**To Uninstall**

Open the "Programs and Features" applet from your Windows Control Panel. Select Email2DB from the list and click the uninstall button.

See Also: Quick Start

# 4.1 Quick Start

This chapter will help you to get started with Email2DB as quickly as possible.

Email2DB is a combined message parser, message automation server and email server. Use Email2DB to read Email Messages, Database Records, Twitter Feeds, Web Pages, RSS Feeds and other sorts of messages from multiple sources - then extract useful information from these messages to update your databases and perform multiple automated actions. Automate the processes that need to occur when your business receives incoming messages and dynamically respond to things happening on the web.

Email2DB consists of two key processes:

**1 - The Email2DB Server Services**

The Email2DB Server runs as a collection of system services on your computer. These services handle the reading and processing of messages. The services will automatically start each time your computer starts, so they will always be running in the background downloading and processing your messages.

**2 - The Email2DB Administrator**

This program is used to configure Email2DB settings. It does not need to be left running for messages to be processed. The Email2DB Administrator can be run on the same computer as the Email2DB Services or on remote computers. It connects to the Email2DB Server via a secure HTTPS connection.

To start the Email2DB Administrator select **Email2DB Administrator** from the **Email2DB Version 3** folder on your Start menu.

## How It Works

Email2DB can read messages from multiple sources (POP3 Servers, IMAP Servers, Exchange Servers, Twitter & RSS Feeds). The Enterprise Edition can also receive messages directly via it's own built-in e mail server and read messages from Web Pages and external database records.

You can create any number of 'Triggers' that are checked against these incoming messages. A Trigger is a set of conditions that Email2DB will check for (for example, a specific 'from' address, or specific words in the subject line). If a message passes the Trigger conditions, Email2DB will process a series of 'Actions' against it. These actions include the updating of a database, sending email responses, printing reports and many more.

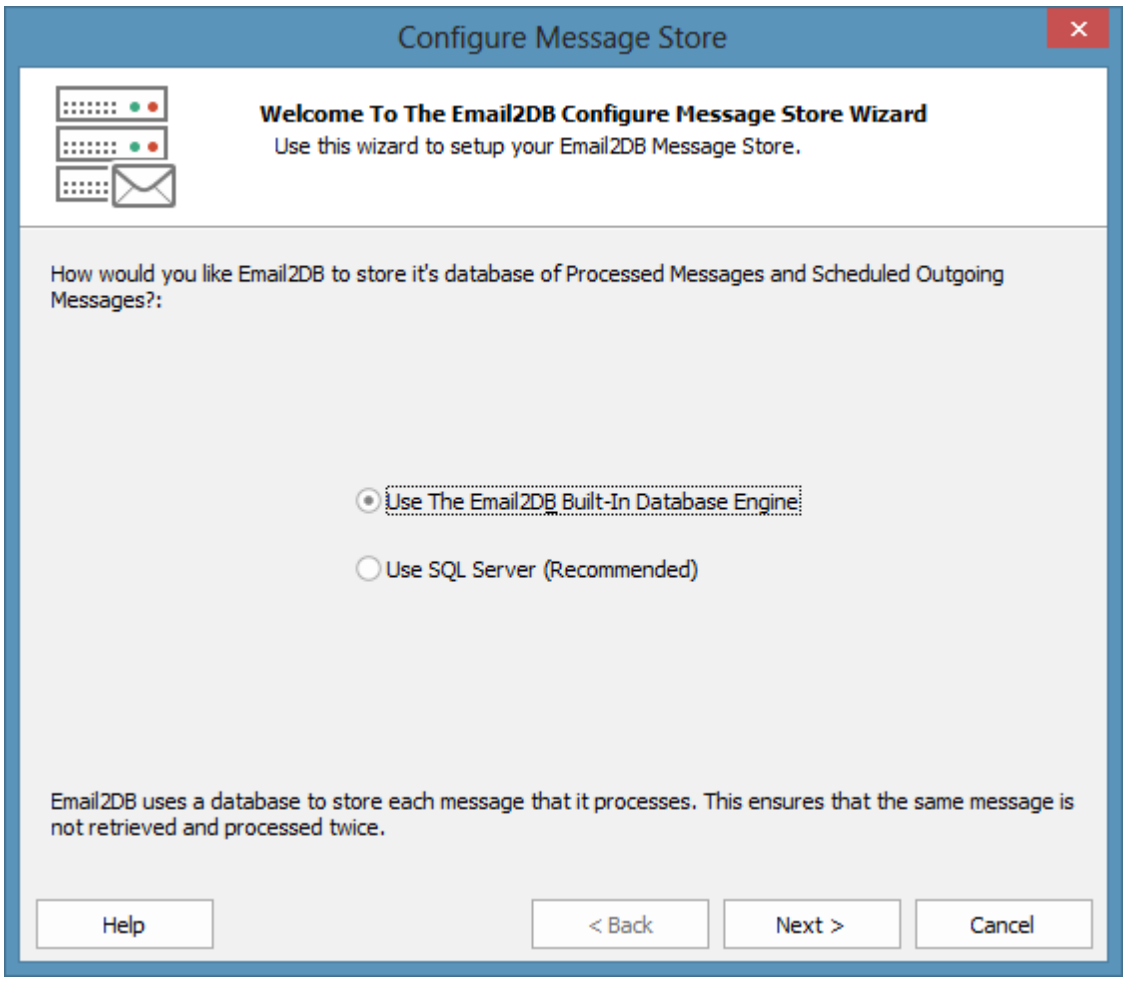
After Email2DB has processed an incoming message it stores a copy of the message in its own database (called the 'Message Store'). This provides a useful central repository of all messages, and prevents the same message from being checked and processed twice. During development of your triggers, you can reprocess messages held in the Message Store to rerun Actions against a message.

## Installing & Starting Email2DB

Install Email2DB by running the Email2DB30.exe setup program. Ensure you are logged on to your computer with full Administrator rights. Email2DB requires Windows 7, 8, XP, Vista, 2003 or 2008 (32 or 64 bit) operating systems.

When you start the Email2DB Administrator for the first time the **Registration Wizard** will start. Run through this to register your copy. If you are evaluating Email2DB and don't have a serial number, click the 'Get A 30 Day Evaluation Serial Number' button to have to have a temporary serial number sent to you.

When the Registration Wizard completes, the **Configure Message Store** wizard will start:

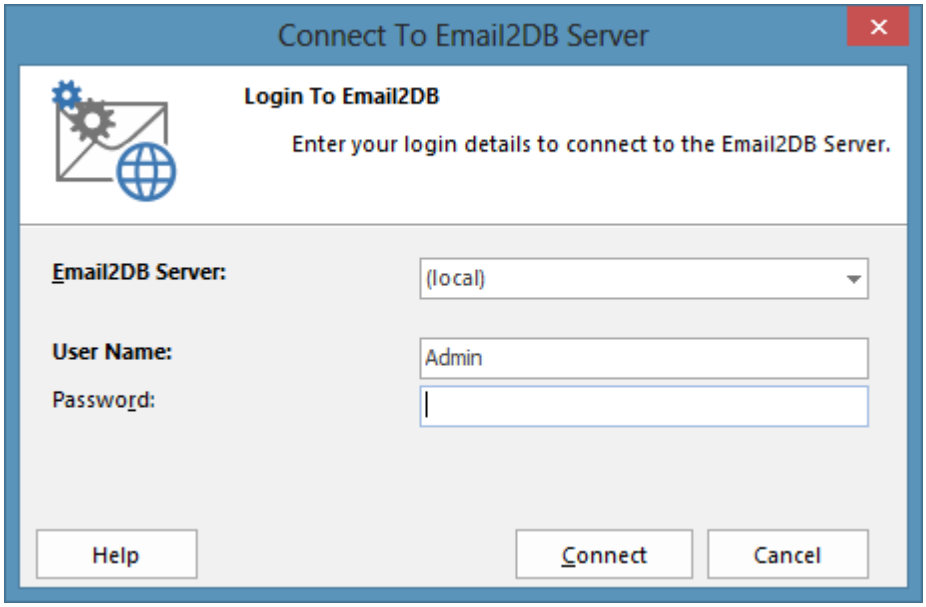



Email2DB stores each message it processes in it's own 'Message Store'. Email2DB can manage this database itself, or you can use an external SQL Server database. If you will be processing large numbers of messages, we recommend using SQL Server 2005 or higher. The best option for this is to use SQL Server 2008/2012 Express and install it on the same computer as Email2DB itself. SQL Server Express is a free download from Microsoft.

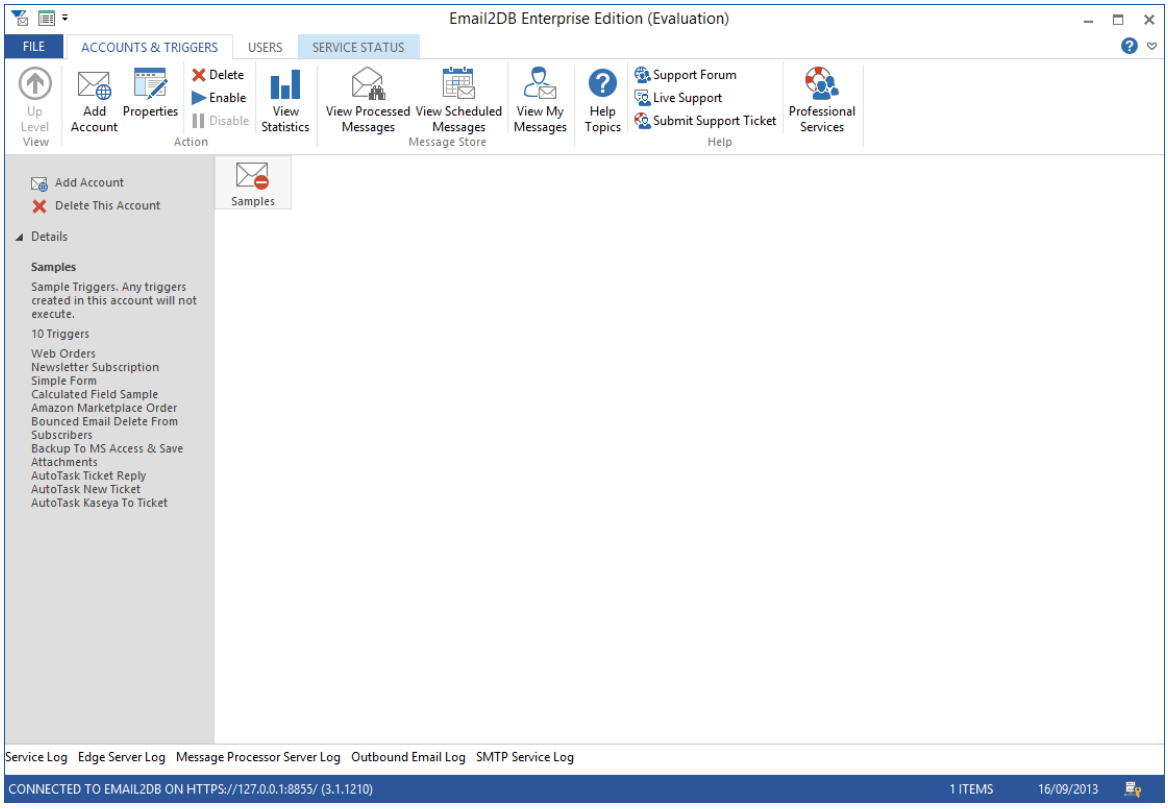
If you have installed Email2DB Version 3 on a machine running Email2DB Version 2 you will now be asked if you want to migrate the settings and message store database.

Once the Configure Message Store wizard completes, the **Email2DB Administrator** will start. When the Administrator first starts it automatically installs and starts the Email2DB Services. These are designed to be left running at all times.

Before you can use the Email2DB Administrator, you need to login. The **Email2DB Server** entry can be left as '(local)' since we are connecting to the Email2DB Server running on the local computer. The default **User Name** is 'Admin' with no password.

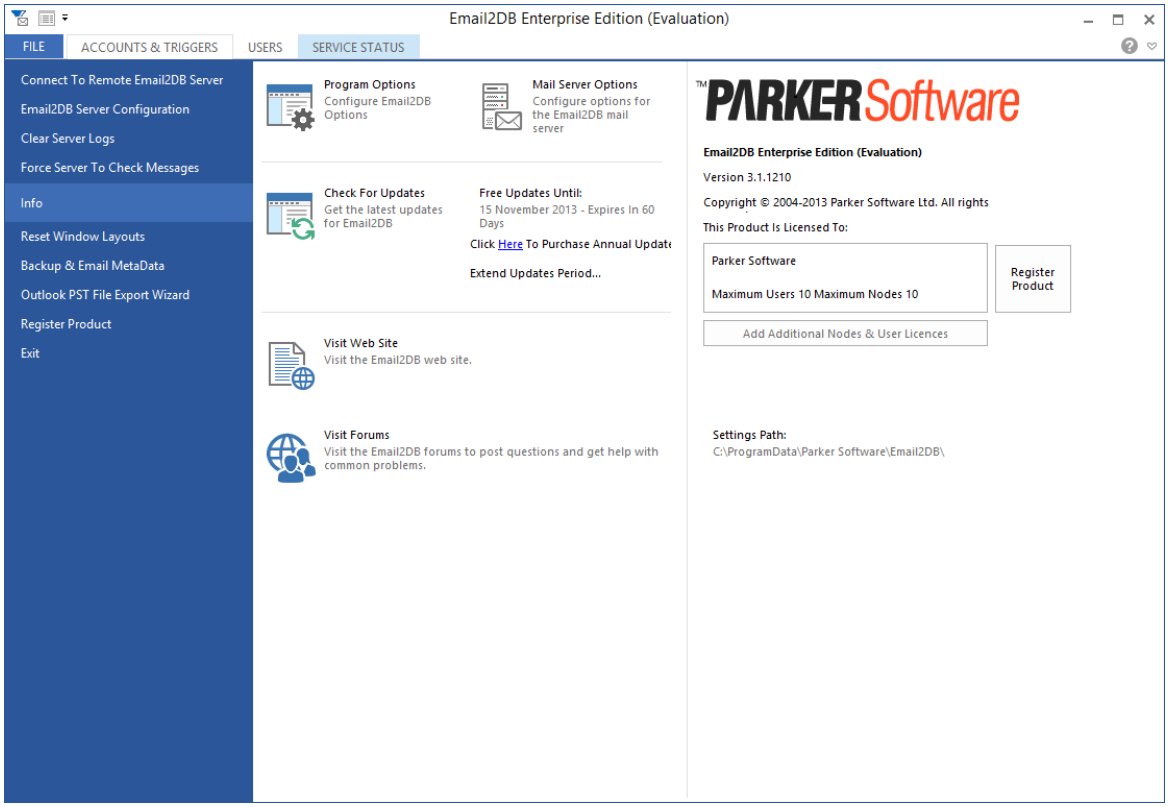



Click **Connect** to login to Email2DB:




## The File Menu

Click **File** tab on the Ribbon bar to display the **File Menu**




The **Program Options** button is used to define global settings. Select **Mail Server Options** if you want to enable the built-in mail server so that Email2DB can accept messages for processing directly (as opposed to 'pulling' messages from other sources).

The **Register** button is used to register your serial number when you purchase Email2DB. You can check for the latest product updates by clicking the **Check For Updates** button.

## Sample Account

By default a 'Samples' account is created. This contains a number of sample triggers. You can open these triggers for tips on how to use Email2DB. You should not add triggers to the samples account as the account is disabled and won't actually do anything.

## Adding An Account

Click the **Add Account** button to create Message Retrieval Account. This tells Email2DB how and where to read your emails (or other types of messages) from. Once you have created an Account, Email2DB will read your messages from the message source and pass each message to one or more 'Triggers'. The triggers define what conditions to apply to the incoming message. If the message matches the trigger conditions then Email2DB can extract information from the message and perform database updates and other actions.

Now See: Creating Message Retrieval Accounts and Message Triggers

## 4.2 Installing Email2DB

Email2DB is a combined message parser, message automation server and email server. Use Email2DB to read Email Messages, Database Records, Twitter Feeds, Web Pages, RSS Feeds and other sorts of messages from multiple sources - then extract useful information from these messages to update your databases and perform multiple automated actions. Automate the processes that need to occur when your business receives incoming messages and dynamically respond to things happening on the web.

The Email2DB Server is a Windows application. It can run on any Windows computer.

System Requirements:
- Windows 7, 8, Vista, 2003 Server, 2008 Server (32 or 64 bit), 2012 Server.
- .NET Framework 3.5
- IIS with ASP.NET enabled (only required if you will be running the Email2DB Administrator or Client on remote computers).
- 500 MB free disk space.

To install Email2DB run the Email2DB31.exe setup program. You need to be logged on with full Administrator rights during the setup. If using Windows 7/8, ensure you right click the setup exe and select 'Run As Administrator'. If using Windows 8 or 2012 you may need to enable .NET Framework 3.5 in your Control Panel - Add/Remove Windows Features. If you want to run the Email2DB Administrator or Client on remote computers then you will also need to enable IIS and ASP.NET. This allows the Email2DB Web Services to be used.

The setup will create an 'Email2DB Version 3' folder on your start menu.

The setup will also install the Email2DB Server Services. This will be set to start automatically when your PC starts up. It will be removed if you un-install Email2DB.

When you start the Email2DB Administrator for the first time the Registration Wizard will start. Run through this to register your copy. If you are evaluating Email2DB and don't have a serial number, click the 'Get A 30 Day Evaluation Serial Number' button to have a temporary serial number sent to you.

**To Uninstall**

Open the "Programs and Features" applet from your Windows Control Panel. Select Email2DB from the list and click the uninstall button.

See Also: Quick Start

## 4.3 Upgrading From Version 2

You can migrate your Email2DB Version 2 Accounts & Triggers and Message Store into Version 3 format. You cannot run Email2DB Version 2 and 3 at the same time on the same computer.

Email2DB Version 2 will be uninstalled when you install Email2DB Version 3 on a machine running Version 2.

When you run Email2DB Version 3 for the first time you will be asked if you want to migrate your Version 2 Settings & Message Store after the **Configure Message Store Wizard** has run.

*IMPORTANT: The Message Store Migration may take some time if your Message Store is large. Please allow for around 30 minutes per GB.*

Your Email2DB Version 2 Settings & Data are left untouched after migration allowing you to un-install Version 3 and re-install Version 2 if required.

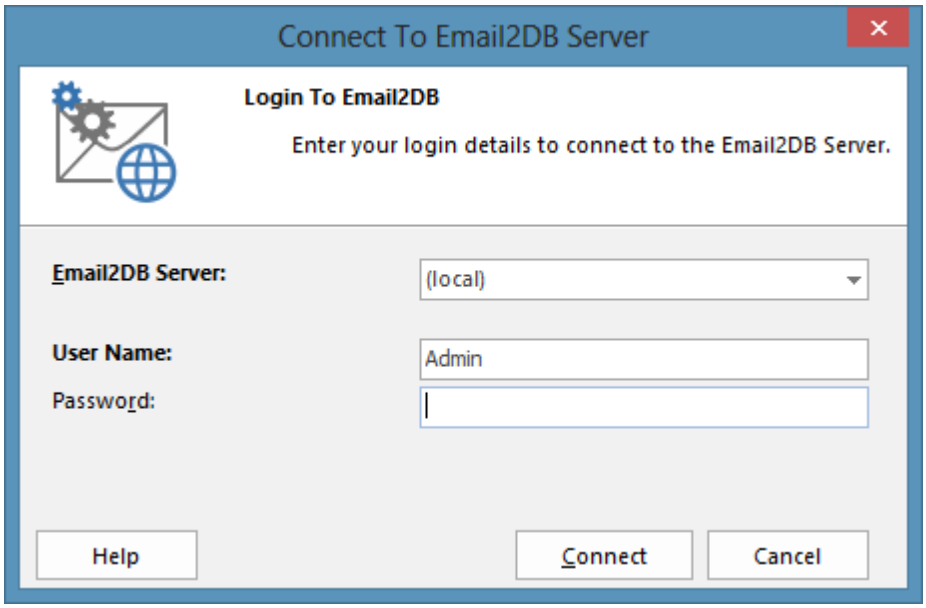
### Important

Accounts will be disabled in Version 3 after they are migrated. This will enable you to verify the migration before re-enabling the Accounts. Accounts that were disabled in Version 2 before the migration will show '(Disabled Before Migration)' in the Account name.

## 4.4 Starting Email2DB

To Start Email2DB, choose **Email2DB Administrator** from your Start menu - **Email2DB Version 3** folder.

Before you can use the Administrator you need to login:




The **Email2DB Server** should be left as (local). This entry is used when connecting to an Email2DB Server that is running on a different computer.

Email2DB creates a default user called 'Admin' with no password. You can change the default user password and create additional users from the **Users** tab within the Email2DB Administrator.

Click the **Connect** button to login.

When the Administrator starts for the first time it will install and start the Email2DB Server services.

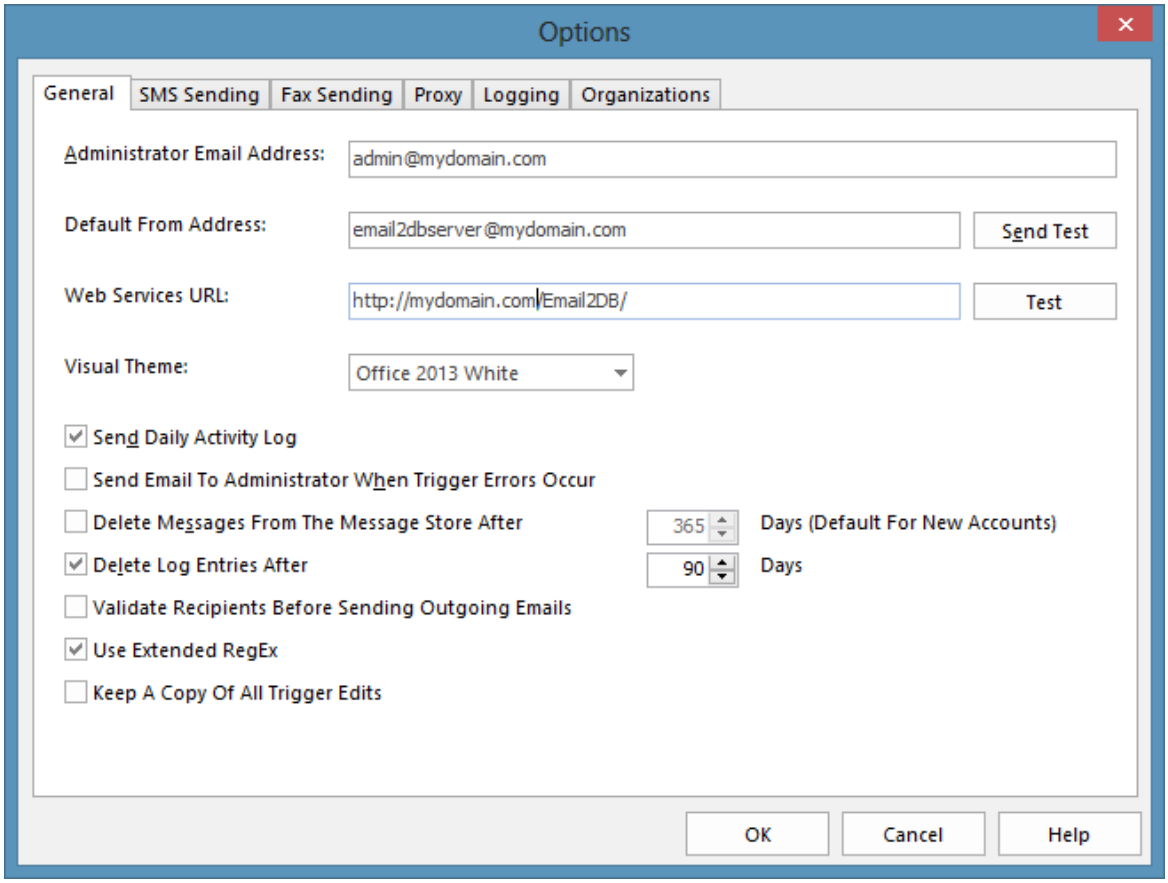
You are now ready to begin setting up Email2DB.

See Also: Creating A Message Retrieval Account
See Also: Message Triggers

## 4.5 Program Options

Click **File** on the Ribbon bar then click the **Program Options** button to access the Email2DB global options.

### General Options




Email2DB can send a daily report to the administrator containing a log of activity for the day. Enter the **Administrator Email Address** to receive this email and check the **Send Daily Activity Log** option.

#### Default From Address

Specify an email address that will be used as the sender of outgoing emails if no specific sender is specified. The **Send Test** button can be used to send a test email to the Administrator Email Address.

#### Web Services URL

The Email2DB Web Services are used by remote Email2DB Client Applications and for Validation Action requests. If you will be using Validation Actions or the Email2DB Client on remote computers then you need to specify the web address of the Email2DB Web Services. The web services are installed during the main setup in a virtual directory called 'Email2DB' under your default web site in IIS. See: The Email2DB Web Services

#### Send Daily Activity Log

If this option is enabled then Email2DB will send an email each day (at midnight). Attached to the email will be the service log entries for that day. The email will be sent to the address specified

above.

**Send Email To Administrator When Trigger Errors Occur**

If this option is enabled then Email2DB will send a report via email to the administrator whenever any trigger actions fail.

**Delete Messages From The Message Store After x Days**

If this option is enabled, Email2DB will automatically delete old checked messages that are older than the number of days specified. Email2DB deletes the old data each day at midnight. This option is useful if Email2DB is processing a large number of emails and you don't need to keep a record of each checked message. This setting provides the default option for new Email2DB Accounts. Each Account has its own setting for Message Deletion. *Note: This option should only be used if you are deleting processed messages from the mail server - otherwise Email2DB could process the same message again once it had been deleted from its checked messages database.*

**Delete Log Entries After**

Email2DB keeps a process log showing each action that is processed against each message. You can view this log when viewing a message in the Message Store. The log can grow quite large - so you should enable this option to remove old log entries. Specify the number of days to keep log entries for.

**Validate Recipients Before Sending**

If this option is selected Email2DB will lookup the recipients MX record before sending Emails. This can reduce the load on the mail server if you expect to have many bad email addresses, and it can help to improve your sender reputation by filtering out bad emails before they get sent.

**Use Extended RegEx**

If this option is enabled, Email2DB will use the Microsoft RegEx library to perform all Find & Extract options when extracting data from your messages. This option is enabled by default. It should only be disabled for backward compatibility with Email2DB Version 2 - if you made heavy use of RegEx using the Email2DB built-in library and your extractions no longer work as expected.

**Keep A Copy Of All Trigger Edits**

Enable this option to switch on Trigger Revision Saving. When enabled Email2DB will store a copy of every revision you make to your Triggers. This allows you to revert to a previous revision. See: Reverting To A Previous Revision

## SMS Sending

Email2DB is capable of sending SMS messages to any mobile device via the Send An SMS Message action. SMS sending is one of the actions that Email2DB can perform as a result of an incoming email. The actual sending of SMS messages is done by our SMS Server Gateway. Before sending SMS messages you need to setup an SMS Server Account and purchase 'credits'.

SMS messages cost between $.12 and $.05 each depending on the country and network of the receiver. The receiver of the SMS is not charged anything. Email2DB can send messages to almost all countries and networks. For full details of coverage and pricing see http://www.email2db.com/ordersms.aspx

Click the **Start The SMS Account Manager** button to start the SMS Server Account Manager. Here you can create a new account and purchase credits and view your current balance.

Once you have setup an account enter the SMS Server Account User Name and Password. Email2DB will use this when it sends SMS messages. Each message sent will reduce your account balance.

## Fax

Email2DB can send outgoing faxes. Use this tab to specify the COMM port number to use. You need to have Fax Modem installed and assigned to the COMM port specified. Click the Search For Fax Modem button and Email2DB will scan your COMM ports and tell you which port to use.

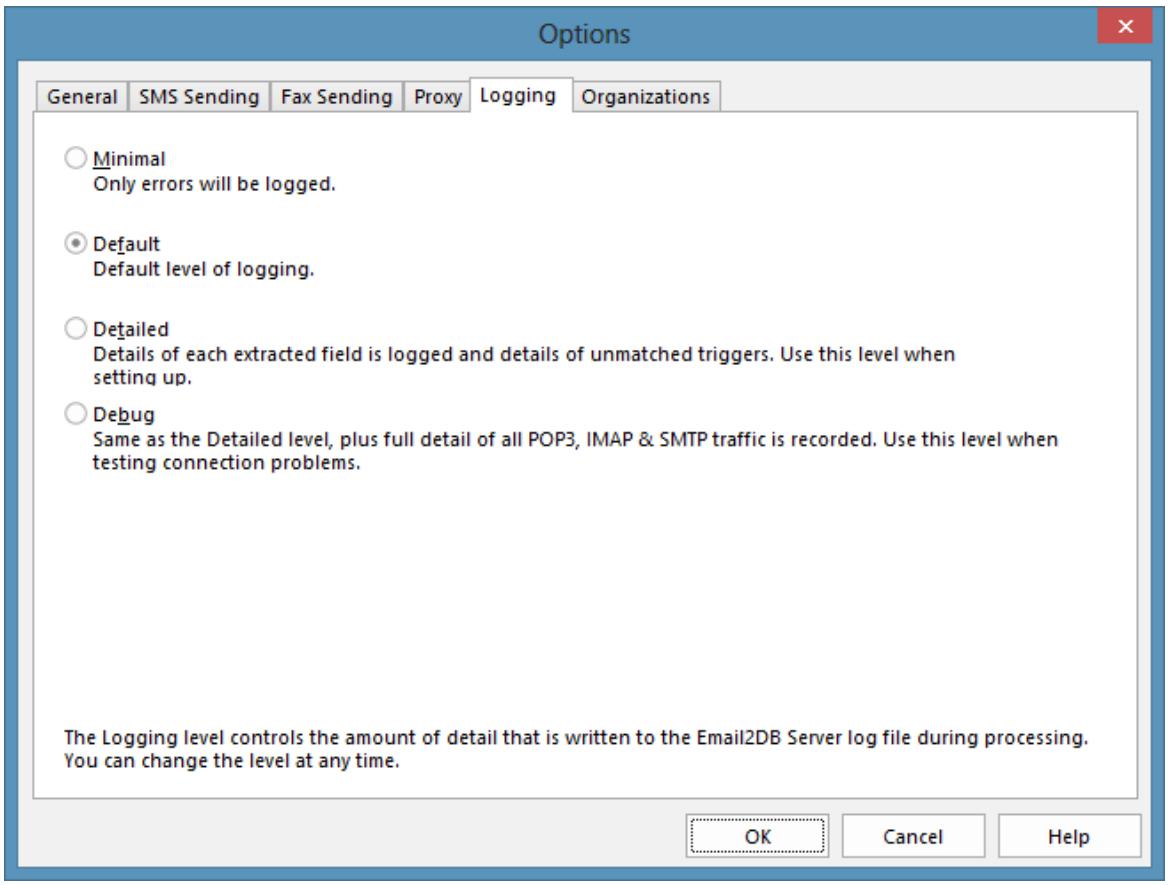
## Proxy Server

If Email2DB is running on a PC that uses a Proxy Server or Firewall device to connect to the Internet you may need to specify the Proxy server type on this tab. Only do this if you are having problems sending outgoing emails.

If a Proxy server type is specified then Email2DB will use it to route ongoing emails to the destination SMTP server instead of connecting directly to it.

## Logging

This tab is used to control the level of detail that is saved to the Email2DB service log. The log can be viewed via the Email2DB Administrator. You can change it at any time.




We recommend that you set the logging level to 'Detailed' whilst you are setting up and testing your triggers. Set it back to 'Default' or 'Minimal' when your trigger settings are finished.

The 'Debug' level logging will also record all POP3/IMAP/SMTP traffic. This is useful in debugging

issues with mail retrieval from your mail server or mail sending issues.

You can define the number of days to keep log entries for on the first tab of the options.

Email2DB also stores the log for each processed message in the Message Store database. You can view the process log for any processed message by viewing the message in the Message Store and clicking the **Trigger Process Log** button.

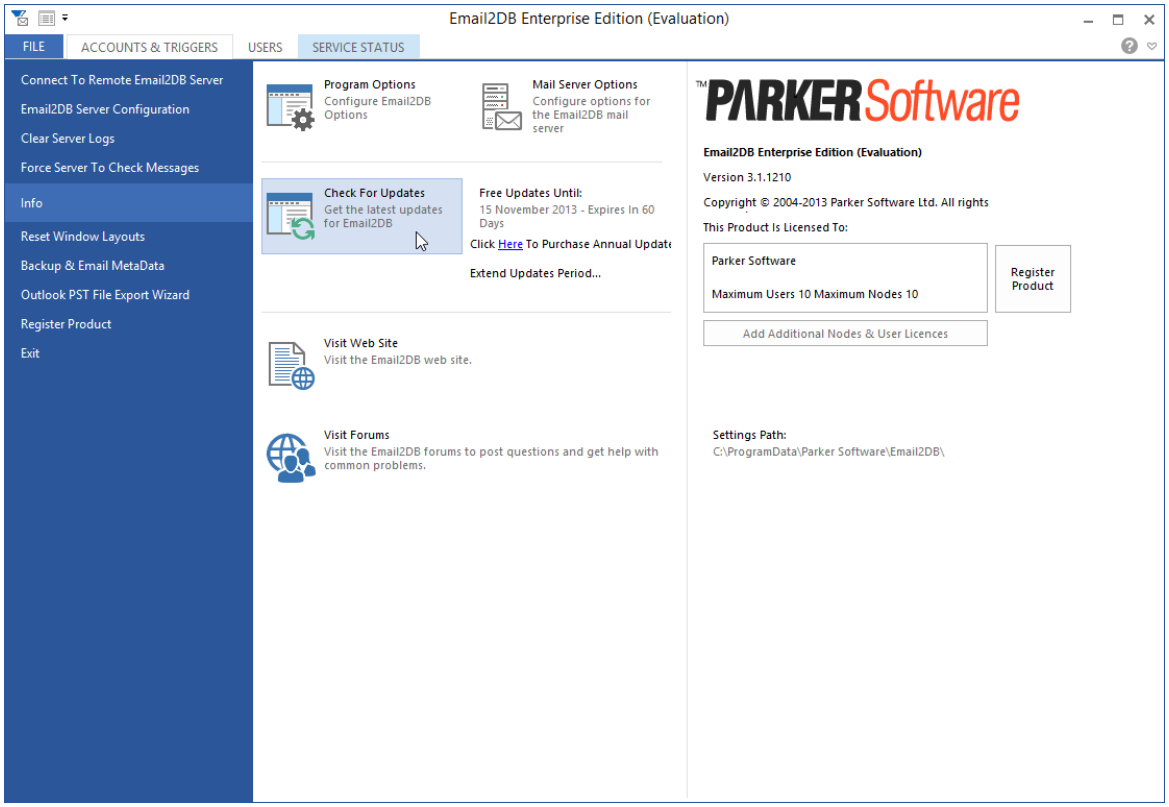
## Organizations

You must have at least one Organization defined. Each user you create is assigned to an Organization. Email2DB creates a default Organization when it is first run. See: Organizations

## 4.6 Checking For Updates

After installation and on a regular basis thereafter, you should check for updated versions of Email2DB.

Click the **File** tab on the Ribbon bar and click the **Check For Updates** button. The date when you can receive free updates until will be shown. Click this date to refresh it if you have purchased annual updates.




This will load the Email2DB web site and display any available updates.

The Email2DB Administrator will also check for updates itself once per month if an Internet connection is available.

# Part V

# 5 Getting Started - Email2DB Cloud Edition

Email2DB Cloud Edition is the hosted edition of our 'Email2DB' product. Email2DB is a combined message parser, message automation server and email server. Use Email2DB to read Email Messages, Twitter Feeds, Web Pages, RSS Feeds and other sorts of messages from multiple sources - then extract useful information from these messages to update your databases and perform multiple automated actions. Automate the processes that need to occur when your business receives incoming messages and dynamically respond to things happening on the web.

You can sign up to Email2DB Cloud Edition here: http://www.email2db.com/cloud/pricing.aspx

Once signed up you will receive a setup email containing your Email2DB Cloud Edition Server Address, Username & Password.

Download the Email2DB Cloud Edition Administrator from http://www.email2db.com/cloud/downloads.aspx

The Email2DB Cloud Edition Administrator is a Windows application. It will run on any Windows Vista, 7, 8, 2003, 2008, 2012 machine (32 or 64bit). It requires 100mb of free disk space.

The Email2DB Cloud Edition Administrator can be installed on any machine with Internet access.

See Also: Quick Start - Email2DB Cloud Edition

# 5.1 Quick Start

This chapter will help you to get started with Email2DB Cloud Edition as quickly as possible.

Email2DB Cloud Edition is the hosted edition of our 'Email2DB' product. Email2DB is a combined message parser, message automation server and email server. Use Email2DB to read Email Messages, Twitter Feeds, Web Pages, RSS Feeds and other sorts of messages from multiple sources - then extract useful information from these messages to update your databases and perform multiple automated actions. Automate the processes that need to occur when your business receives incoming messages and dynamically respond to things happening on the web.

To start the Email2DB Cloud Edition Administrator select **Email2DB Administrator** from the **Email2DB Cloud Edition** folder on your Start menu.

## How It Works

Email2DB can read messages from multiple sources (POP3 Servers, IMAP Servers, Exchange Servers, Twitter & RSS Feeds). You can also send messages for processing directly to your Email2DB email addresses.

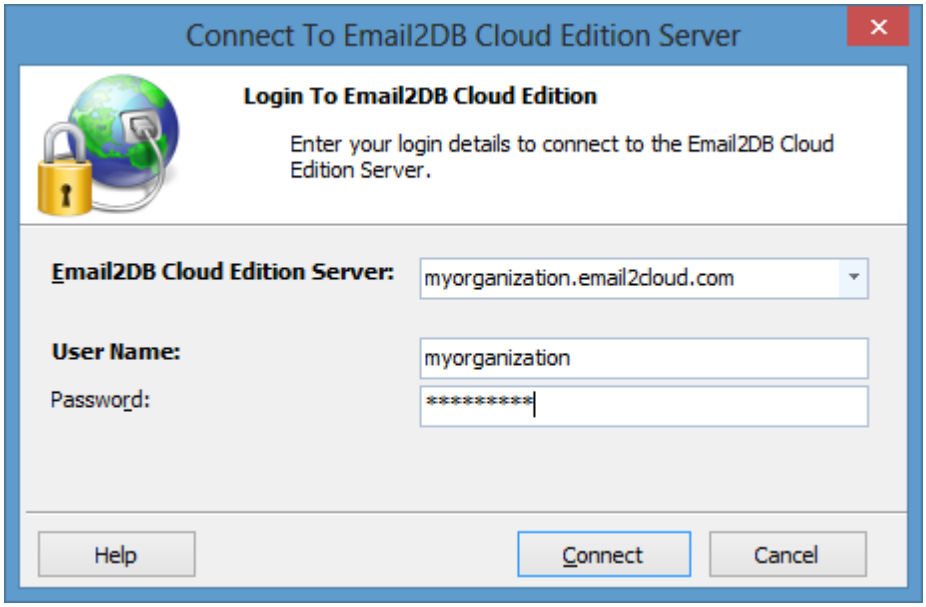
You can create any number of 'Triggers' that are checked against these incoming messages. A Trigger is a set of conditions that Email2DB will check for (for example, a specific 'from' address, or specific words in the subject line). If a message passes the Trigger conditions, Email2DB will process a series of 'Actions' against it. These actions include the updating of a database, sending email responses, printing reports and many more.

After Email2DB has processed an incoming message it stores a copy of the message the Message Store. This provides a useful central repository of all messages, and prevents the same message from being checked and processed twice. During development of your Triggers, you can reprocess messages held in the Message Store to rerun Actions against a message.

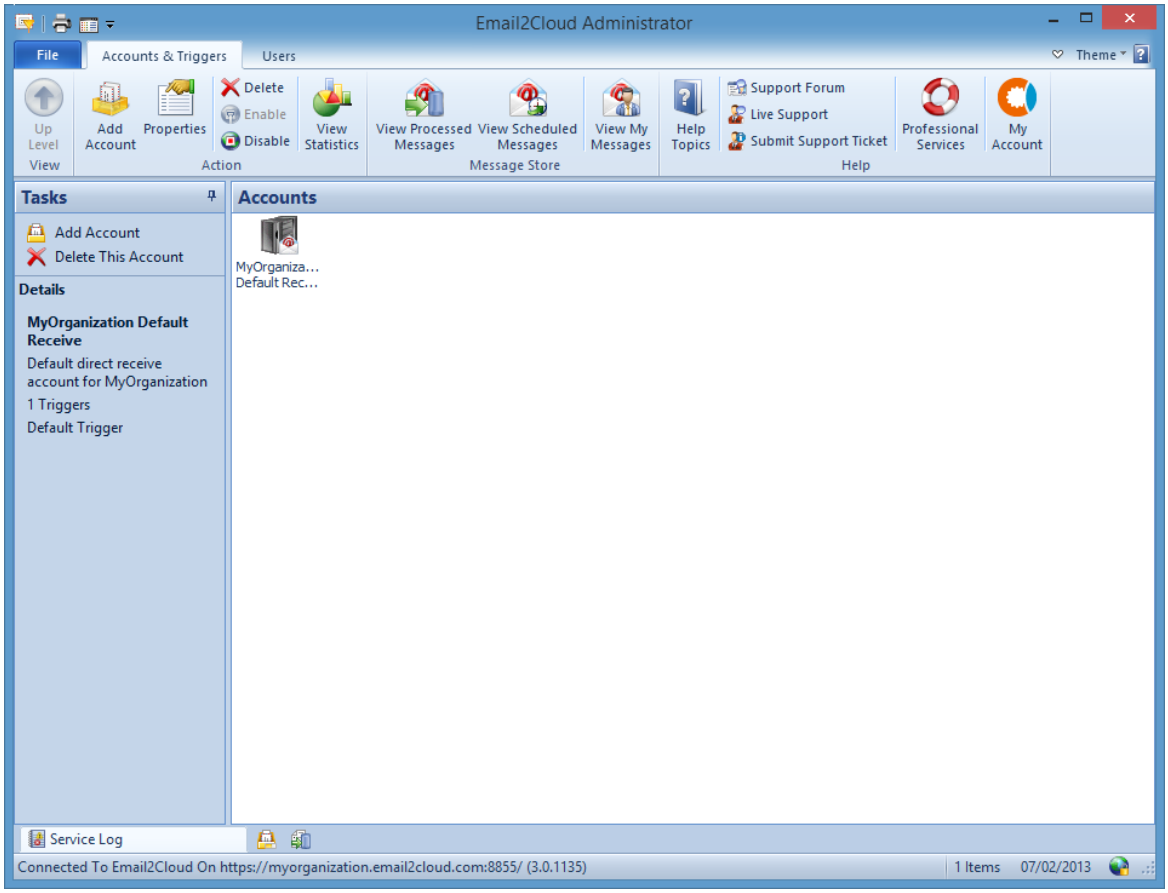
## Installing & Starting Email2DB

Install the Email2DB Administrator by running the Email2CloudAdministrator.exe setup program. The Email2DB Administrator requires Windows 7, 8, XP, Vista, 2003 or 2008 (32 or 64 bit) operating systems.

Before you can use the Email2DB Administrator, you need to login. Refer to your Setup Email for your **Email2DB Cloud Server**, **User Name** & **Password**.




Click **Connect** to login to Email2DB:




By Default your new Email2DB server will contain one Email2DB Account. This Account is setup to receive emails at 'backup@{your organization}.email2cloud.com'. Any email sent to this address will be processed by the Triggers in this Account. By default the Trigger simply stores a copy of the message in the Message Store and sends an email back to the sender with a URL link to view this message in the Message Store.

You can add more Accounts for any alias @{your organization}.email2cloud.com and setup Triggers

to do different things. You can also create Accounts to read 'pull' messages from other email accounts (Gmail, Yahoo, Exchange, Office 365 etc). Triggers can be created to perform automated actions when each message is read.

## Adding An Account

Click the **Add Account** button to create Message Retrieval Account. This tells Email2DB how and where to read your emails (or other types of messages) from. Once you have created an Account, Email2DB will read your messages from the message source and pass each message to one or more 'Triggers'. The triggers define what conditions to apply to the incoming message. If the message matches the trigger conditions then Email2DB can extract information from the message and perform database updates and other actions.

Now See: Creating Message Retrieval Accounts and Message Triggers

## 5.2 Checking For Updates

After installation and on a regular basis thereafter, you should check for updated versions of the Email2DB Administrator.

Click the **File** tab on the Ribbon bar and click the **Check For Updates** button.

This will load the Email2DB web site and display any available updates for the Administrator.

Your Email2DB Cloud Edition Server is updated automatically.

## 5.3 Accessing Your Email2Cloud Account

On the **File** menu click the **My Account** button.

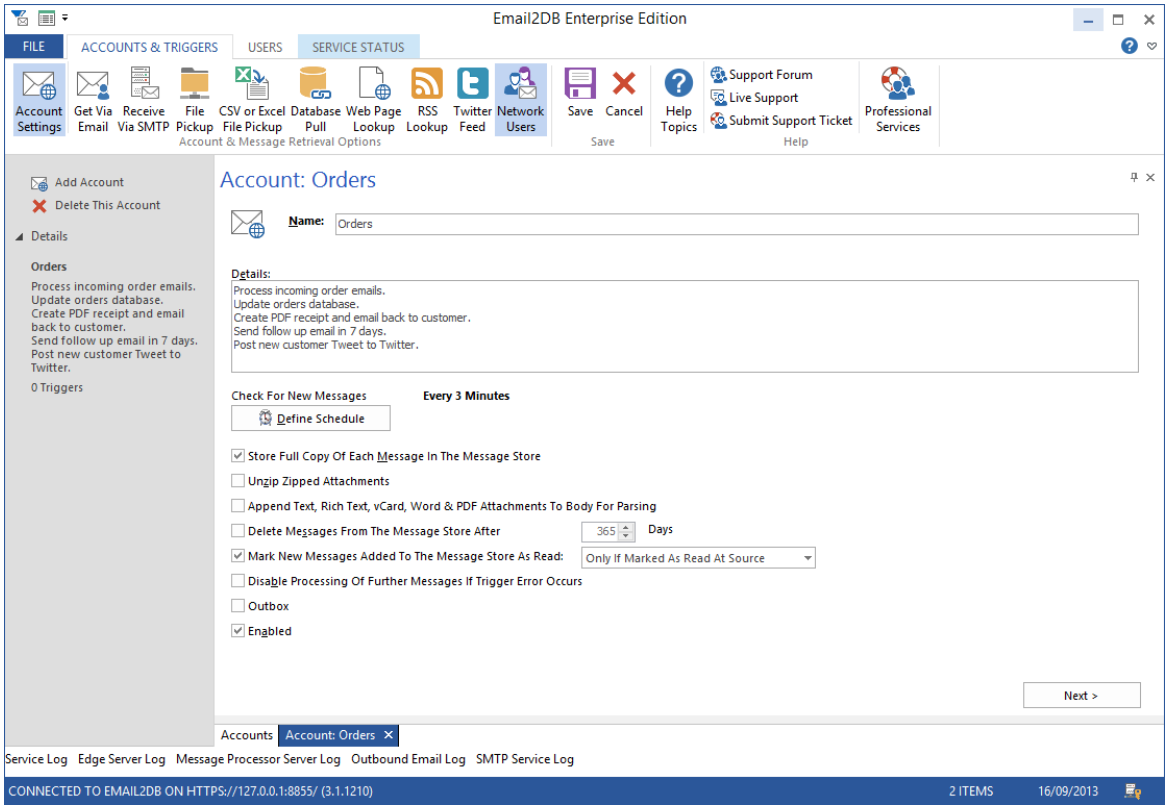
This will launch a web page showing your current Email2DB Cloud Edition Account details. Your message usage and storage will be shown for the current month, previous month and in total.

# Part VI

# 6 Message Retrieval Accounts

Within Email2DB you create one or more 'Accounts'. An Account defines where Email2DB will read and parse messages from. Within each Account you create one or more 'Triggers'. A Trigger defines what information Email2DB will check for in incoming messages and what actions it should perform when a message matches the Trigger conditions.

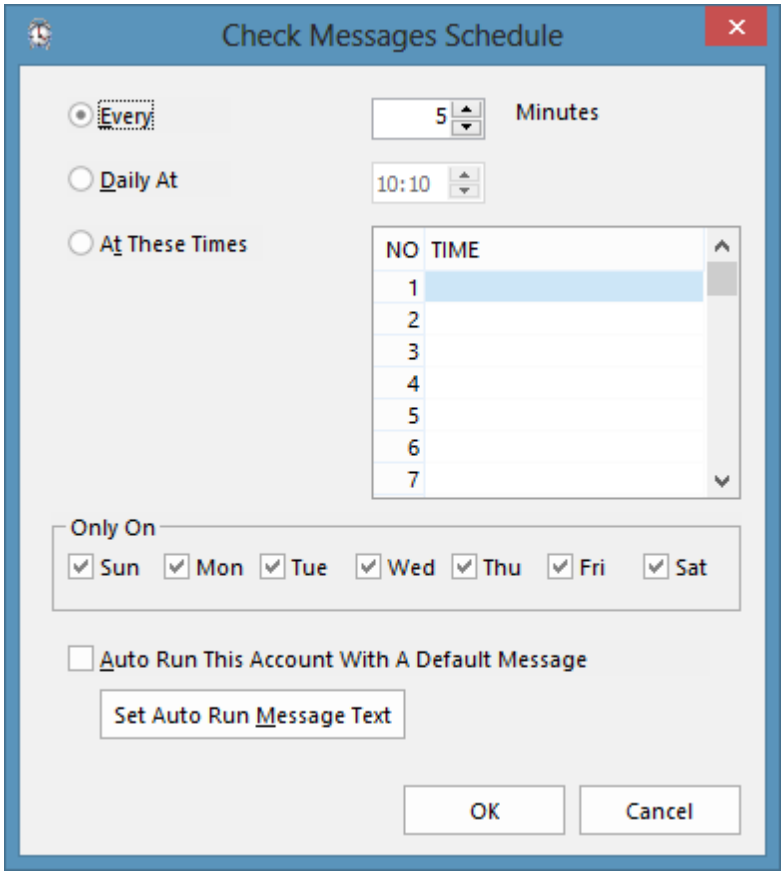
This form will be displayed automatically if you start Email2DB and no accounts exist. You can also click **Add Account** from the Ribbon Bar.




Enter the **Name** and **Details** of the Account.

### Check For New Message Every

Specify how often you want Email2DB to check for new messages from all the sources defined for this account. Click the **Define Schedule** button to setup the checking frequency schedule:




You can choose to check for new messages Every x minutes, daily at a fixed time or at a defined list of times. You can also select the days of the week that this account will be active.

**Store Full Copy Of Each Message In The Message Store**

When Email2DB reads messages from the source, it stores a copy of the message unique identifier in its own 'Message Store' database. This prevents the same message from being checked and processed twice for the same Email2DB account. By default Email2DB will just store the unique identifier and the message from/to address and subject in this database. Enable this option if you want Email2DB to also store the full message (including attachments).

Enabling this option will of course increase the size of the Email2DB Message Store database.

See Also: Configuring The Message Store Database

*Note: You can view the contents of the Message Store by right-clicking the Account and selecting View Processed Messages*

**Unzip Zipped Attachments**

If this option is enabled, Email2DB unzips and replaces any Zip file attachments with the expanded contents. For example, if an email contained a single Zip file containing 3 PDF files as an attachment, then when the email is received by Email2DB the email would contain 3 PDF file attachments, and the Zip attachment would be gone. If an email contains multiple Zip file attachments, each Zip is expanded and replaced with the contents.

**Append Text, Rich Text, VCard, Word & PDF Attachments To Body For Parsing**

If this option is enabled then Email2DB will automatically add text, VCard, Word & PDF attachments to

the plain text body of the message for parsing. You can then parse and extract data from the body text as normal. Rich text, Word and PDF attachments will first be converted to plain text before being added to the body.

**Delete Messages From The Message Store After x Days**

If this option is enabled, Email2DB will automatically delete old messages from the Message Store that are older than the number of days specified. Email2DB deletes the old data each day at midnight. This option is useful if Email2DB is processing a large number of emails and you don't need to keep a record of each processed message.

**Mark New Messages Added To The Message Store As Read**

When viewing the Message Store, messages that are not marked as read will show in bold. Opening a message will mark it as read. This option allows you to set the default read status for new messages. For messages read via Exchange Web Services, Email2DB is able to carry over the read status from the server.

**Disable Processing Of Further Messages If Trigger Error Occurs**

Enable this option if you want Email2DB to stop processing messages for this account if an error is generated whilst it is processing an email. Email2DB will then mark the Account as paused. You can un-pause the Account via the Email2DB Administrator. This option is useful during testing of your triggers.

**Outbox**

You can define one message account to be the 'Outbox' account. Any emails sent by Email2DB will be checked against triggers in this Account. This enables you to run processes on emails that Email2DB sends as well as receives. (For example, you could simply use this Account to keep a record of each email sent). See Also: Outbox Processing

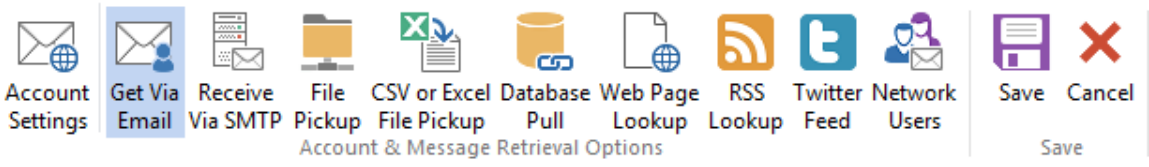
**Enabled**

Check this box to enable or disable processing of emails from this account.

## Message Retrieval Options

You then need to define where Email2DB will check for and retrieve new messages. Multiple sources can be defined for each account (email, file pickup, database pull etc), however in most cases a single Account will read messages from a single source.

The Ribbon Bar shows the message retrieval options:




| | |
|---|---|
| Get Via Email | For reading messages from mail servers using POP3, IMAP or Exchange Web Services. |
| Receive Via SMTP | For receiving messages directly via the built-in mail server. |
| File Pickup | For reading messages placed in a local pickup folder. |
| CSV or Excel File Pickup | For reading messages for each new row found in CSV files or Excel Spreadsheets. |

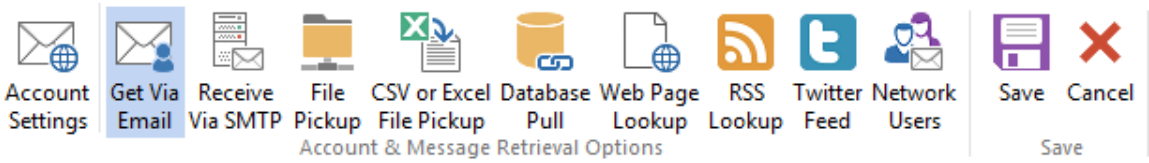
| | |
|---|---|
| Database Pull | For reading messages from a database. |
| Web Page Lookup | For monitoring changes to a web page. |
| RSS Lookup | For reading messages from an RSS feed. |
| Twitter Feed | For reading Tweets from a Twitter feed. |
| Network Users | For allowing users running the Email2DB Client to send messages. |

The enabled message retrieval options for the current account will be highlighted.

See Also: Reading Messages

## 6.2 Reading Messages

Email2DB can read and parse messages from a variety of different sources. When setting up an Email2DB Account, the ribbon bar contains a number of message source buttons:




The buttons will be highlighted if that message source is enabled for the Account.

**Get Via Email:**

**POP3 Servers**
Email2DB can read messages from any POP3 mail server.

**IMAP Servers**
Email2DB can read messages from any IMAP compatible mail server. This includes Microsoft Exchange Server.

**Exchange Servers**
Email2DB can sync messages with a Microsoft Exchange Server 2007/2010 via Exchange Web Services (EWS). This includes Office 365.

**Microsoft Outlook**
Email2DB can read messages directly from Microsoft Outlook folders via the Email2DB Client application. (See Also: Outlook PST Export Wizard)

**Network Users**
You can install the Email2DB 'Client' Application on multiple remote workstations. Network users can drag & drop messages onto the Email2DB Client application. These will be sent to Email2DB for processing.

**File Pickup**
Email2DB can read RFC 2822 formatted text files (EML) or Outlook Message Files (MSG) from any folder on your system.

**CSV or Excel File Pickup**
Email2DB can read CSV (comma separated values) files and Excel Spreadsheets (2007 or higher) from any folder on your system.

### Additional Message Sources Available With Email2DB Enterprise Edition

The Enterprise & Hosted Editions of Email2DB can also process messages from the following sources:

**Receive Via SMTP**
The Enterprise Edition can also receive emails directly via a built-in mail server using the SMTP Receive option. This option gives the best possible performance as Email2DB will respond to incoming emails instantly, instead of having to periodically login and download emails from external mail servers.

**Database Pull**
Email2DB can connect to an external database and check for new records.

**Web Page Lookup**
Email2DB can read a web page. It will use the content of the web page as it's message text. Triggers will execute whenever the HTML returned changes.

**RSS Lookup**
Email2DB can read messages from any RSS or Twitter feed. Triggers will execute for each new message added to the feed.

**Twitter Feed**
Email2DB can connect to Twitter and download the latest Tweets for any given search term.

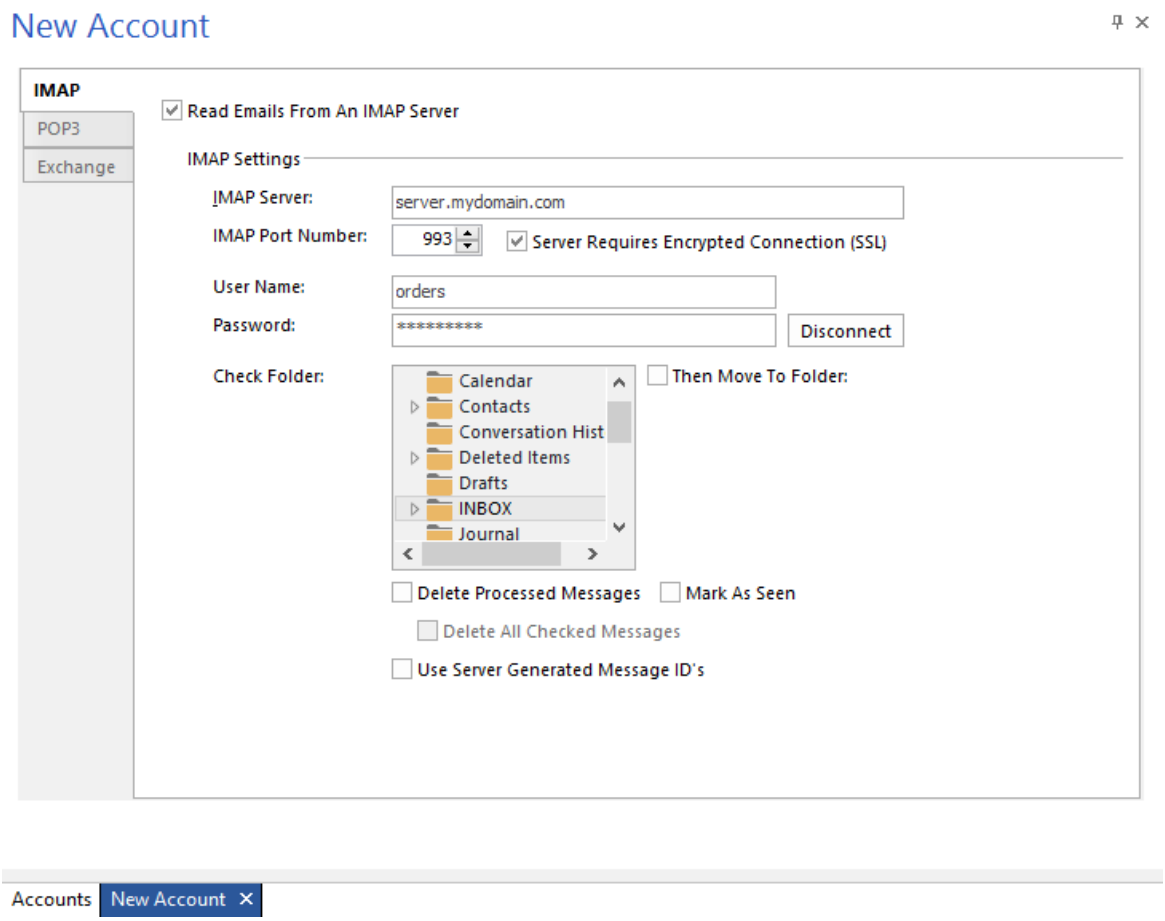
**Web Services**
Messages can be sent to Email2DB for processing via the Email2DB Web Services. This enables developers to create custom message sources.

### 6.2.1 Via Email From IMAP

IMAP is a more advanced protocol than POP3 and is supported by most modern mail servers. If you have a choice to use IMAP or POP3 to read your emails you should always choose IMAP as this will give better performance.

Click the **Get Via Email** button - then select the **IMAP** tab, if you want Email2DB to retrieve email messages from any IMAP Server

Click the **Read Emails From An IMAP Server** checkbox to enable IMAP processing.




#### IMAP Server

Enter the IP address or DNS name of the IMAP mail server. If you are connecting to a Microsoft Exchange server then ensure that the IMAP connector is installed and running and that the IMAP port (143) is open on the Exchange server PC's firewall. For Gmail use: imap.gmail.com

#### IMAP Port Number

This defaults to 143 and normally does not need to be changed (unless secure IMAP is used - see below).

#### Server Requires Encrypted Connection (SSL)

Select this option if your IMAP server requires a secure connection. The port number will automatically change to 993 - which is the port IMAP uses for secure connections.

### User Name/Password

Enter the user name and password for the IMAP account that you want to use.

### Check Folder

Select the folder that you want Email2DB to read and process email messages from.

Click the **Connect** button to display the folders available for the account on the IMAP Server. You must select one of the folders.

### Then Move To Folder

Select this option if you want Email2DB to move messages it has downloaded to another folder. You must then select another IMAP folder.

### Mark As Seen

Select this option if you want Email2DB to set the 'SEEN' flag against each message that it reads. This option should be selected as it will improve performance. Each time Email2DB checks the mail server for new emails, it only has to check 'UNSEEN' messages if this option is enabled. Otherwise it must check all messages since the date of the last checked message.

### Delete Processed Messages

Check this option if you want Email2DB to delete 'Processed' messages from the mail server. By 'processed' we mean only those messages that match the Trigger Conditions . Other messages in the same account that do not match trigger conditions will not be deleted.

### Delete All Checked Messages

Select this option if you also want to delete messages that do not match the trigger conditions.

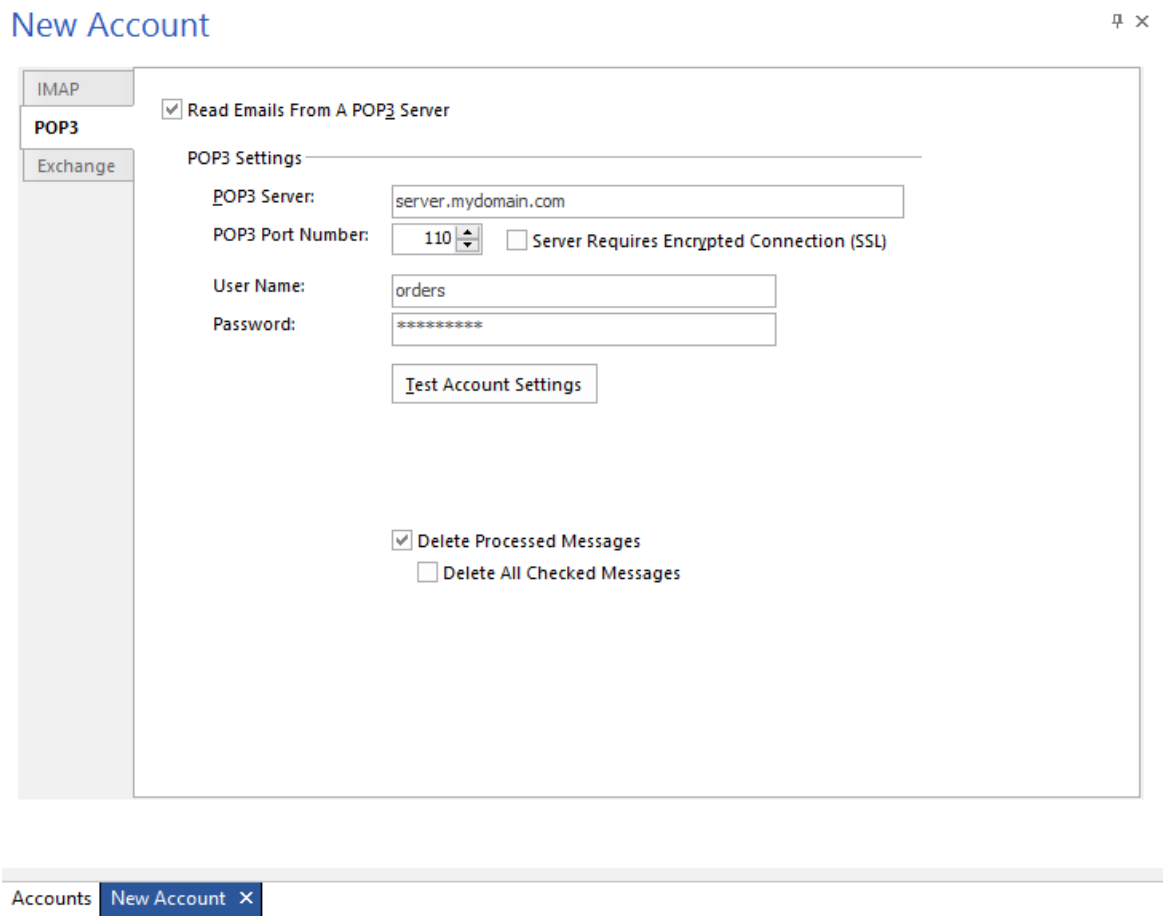
*Note: Even if you choose not to delete processed messages Email2DB will never process the same message twice. This is because it maintains its own mail store of checked messages.*

### 6.2.2 Via Email From POP3

POP3 is an email protocol supported by most email servers. POP3 works best when emails are read from the mail box and then deleted. POP3 is also designed for single user access. You must ensure that no other email clients will read and delete emails at the same time as Email2DB. If your mail server supports both POP3 and IMAP you should always use the IMAP protocol.

Click the **Get Via Email** button - then select the **POP3** tab, if you want Email2DB to retrieve email messages from any POP3 Server.

Click the **Read Emails From A POP3 Server** option to enable POP3 processing.




**POP3 Server**

Enter the IP address or DNS name of the mail server.

**POP3 Port Number**

This defaults to 110 and normally should not be changed (unless a secure connection is used - see below).

**Server Requires Encrypted Connection (SSL)**

Select this option if your POP3 server requires a secure connection. The port number will automatically change to 995 - which is the default port POP3 uses for secure connections.

**User Name/Password**

Enter the user name and password for the account that Email2DB will check. Click the **Test Account Settings** to verify that Email2DB can connect and logon to the POP3 account.

**Delete Processed Messages**

Check this option if you want Email2DB to delete 'Processed' messages from the mail server. By 'processed' we mean only those messages that match the Trigger Conditions . Other messages in the same account that do not match trigger conditions will not be deleted.

**Delete All Checked Messages**

Select this option if you also want to delete messages that do not match the trigger conditions.

*Note: It is recommended that you enable this option for POP3 retrieved messages. This is because the POP3 protocol does not have any mechanism where Email2DB can only read new messages. So it has to re-check every message each time it checks for new emails. This can become very slow if the POP3 account contains many messages.*

*Note: Even if you choose not to delete processed messages Email2DB will never process the same message twice. This is because it maintains its own mail store of checked messages.*

### 6.2.3 Via Email From Exchange EWS

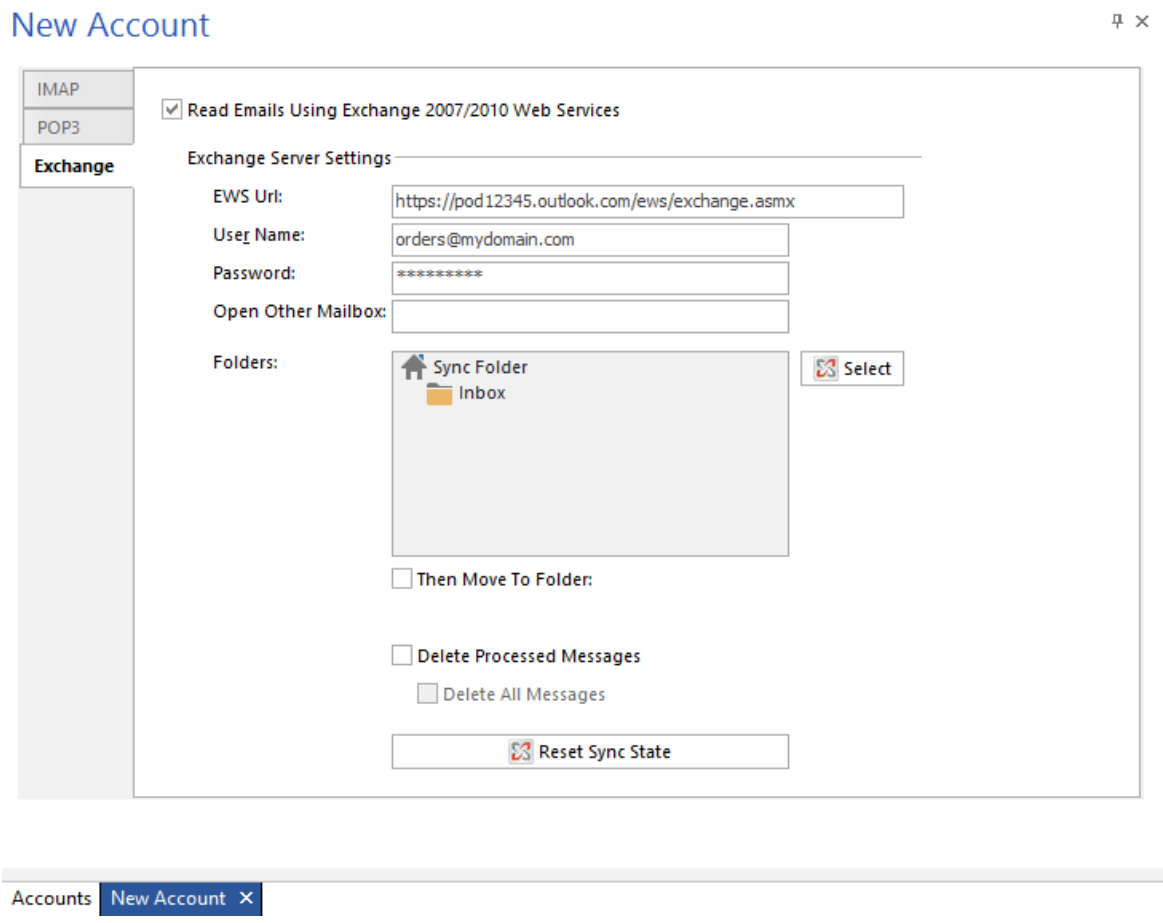
If you are using Microsoft Exchange Server 2007 or higher (including hosted Exchange or Office 365) then you can use Exchange Web Services (EWS) to sync messages with Email2DB.

The advantage of Exchange Sync is that only new messages are downloaded on each synchronization. However if your mailbox has a large number of messages then the first sync can take some time as Email2DB has to download each message - regardless of your Trigger Conditions.

Exchange Web Services uses a secure HTTP connection on port 443.

Click the **Get Via Email** button - then select the **Exchange** tab, if you want Email2DB to retrieve email messages from an Exchange Server.

Click the **Read Emails Using Exchange 2007/2010 Web Services** option to enable EWS processing.




**EWS Url**

This is the https address for your Exchange Server web services. By default this will be *https://yourserver/ews/exchange.asmx* - where 'yourserver' is the ip address of domain/computer name of your Exchange Server. For Office 365 you can find your server name using the Outlook Web Access - My Account settings page. Click the Settings For POP3, IMAP and SMTP access.. link. The server name will be something like pod12345.outlook.com

**User Name/Password**

Enter your user name & password for the mailbox you want to sync with. These are case sensitive entries.

**Open Other Mailbox**

You can optionally open another mailbox using the user name/password specified - provided the user has been given access to the other mailbox by the Exchange Administrator. The user name or email address of the other user can be specified.

**Folders**

You must now select the folder to sync. Click the **Select** button to view all of the available folders for the given user.

**Then Move To Folder**

Select this option if you want Email2DB to move messages it has downloaded to another folder. You must then select another folder on your Exchange Server.

**Delete Processed Messages**

Check this option if you want Email2DB to delete 'Processed' messages from the mail server. By 'processed' we mean only those messages that match the Trigger Conditions . Other messages in the same account that do not match trigger conditions will not be deleted.

**Delete All Checked Messages**

Select this option if you also want to delete messages that do not match the trigger conditions.

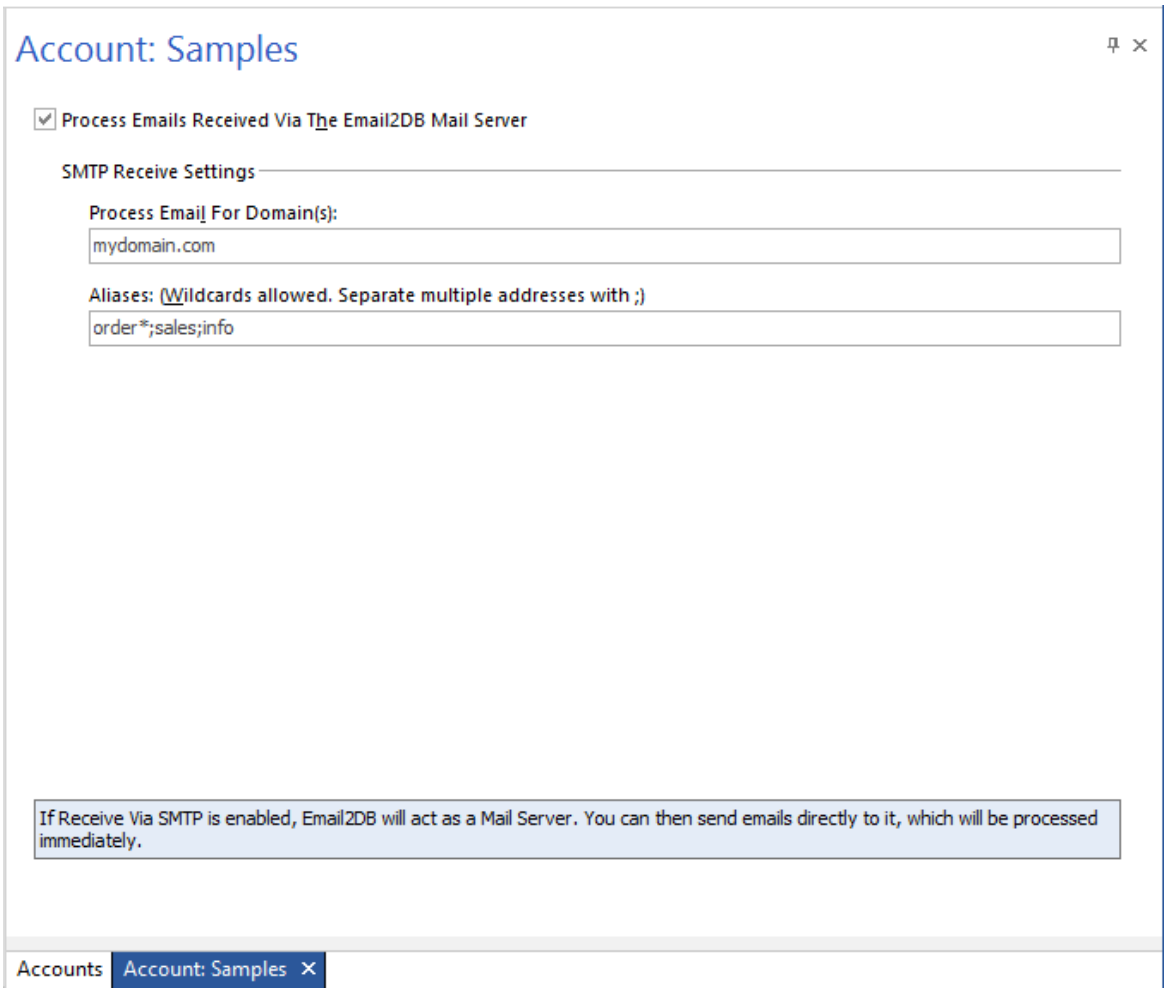
**Reset Sync State**

Click this button if you need to re-sync with your Exchange Server. You will need to do this if you change any of your Triggers and want Email2DB to re-download all messages again.

### 6.2.4 Via SMTP Receive

The Email2DB Enterprise & Hosted Editions can receive incoming emails directly via the built-in Mail Server. This enables emails to be sent directly to Email2DB via SMTP. When Email2DB receives an email via the built-in mail server it will be processed immediately.

Click the **Receive Via SMTP** option on the Ribbon Bar.




Click the **Process Emails Received Via The Email2DB Mail Server** option to enable the mail server for this Account.

#### Process Email For Domain

Enter the domain name for this account. When an email is received by the mail server, Email2DB will check the domain of the message (the text after the @ sign) and only process emails for this account that have matching domain names. You can specify multiple domains, separated with ;.

*Note: The domain name is the part after the '@' in the email address. So for example, for email address support@parkersoft.co.uk - the domain is 'parkersoft.co.uk'*

#### Aliases

Enter the aliases to accept email for. You can separate multiple aliases with a semi-colon. Enter * to accept all aliases for the domain.

In the above example, any emails received for orders@mydomain.com and sales@mydomain.com

will be processed through this Account.

See Also: Mail Server Options

## Testing The SMTP Server

Create a new Email2DB Account and enable the Receive Via SMTP option.

Enter 'testdomain.com' for the Domain name and 'test' for the Alias.

Save the Account. Now create a Trigger for the new account called 'Test Trigger'. Simply save the trigger using it's default values. This will create a trigger for ALL incoming emails that will simply show a popup message and store the email in Email2DB's database.

To test the direct send option, simply use Outlook or Outlook express (or any other email client). Create a new email account and point the SMTP and POP3 servers to the IP address or host name of the Email2DB PC. If your email client is on the same PC as Email2DB you can specify 'localhost'.

Now use your email client to send an email to 'test@testdomain.com'

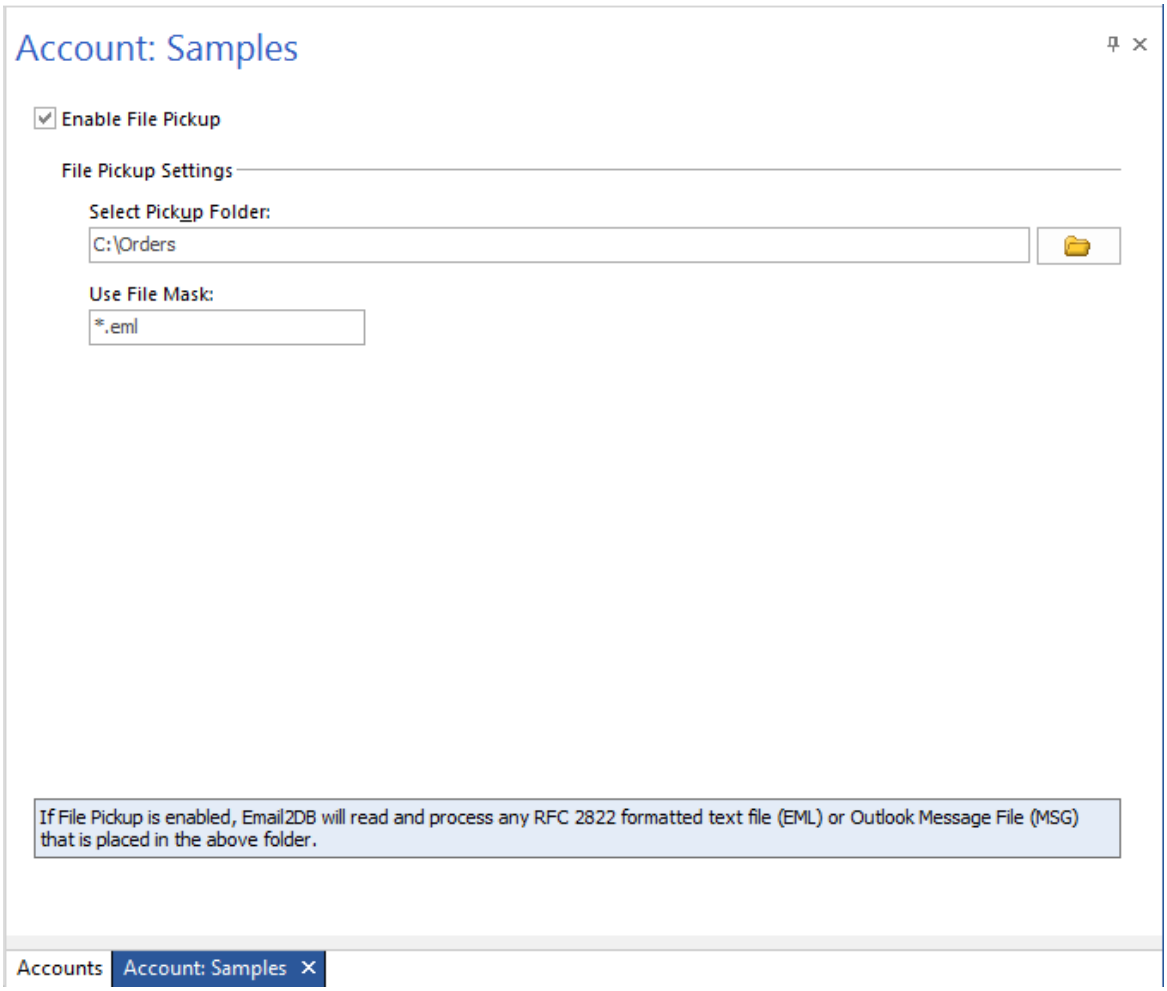
You will immediately see the popup appear and the trigger will show as processed in the Email2DB server log. The email will also appear in the Message Store (you can view processed messages by right-clicking the Account and selecting 'View Processed Messages').

You can test if outgoing emails are relayed by sending an email to any other valid domain.

### 6.2.5 File Pickup

Email2DB can read email messages directly from RFC 2822 formatted text files or from Microsoft Outlook MSG files.

Click the **File Pickup** option on the Ribbon bar.




Click the **Enable File Pickup** option to enable File Pickup.

Select the **Pickup Folder** on your PC that Email2DB should watch for new files. Enter the **File Mask** or enter *.* for all files.

Email2DB will then check the folder every x minutes (depending on the 'Check For New Mail Every' entry). Any files that match the 'file mask' will be read and parsed. They will be processed as normal emails. Files will be deleted after they are processed.

RFC 2822 Message Format

The format and structure of a basic Internet email message, as defined in RFC 2822, looks like this:

From: someone@somewhere.com
To: me@mysite.com
`Subject: Blah blah`

body text

The first section contains the headers. Each header must end with a : (colon). After the headers is a blank line followed by the body text.

You can test the file pickup option of Email2DB by creating a text file using Notepad in the above format. Save it to the folder specified on the File Pickup tab. In a minute or so the file will disappear and be processed by Email2DB.

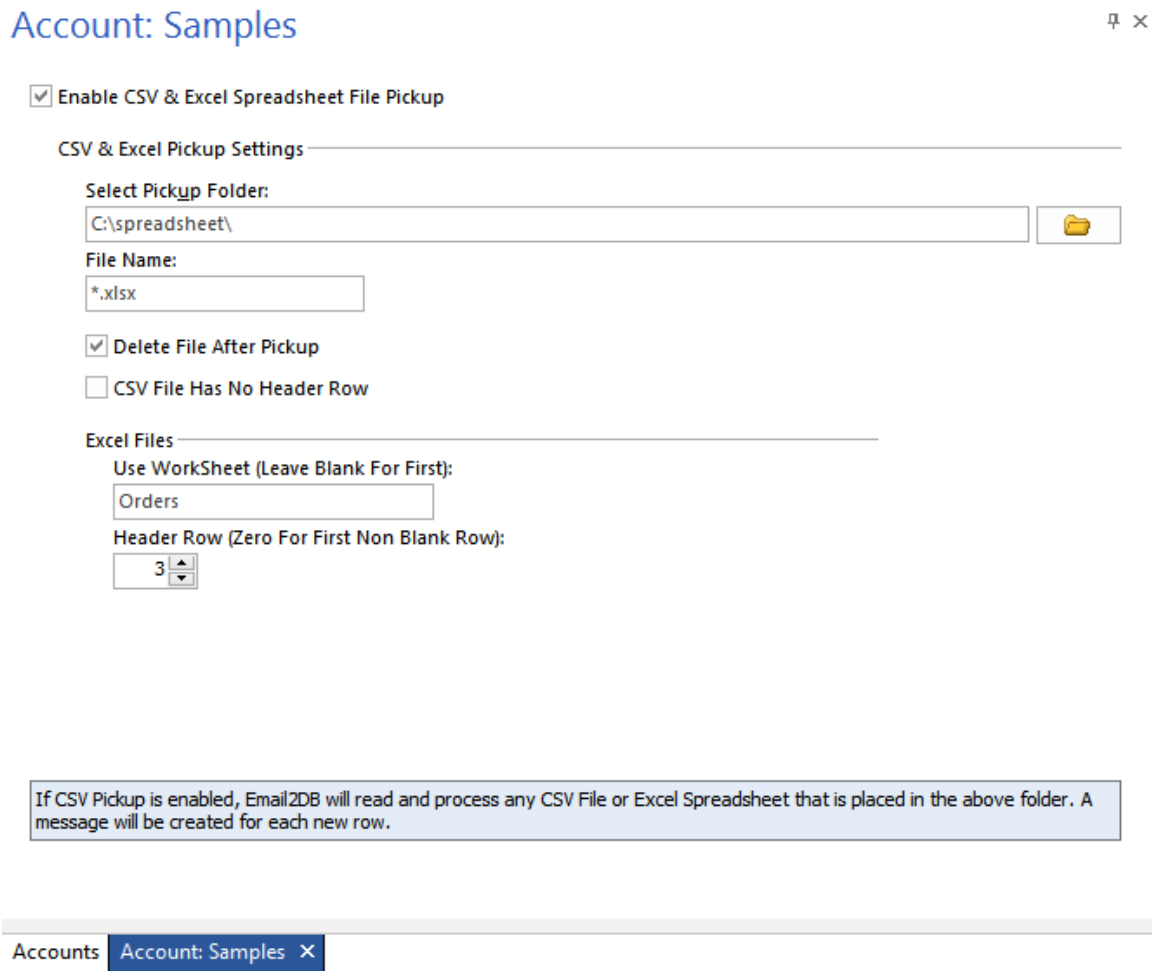
## Microsoft Outlook MSG Files

If Outlook MSG files are placed in the File Pickup folder (or dragged and dropped from Outlook itself), Email2DB will convert them to EML files before processing. Outlook does not need to be installed on the Email2DB computer. Supports Outlook 2003 and higher.

### 6.2.6 CSV or Excel Spreadsheet Pickup

Email2DB can read CSV (Comma Separated Values) files and Microsoft Excel Spreadsheet files (Excel 2007 or higher). It can create a new message for each new row in the CSV file or Spreadsheet.

Click the **CSV or Excel File Pickup** option on the Ribbon bar.




Click the **Enable CSV & Excel Spreadsheet File Pickup** option to enable.

Select the **Pickup Folder** on your PC that Email2DB should watch for new files. Enter the **File Name** or enter *.xlsx for all Excel files or *.csv for all CSV files.

Enable the **Delete File After Pickup** option if you want Email2DB to delete files after it has processed the contents. If this option is not selected then Email2DB will remember the row number for each processed file. It will then only create messages for NEW rows added to the existing file.

Enable the **CSV File Has No Header Row** option if you are reading a CSV text file and the data starts on row one. By default when a CSV file is picked up the first row is assumed to be the header row containing the field names.

For Excel files you can optionally enter the **Worksheet** to use if the spreadsheet contains multiple worksheets. Leave blank to use the first worksheet. You can also enter the **Header Row** number. This is the row in the spreadsheet that contains the column headers. If zero then the first non-blank row will be used.

## How It Works

The CSV & Spreadsheet File Pickup option differs from the regular File Pickup option in that a new message will be created for each row - whereas the File Pickup will create a single message for the entire file.

For example, consider the following spreadsheet:

| Product | Name | Quantity | Value |
|---|---|---|---|
| Item1 | Item 1 description | 100 | 1.20 |
| Item 2 | Item 2 description | 200 | 1.30 |

When Email2DB reads this spreadsheet (or CSV file) it will create 2 messages in the following format:

Message 1:

```
Row: 1
Item 1: Item1 description
Quantity: 100
Value: 1.20
```

Message 2:

```
Row: 2
Item 1: Item 2 description
Quantity: 200
Value: 1.30
```

The subject of the message will be '[filename] Row x'

The message body will be set to each column on a line of its own in the format 'header row text: column value'.

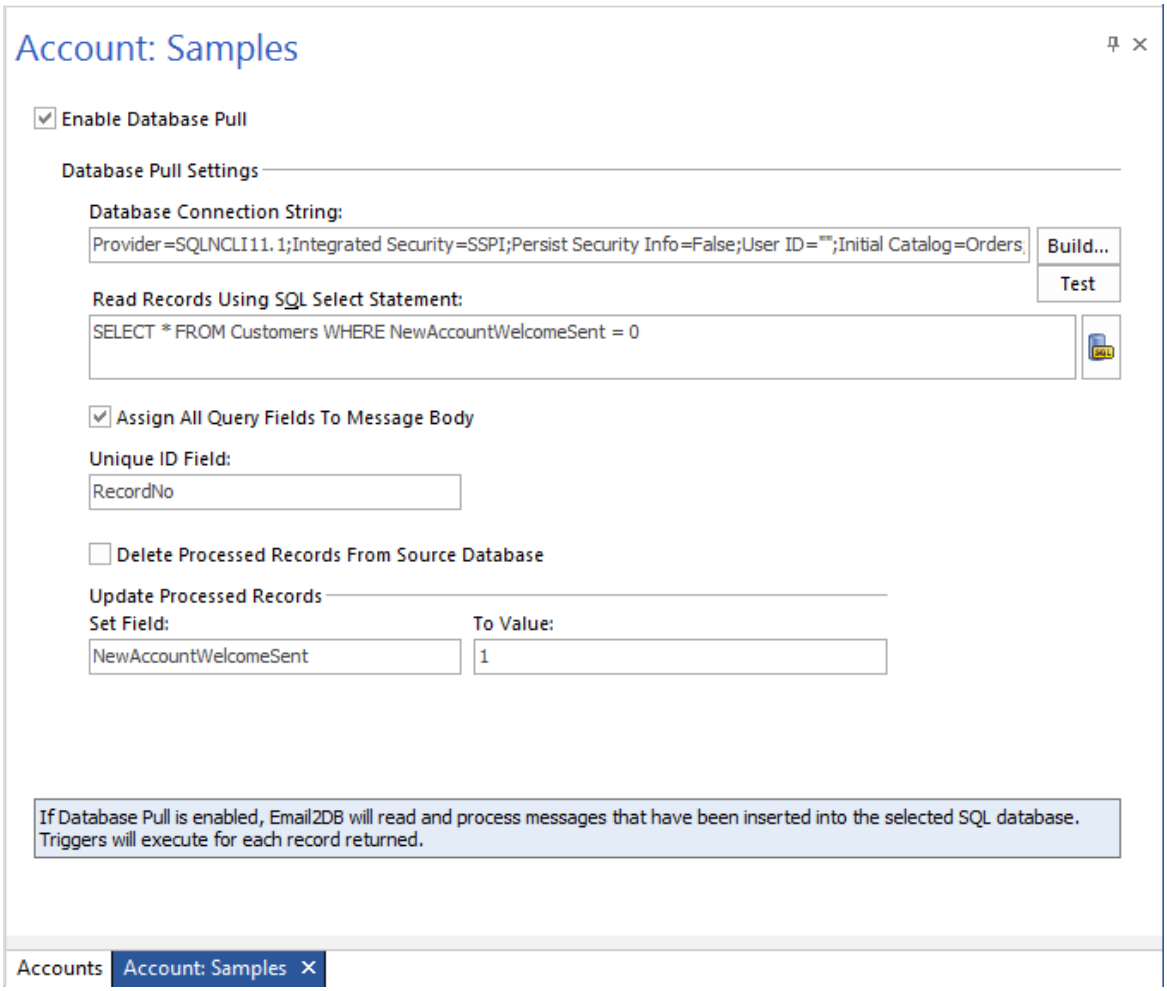
This layout will enable you to easily parse the contents.

Email2DB assumes that new rows will be added to the end of the file. New messages will be created for each new row found.

### 6.2.7 Database Pull

The Database Pull option allows you to retrieve message text from a database which you can then execute Trigger actions against.

Click the **Database Pull** option on the Ribbon bar.




Click the **Enable Database Pull** option to enable Database Pull.

#### How It Works

At the Account pre-defined interval (for example, every 3 minutes), Email2DB will read records from your database using a SQL 'SELECT' statement that you define. For each record returned it will pass the fields as the 'message text' to the Account Triggers. The first trigger that matches the trigger conditions will execute as with other message types. Email2DB will then either delete the records from your database or set one of the fields to a value of your choice.

For example, suppose you need to automatically send an email to new customers that are added to a database. You could have Email2DB check the 'Customers' table for new records at regular intervals. If a new record is found the message text could be set to the Customers records. Triggers could then send the customer an email. Email2DB will then update the customer table to indicate that the customer is no longer new.

#### Options

Enter the **Database Connection String** to your database or click the **Build** button to build the

connection string using the **Select Data Source** dialog. Any data source can be used that is available on your computer. See Also: Connection Strings

You must then enter a **SQL SELECT** Statement (or VIEW or stored procedure) that will be executed against your database to return one or more records. You must include a WHERE clause so that previously read records will not be re-read each time.

**Assign All Query Fields To Message Body**

If this option is enabled then Email2DB will set the message body text to all the fields returned from the query. The text will be set to the following format:

Fieldname: value
Fieldname: value
etc

If you do not use this option then you can specify a single field to be assigned to the message body. In this case, just the field value will be assigned the message body on its own.

The **Use Unique ID Field** can be used to enter a field name from the SELECT statement that contains a unique value for the record in the database. If you enter a field here, Email2DB will ensure that the same record is not processed twice. If no Unique ID is used then the same records could be processed multiple times if they are not filtered out using the WHERE clause (or deleted).

Select the **Delete Processed Records From Source Database** option if you want Email2DB to delete each record returned from the SELECT statement from your database after it has processed them.

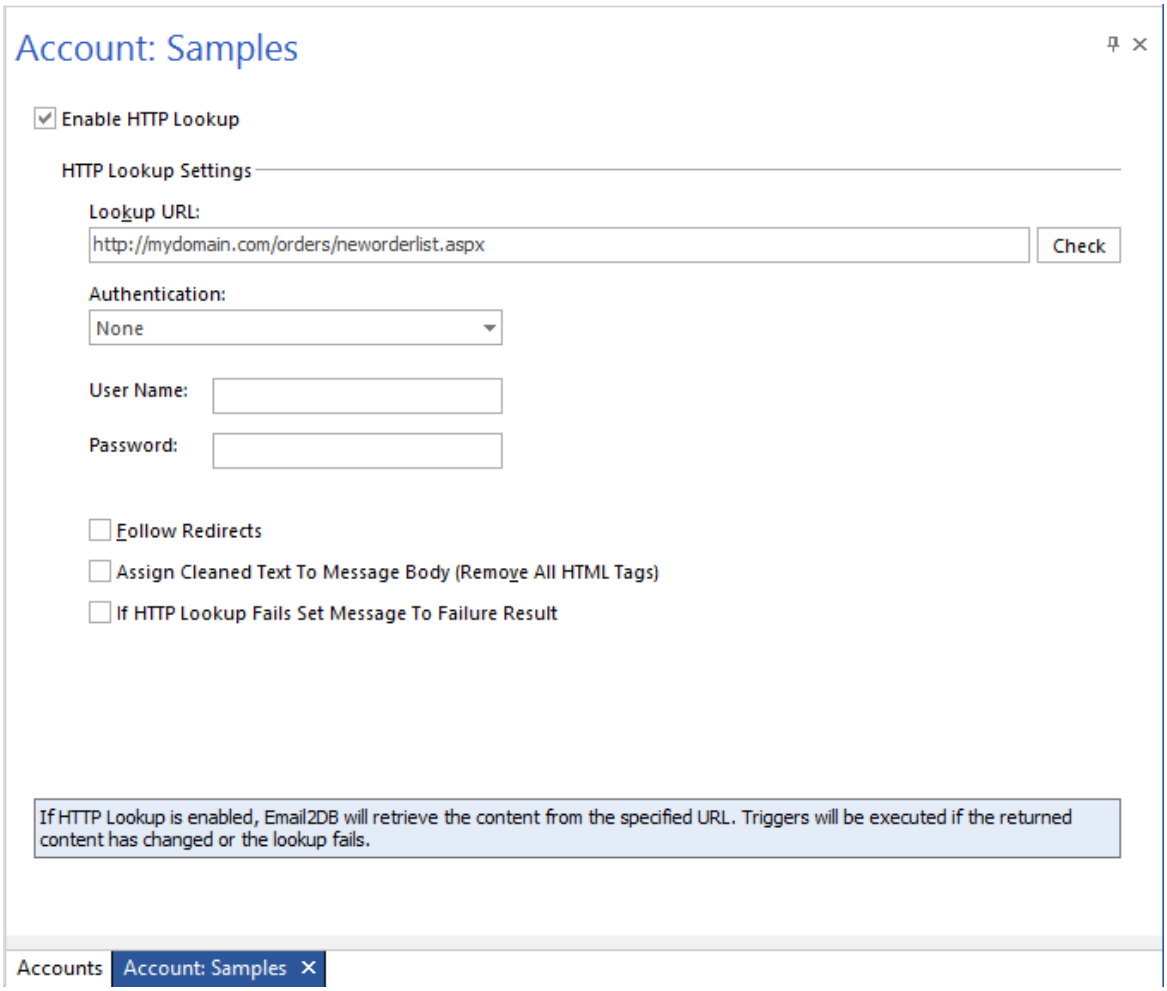
Alternatively you can **Update Processed Records**. In the **Set Field** entry, enter the database field to update. In the **To Value** entry, enter the value you want to set the field to. You can use % fieldname% replacements here.

*Performance Note: You should design your SELECT statement, VIEW or Stored Procedure, so that only the records you need to process are returned each time the database is checked, unless you know that only a few records will be returned each time.*

### 6.2.8 Web Page Lookup

Email2DB can also read text from any web page. This allows you to monitor a web page and then execute a Trigger when the content changes.

Click the **Web Page Lookup** option on the Ribbon bar.




Click **Enable HTTP Lookup** option to enable Web Page Lookup.

Enter the **Lookup URL**. This is the full URL of the web page you want to monitor - including any query string. Click the **Check** button to validate the URL.

If the web site requires a login enter the **Authentication** type along with the **User Name** and **Password**. Email2DB will then login first before requesting the URL.

#### Follow Redirects

Enable this option if Email2DB should follow any redirects.

#### Assign Cleaned Text To Message Body

If this option is enabled then Email2DB will strip all HTML tags from the returned page before assigning it to the message text that is used for Trigger parsing.

#### If HTTP Lookup Fails Set Message To Failure Result

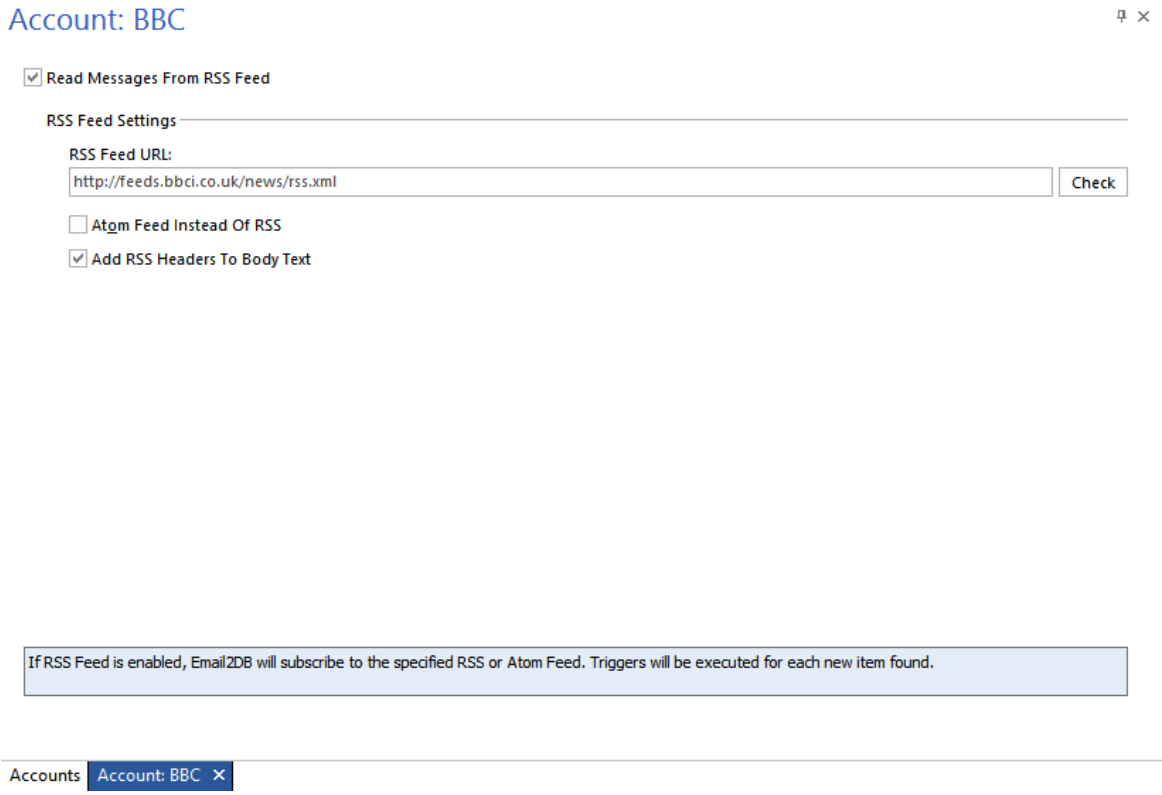
If this option is enabled then Email2DB will assign the error page text to the message body for Trigger parsing. This is useful if you want to execute a trigger when a web page lookup fails (for example, a 404 error).

When Email2DB reads the web page it creates a new message with the **MSG_HTML** set to the HTML and the **MSG_Body** set to the plaint text version of the HTML if the **Assign Cleaned Text To Message Body** option is enabled. The message headers will contain all the HTTP headers returned by the request. The **MSG_Subject** will be set to be set to the page <title> tag. Email2DB also creates an additional header called 'TimeTaken' - which will be set to the number of milliseconds the request took - this allows you to create Triggers based on how fast the web server responds.

### 6.2.9 RSS Lookup

Email2DB can also read messages from any RSS or Atom feed. Triggers will be executed when new items are added to the feed.

Click the **RSS Lookup** option on the Ribbon bar.




The RSS Feed option allows you to read RSS feed messages that you can then execute Triggers against.

Click the **Read Messages From RSS Feed** option to enable the RSS Feed Lookup.

Enter the **RSS Feed URL** that you want to subscribe to.

Enable **Atom Feed Instead Of RSS** if the feed is using the newer Atom format.

Enable **Add RSS Headers To Body Text** if you want additional RSS/Atom headers appending to the body text for parsing. If not selected, then just the item text will be added to the body.

#### How It Works

At the Account predefined interval (for example, every 10 minutes), Email2DB will read new messages from the defined RSS Feed. Each new item returned from the RSS Feed will be treated as a separate message. The message Body will be set to the text of the feed item.

The message Subject will be set to the Title tag of the item.

The message From address will be set to the item author, or source if the author is not specified.

You can then use your trigger conditions to examine the headers/content of the RSS Item and execute actions as you require.

### 6.2.10 Twitter Feed

Email2DB can read Tweets from a Twitter feed using your Twitter Account credentials. An Email2DB Message will be created for each new Tweet. You can then fire Email2DB Triggers based on the content.

Click the **Twitter Feed** option on the Ribbon bar.

Click the **Read Messages From A Twitter Feed** option to enable Twitter Feed reading.

Click the **Get Pin** button to authenticate with Twitter. Email2DB uses the secure OAuth mechanism to authenticate with Twitter. This ensures your Twitter password is never stored in the Email2DB Metadata.

When monitoring Twitter you have four options:

1. **Get My Timeline** - reads Tweets that you would normally see when you view Twitter with your account.
2. **Get Mentions** - reads all Tweets that mention you - this monitors all Tweets that include your Screen Name.
3. **Get Timeline** For Screen Name - reads all Tweets for a specified Screen Name. Enter the Screen Name to monitor.
4. **Get Tweets For Search Term** - reads all Tweets for the specified search string.

Each time Email2DB connects to Twitter it will download new Tweets since the last time it connected.

Enable the **Include Retweets** option if you want re-tweets included in the returned Tweets.

**Add User Information Fields To Plain Text**

If this option is enabled then some addition Tweet sender fields will be added to the plain text. For example:

```
Tweet          : UK banks should reveal all employees earning more than £1m a year,
#Labour's Ed Miliband says. Watch #PMQs LIVE: http://t.co/KbU9c2gp

Name           : BBC Breaking News
Screen Name    : BBCBreaking
Description    : Breaking news alerts and updates from the BBC. For news, features,
analysis follow @BBCWorld (our World edition) and @BBCNews (our UK edition).
Profile Image  : http://a0.twimg.com/profile_images/1143158124/
BBC_avatar_normal.jpg
Url            : http://www.bbc.co.uk/news
Followers      : 2579968
Time Zone      : London
Verified       : true
Following      : true
Language       : en
Location       : London, UK
```

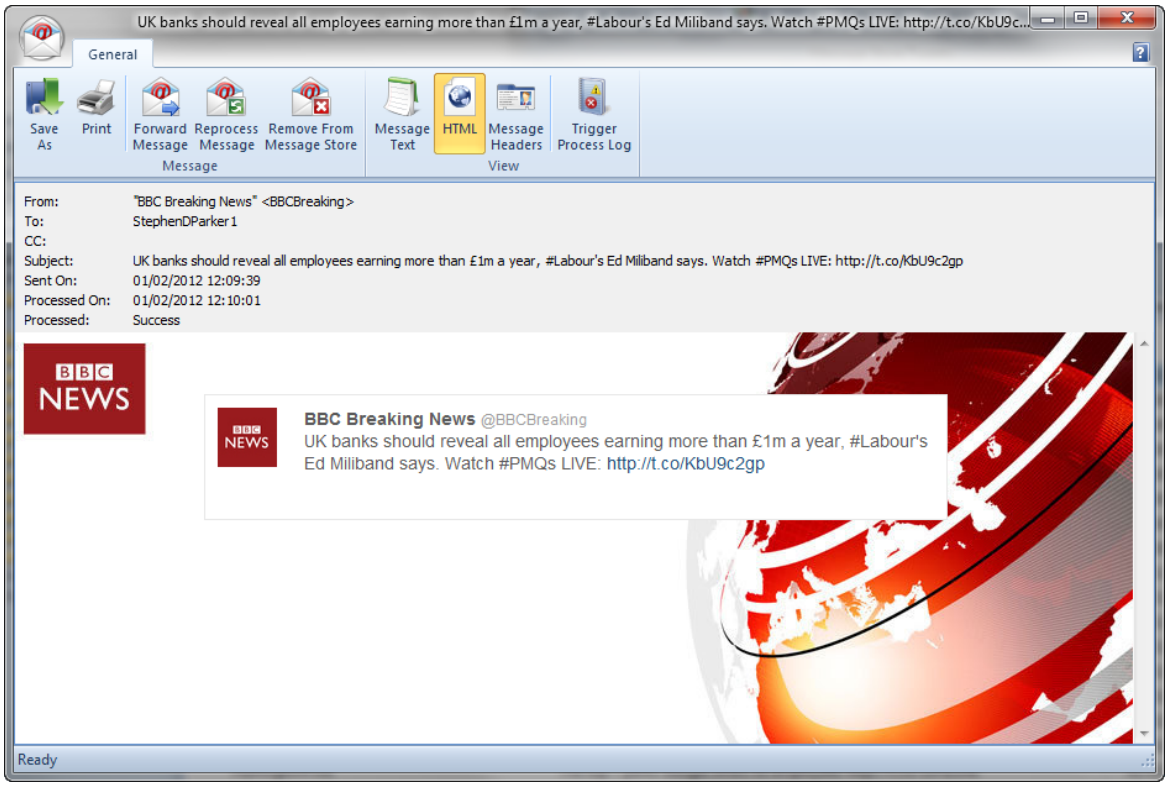
You can then parse and act on the message as you would any other type of message.

If not enabled then the body of the message will be set to the tweet text only. The message subject will be set to the Tweet text in both cases.

**Create HTML Page In Addition To Plain Text**

Enable this option if you want Email2DB to create a HTML version of the message in addition to the

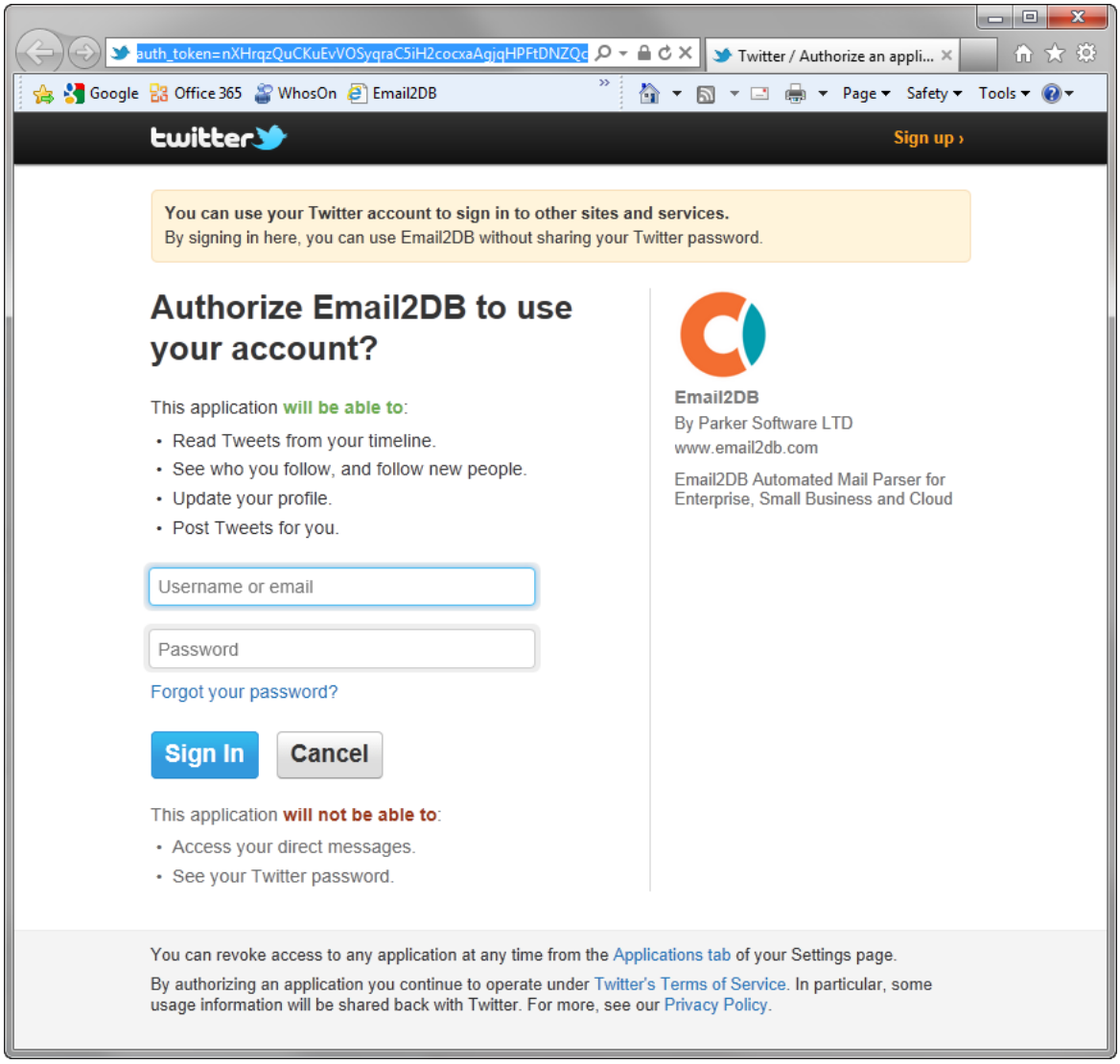
plain text. This will be viewable in the Message Store viewer:




## Authenticating With Twitter

Click the **Get Pin** button to start the authentication process.

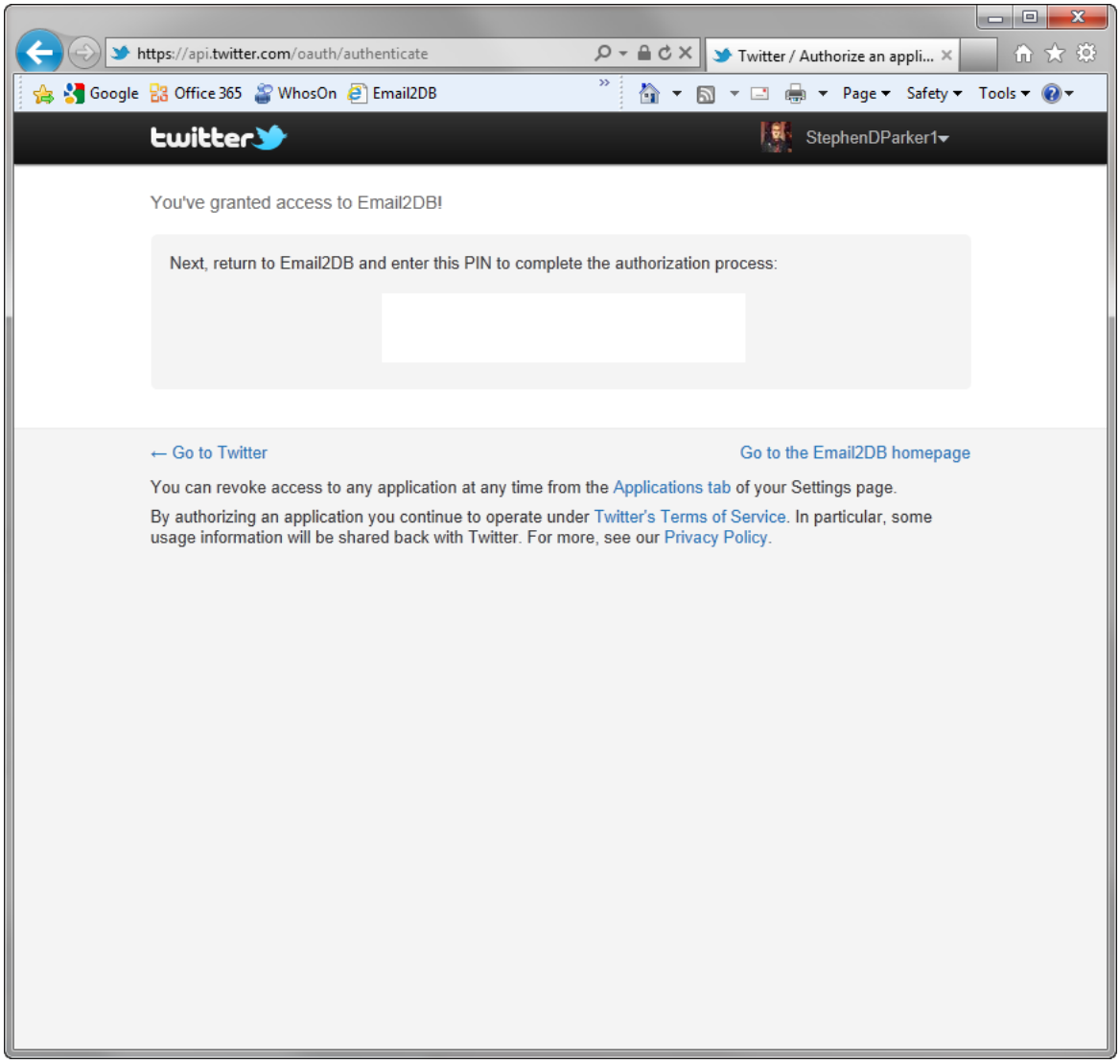
A Twitter login page will then be launched in your web browser:



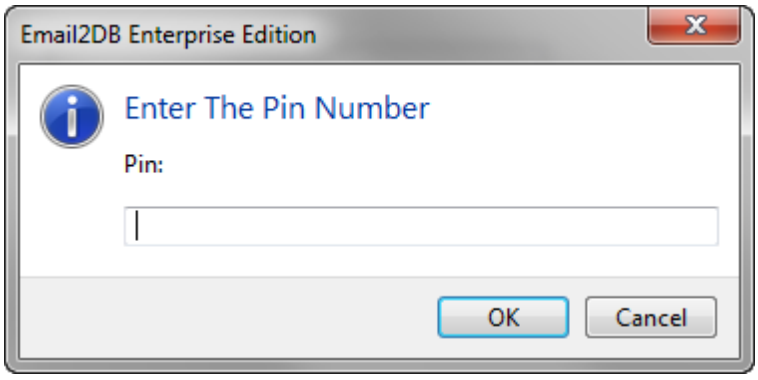

Login with your Twitter user name & password. Note: This information is NOT shared with Email2DB.

By logging in you are authorizing the Email2DB Application access to your Tweets.

Once you have signed in a Pin number will be displayed:




Enter the Pin number in the Email2DB prompt:




Once the correct Pin is entered Email2DB will be able to access your Twitter feed. You can disable access at any time in your Twitter settings - Applications tab.

### 6.2.11 Network Users

Email2DB can also receive messages from remote users running the Email2DB Client application. Users can drag and drop any message into the Email2DB Client. These messages will then be sent to Email2DB for processing. The Email2DB Client application can also integrate with Microsoft Outlook.

Click the **Network Users** option on the Ribbon bar.

Click the **Accept Messages From Remote Email2DB Clients** option to enable this option.

You can install the Email2DB Client application on multiple remote network workstations (or over the Internet). The Email2DB Client connects to the Email2DB Server via a secure HTTP connection. Users can then drag and drop any message onto the Email2DB Client. The Email2DB Client then creates a message which it sends to the Email2DB Server for processing.

Any type of file can be dropped onto the Email2DB Client. Email messages and Outlook MSG files will be sent as is. Text files will be converted to an RFC822 email formatted file before sending. Any other file type will be converted to an email message with the file included as an attachment.

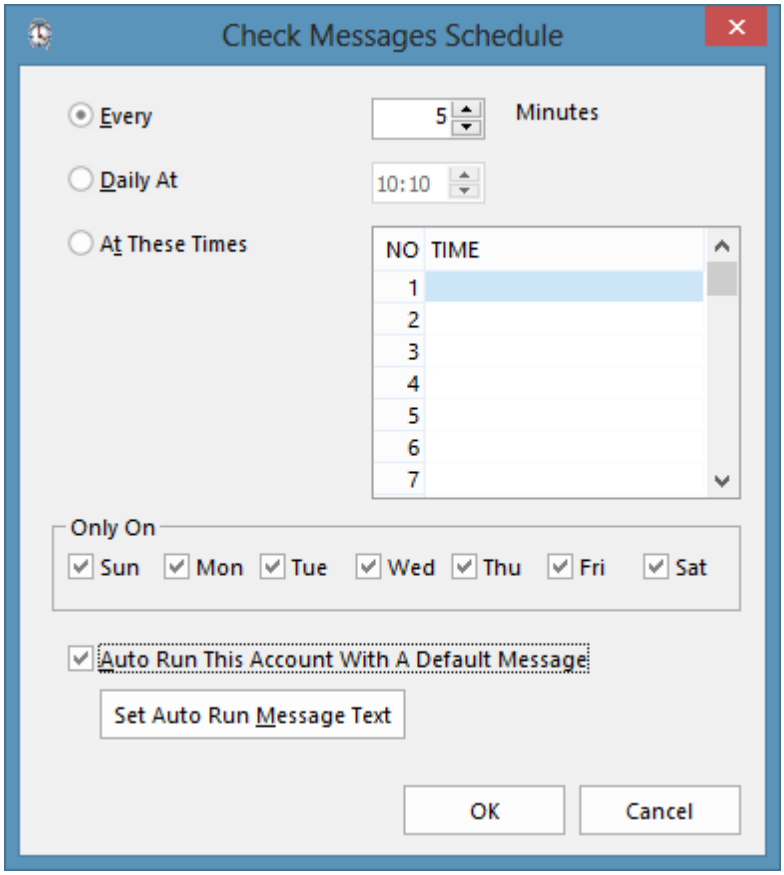
For any message where no to address or subject can be obtained then the values specified in the **Default To Address** and **Default Subject** will be used.

See Also: The Email2DB Client Application

## 6.3 Auto Running

Email2DB can also execute Triggers at predefined intervals without having to read an incoming message. This is useful in instances where you want to execute a Trigger to do some actions at set times of the day - regardless of whether new messages exist.

This is done by enabling the **Auto Run This Account With A Default Message** option on the **Account Schedule** form:




If this option is enabled then Email2DB will create a new message at the Account schedule. This message will be added to the Edge Queue and be processed by the Message Processor - thus executing the Triggers for the Account.

You can define the message text by clicking the **Set Auto Run Message Text** button. This is optional - as you may not need to actually extract data from a message when executing Triggers in this way.

## 6.4 Outbox Processing

You can create a special Account that can be used to process emails sent by Email2DB (as opposed to messages it retrieves from various sources). This allows you to execute actions on outgoing messages.

You can define a single Email2DB Account as the 'Outbox' account.

On the **Account Settings** option of the Account properties, select the **Outbox** option. The other tabs then disappear.

Only one Account can be defined as the Outbox.

You can now create message Triggers as normal within this Account.

Any email messages that Email2DB sends as a result of Triggers executing on other accounts will be passed to this Account once they are sent.

*Note: If any Triggers in the Outbox account themselves send outgoing emails, then these will not be processed by the Outbox account (since this would cause an infinite loop).*

### Saving All Sent Messages To The Sent Items Folder In The Message Store

The Email2DB Message store contains a folder called 'Sent Items'. You can setup a Trigger in the Outbox so that all sent messages are saved to this folder.

To do this, follow these steps.

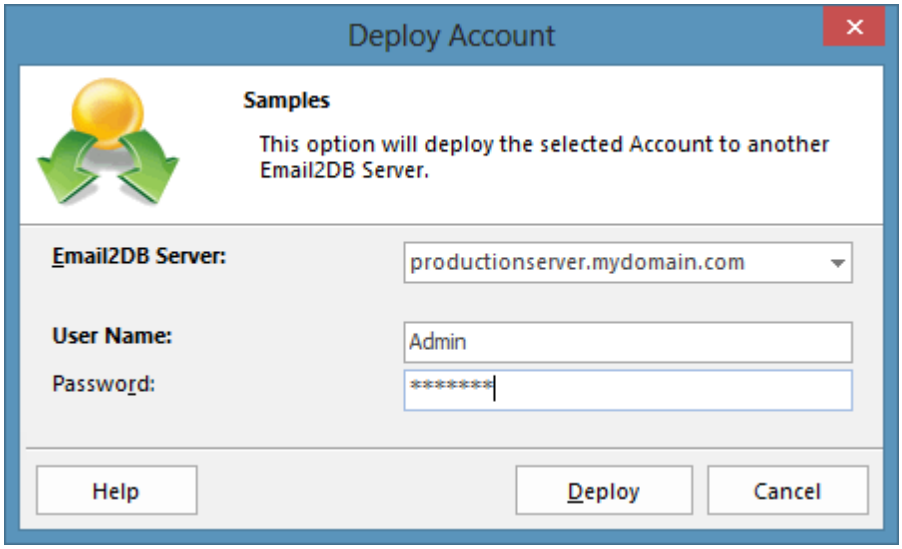
1. Create the Outbox Account.
2. Make sure the Store Full Copy Of Each Message In The Message Store option is enabled.
3. Create a new trigger in the Outbox Account. Call this 'All' and leave all the trigger conditions as their defaults .
4. On the Save To tab of the Trigger, select the Sent Items folder.

You could create additional folders or sub-folders in the Message Store and save sent messages to different folders using different Triggers with different Trigger conditions.

*Note: Outbox Processing is not available with Email2DB Hosted Edition.*

## 6.5 Deploying Accounts To Other Servers

You can create an Account on one Email2DB computer and then deploy it to another Email2DB installation running on a different computer. This enables you to develop and test an Account & it's Triggers on a development computer before deploying it to your production Email2DB Server.




Right-click any of your Accounts and select **Deploy To** from the popup menu.

*Note: Only users logged into Email2DB with Administrator rights will be allowed to do this.*

Enter the **Email2DB Server** that you want to deploy the selected Account to. This is the IP address or host name of the remote Email2DB Server.

Enter the **User Name** & **Password** for a valid user on the selected server.

Click **Deploy** to upload the Account.

The Account and all it's Triggers will then be uploaded to the selected Email2DB Server. If the Account already exists on the selected Email2DB Server then it will be updated, otherwise a new Account will be created.

# Part

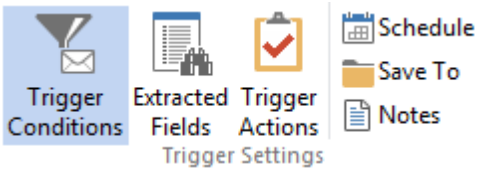
# 7 Message Triggers

Once you have setup an Account that defines 'where' Email2DB will read messages from, you can create any number of 'Triggers' that define 'if' a message should be processed, and 'what' to do with it.

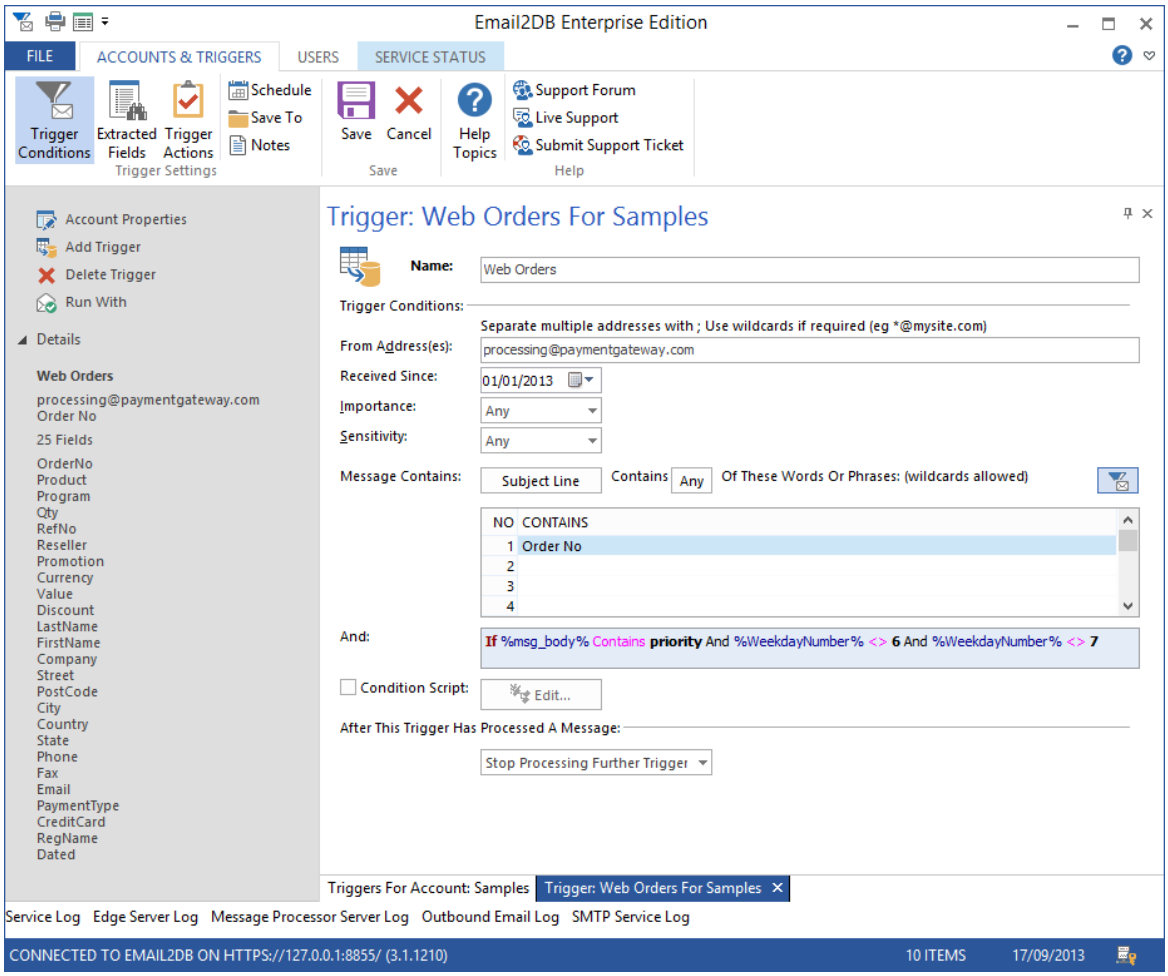
A Trigger is a set of rules that Email2DB applies to incoming messages. If the message matches these rules then Email2DB executes a set of 'Actions' against the trigger, if not Email2DB moves to the next Trigger for the Account and compares that. By default new Triggers will accept ALL new messages received by the Account unless you specify any 'Trigger Conditions'.

To create a trigger first double-click the account to open it. Now click the **Add Trigger** button.

The Trigger form will be shown. The Ribbon bar contains:




Click **Trigger Conditions** to define the Name and Trigger Conditions for this Trigger.




Enter a **Name** for the Trigger.

## Trigger Conditions

In the Trigger Conditions section you specify criteria that Email2DB checks before it executes any Actions.

### Enabled

Use this option to enable or disable processing of emails for this Trigger. This can be useful during testing.. allowing you to create a number of separate Triggers and enabling just the triggers you want to test.

### From Address(es):

If you only want messages from certain addresses checked enter the addresses here. Separate each address with a semi-colon. Wildcards are allowed, for example: *@mysite.com would cause any message from mysite.com domain to be checked.

### Received Since

Specify a date you want messages to be processed from. When a new message is received Email2DB will check the message date. Only messages with a date equal or greater than this date will be processed. When you create a new Trigger this date defaults to the current date. You should change it to a past date if you want to include old messages that may already exist from your message source.

### Importance/Sensitivity

If you only want messages flagged with certain importance or sensitivity level then select it here.

### Message Contains

Enter a list of words or phrases that Email2DB will check for in the subject line and/or message body. If any of the words/phrases are found then will the message be processed.

Click the **Subject Line** button to toggle between checking the Subject only and check both the Subject AND Body text.

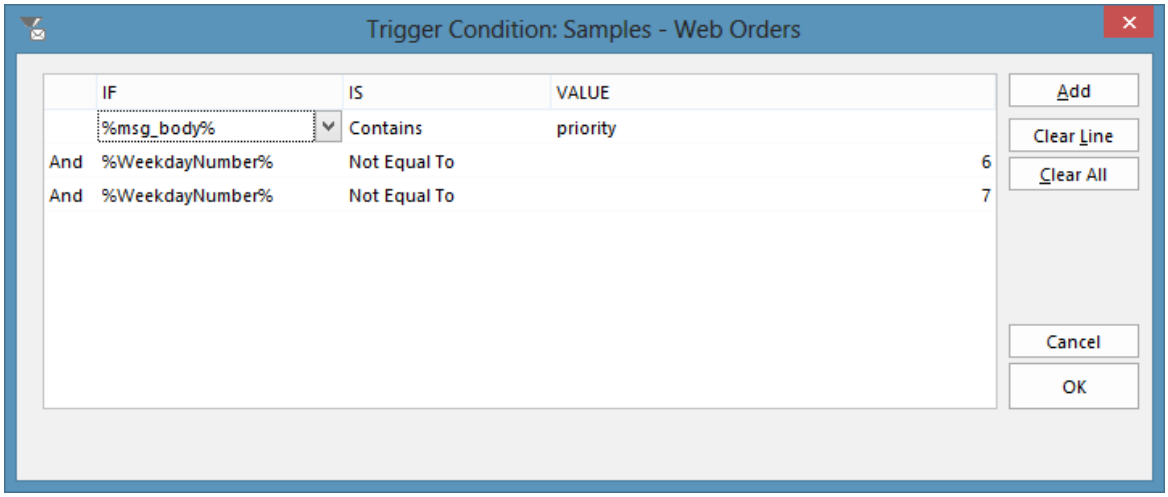
Click the **ANY** button to change the condition to **ALL**. Then ALL words/phrases must exist for the message to be processed instead of ANY of them.

Each line can contain its own word or phase and you can use regular expressions. Checks are NOT case sensitive.

*Note: Words & phrases can also be NEGATIVE by preceding the word or phrase line with '---'*

## Condition Builder

The  button will open the **Condition Builder**. The Condition Builder allows you to define many more conditional checks that can be performed on the incoming message.




Using the Condition Builder you can construct an IF statement to apply to the incoming message.

In the **If** column you select a message property.

In the **Is** column you select one of the following:

- Equal To
- Not Equal To
- Less Than
- Greater Than
- Less Than Or Equal To
- Greater Than Or Equal To
- Is Blank
- Is Not Blank
- Contains
- Does Not Contain
- Starts With
- Length Equal
- Length Less Than
- Length Greater Than

In the Value column you can type a value to compare against.

Click the **Add** button to add another line. The new line can be assigned as an AND or OR clause.

## Condition Script

You can also write an Email2DB Basic Script to fine tune the trigger conditions. You can use this to check message headers or create custom rules. See: Trigger Condition Scripts

## After This Trigger Has Processed A Message

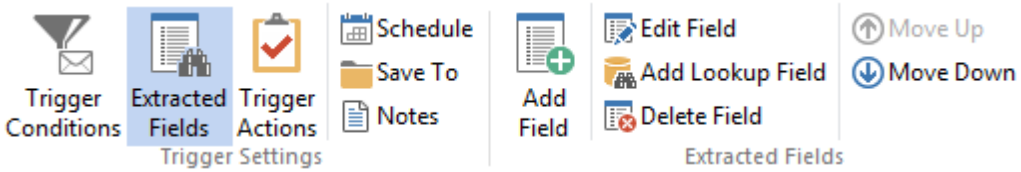
In this section you define what happens after the Trigger has executed because it matched an incoming message. By default Email2DB will stop processing further triggers for the message. You can select for another trigger to be checked and processed.

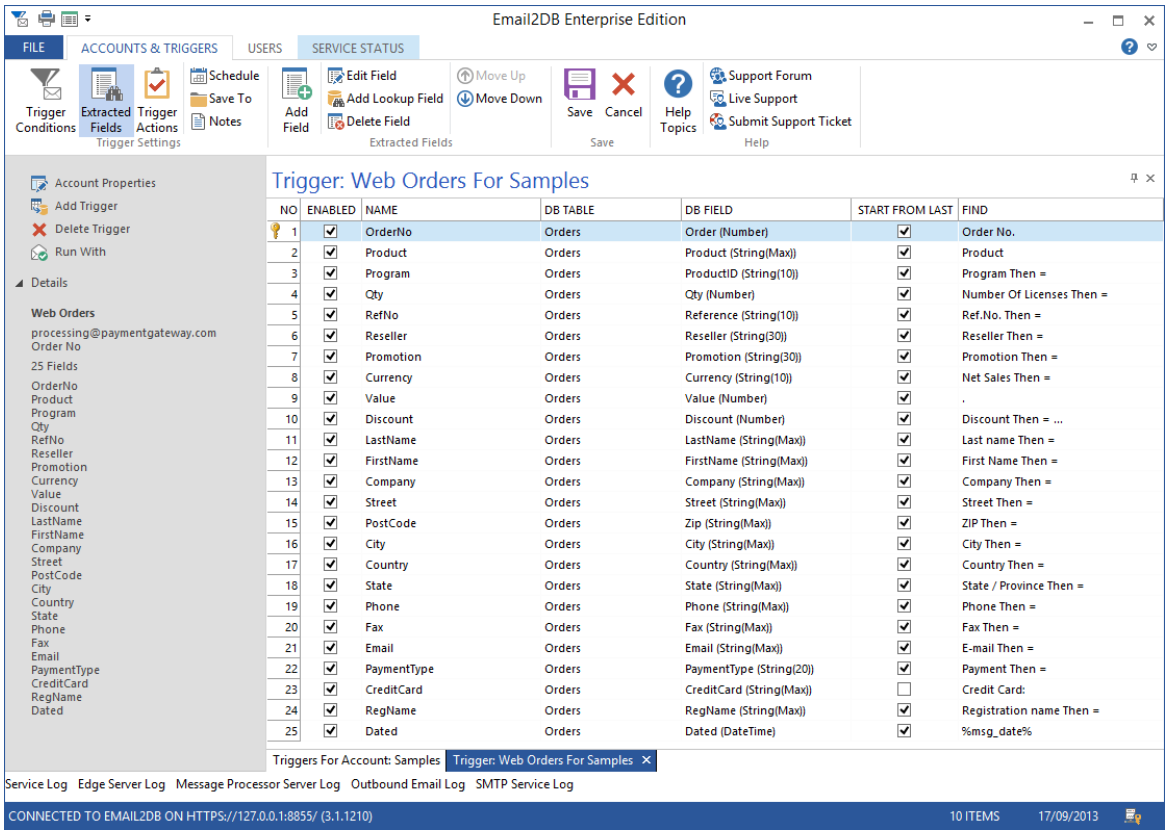
Once you have defined the trigger conditions you can then setup the fields to extract from the email.

See Also: Extracting Fields

# 7.1 Extracting Fields

Click the **Extracted Fields** option to enter a list of fields that Email2DB will extract from messages that match the Trigger Conditions.



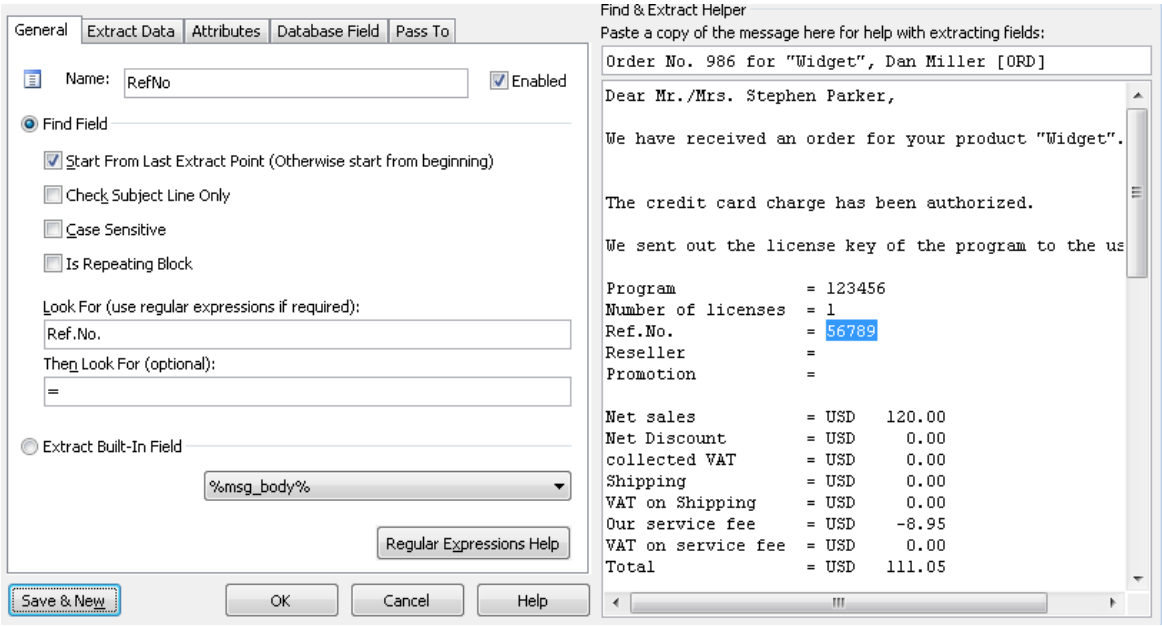

For each Trigger you can specify any number of 'Fields'. A Field is a distinct piece of data that Email2DB will extract from the message and assign it a 'Field Name'.

*Note: For Email messages, Email2DB will also scan and parse any text or XML attachments in addition to the body text.*

Once all fields have been extracted Email2DB can update your database and perform other Actions.

To create a new field click the **Add Field** button.

## Finding Fields




Before creating any fields it is a good idea to paste a copy of the message subject and body you are extracting data from into the **Find & Extract Helper** boxes. Email2DB will then highlight the data it will extract for each field as you specify the extraction properties. The message Subject and body will be saved with the Trigger - so you only need to paste it once.

Enter a **Name** for the field.

In the **Find Field** section you specify how Email2DB will find the field from the message.

### Start From Last Extract Point

Normally Email2DB moves the 'extraction point' as it moves through the message extracting data. Uncheck this option if you want Email2DB to start from the beginning of the message when it looks for this field. Once a field is extracted the extraction point will be set to the end of the field. The next field extraction starts from this point unless you uncheck this option.

### Check Subject Line Only

Normally Email2DB checks both the subject line and message text. Uncheck this option if you want Email2DB to only check the subject line text when it looks for this field.

### Case Sensitive

By default Email2DB ignores case when it looks for fields. So 'order number' and 'Order Number' with both match when searching for 'Order number'. Uncheck this option if you want to perform a case sensitive search.

### Is Repeating Block

This option sets the field as a 'repeating block'. If set Email2DB will enter a loop and repeat the extraction of this field until it finds the next field to extract (or the end of the message). See: Repeating Blocks for more information.

### Look For

Enter the text that Email2DB should look for when searching the message for this field. This should be the text that is the same for each mail and uniquely identifies the field.

### Then Look For

Enter an additional text string that Email2DB looks for AFTER it has found the above text. This is optional but is useful when data is formatted in the message using an unknown number of TABS or spaces. Consider the following
line:

```
Customer code                          : ABC
```

We would search for 'Customer code' and then ':' because we don't know how many spaces are between 'Customer code' and ':'. The field extraction would then start after the ':'.

In both the 'Look For' and 'Then Look For' entries you can make use of Regular Expressions to assist with searching.

### Control Characters

In both the Look For and Then Look For fields you can include a number of control characters in addition to regular expressions:

| | |
|---|---|
| <CR> | Carriage return |
| <LF> | Line feed |
| <CRLF> | Carriage return/line feed |
| <TAB> | Tab |
| <ESC> | Escape |
| * | When used on its own the * character finds the next non-space or non-control character. |
| <xxx> | Where xxx is the ASCII character code |

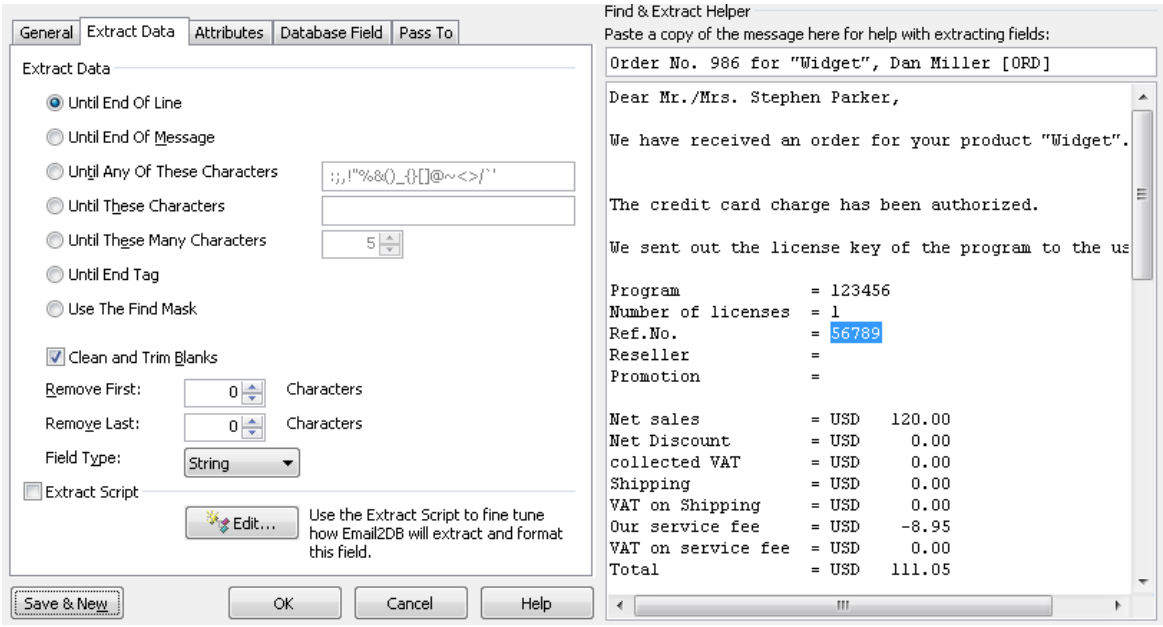
This can be useful when searching for data. For example, suppose the text contains:

```
Your serial number is:
1234-5678
```

We want to extract the serial number, so we could look for 'Your serial number is:' and then look for '<CRLF>' - because the serial number is on the next line. Another way of doing the above would be to look for 'Your serial number is:' and then look for '*' - which would effectively look for anything after 'Your serial number is:'.

## Extracting Data

Click the **Extract Data** tab to define how Email2DB will extract data for this field once it has found it.




There are a number of options you can use to extract data from the message (all options start the extraction after the 'Look For', and optionally 'Then Look For' text):

**Until End Of Line**

Extract all data up to the end of the line (or the end of the message if there are no more lines).

**Until End Of Message**

Extract all data up to the end of the message.

**Until Any Of These Characters**

Extract data until any of the following characters are found. You can then specify a list of characters to search for. If any one of the characters are found then extracting will stop.

**Until These Characters**

Extract data until specific words or characters are found. You can then specify characters, words or phases to search for. Extraction will stop when the words are found. Regular expressions permitted.

**Until These Many Characters**

You can manually specify a number of characters to extract.

**Until End Tag**

Select this option if you are extracting HTML or XML tags. If the 'Look For' value is a tag, for example: <mytag> then Email2DB will extract up to the end tag </mytag>. This option will be automatically selected on new fields if you enter a tag in the 'Look For' entry.

**Use The Find Mask**

Select this option to extract the field INCLUDING the 'Find' mask. This is useful when you want to find AND extract using a regular expression. The data would be extracted starting from the 'Find' value. For example, if the 'Look For' is set to the regular expression: "[a-zA-Z0-9._-]+@[a-zA-Z0-9_-]+\.[a-zA-Z.]+" (which is the regular expression for an email address) and the 'Use The Find Mask' option is

selected.. then the first Email address will be found AND extracted. If the 'Use The Find Mask' is not selected then the first email address will be found and extraction will start AFTER the end of the email address.

### Clean And Trim Blanks

Enable this option if you want the extracted data to be cleaned and trimmed. This will remove any spaces, tabs or carriage return/line feed characters from the field data.

### Remove First/Last

You can also select to remove a number of characters from the beginning and end of the extracted data.

### Field Type

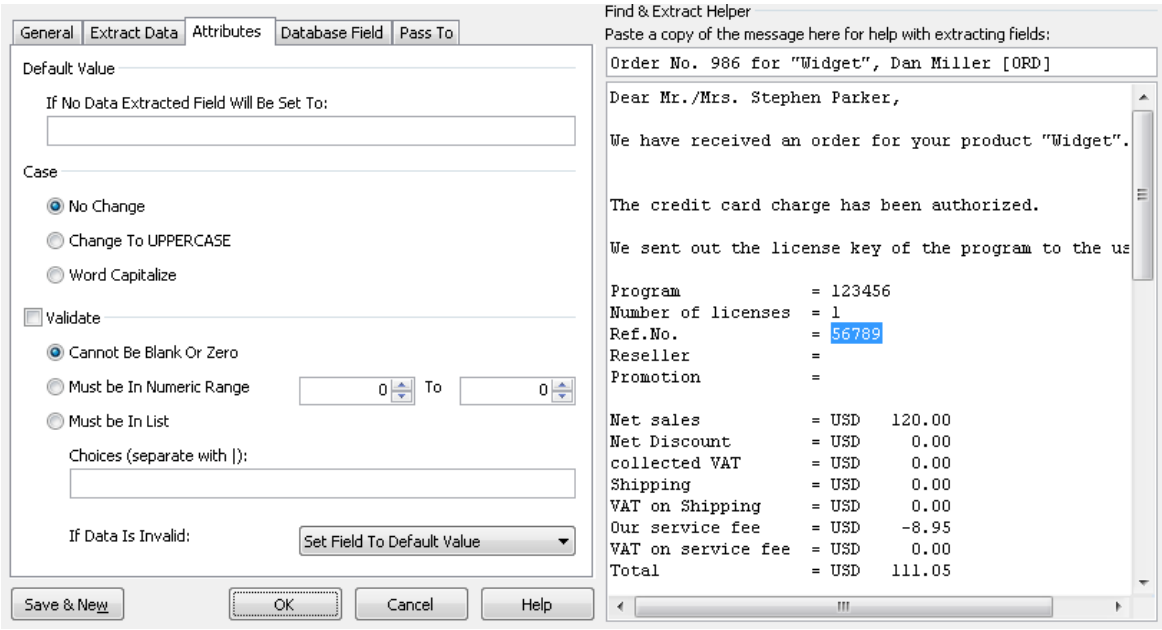
You can select the type of field from this drop down menu. If you know the data is always numeric, for example, then select 'Numeric', otherwise leave as 'String'.

### Extract Script

Enable this option if you want to write a custom Email2DB Basic script to extract the field data from the message. This gives complete control over how you extract the field. You can also access all the Headers from the message. See: Field Extraction Scripts

## Additional Attributes

Select the **Attributes** tab to define optional additional attributes for the field.




### Default Value

Enter a value that will be assigned to the field if no data is found or extracted from the message text.

### Case

This option allows you to change the extracted value to UPPER CASE or to apply Word Capitalization.

### Validate

In this section you can define validation rules for the extracted field and you can define what action Email2DB should take if the extracted data is invalid.

Select the Validate option to enable validation for this field. Select Cannot Be Blank Or Zero option if the field must be a value (or be non-zero in the case of numeric fields). For numeric fields you can also select a valid Numeric Range. The Must Be In List option allows you to define a list of valid values. In the Choices entry specify the list of valid values for the field. Separate each value with a | (pipe) character. For example, if the field must be Y or N - specify 'Y|N'

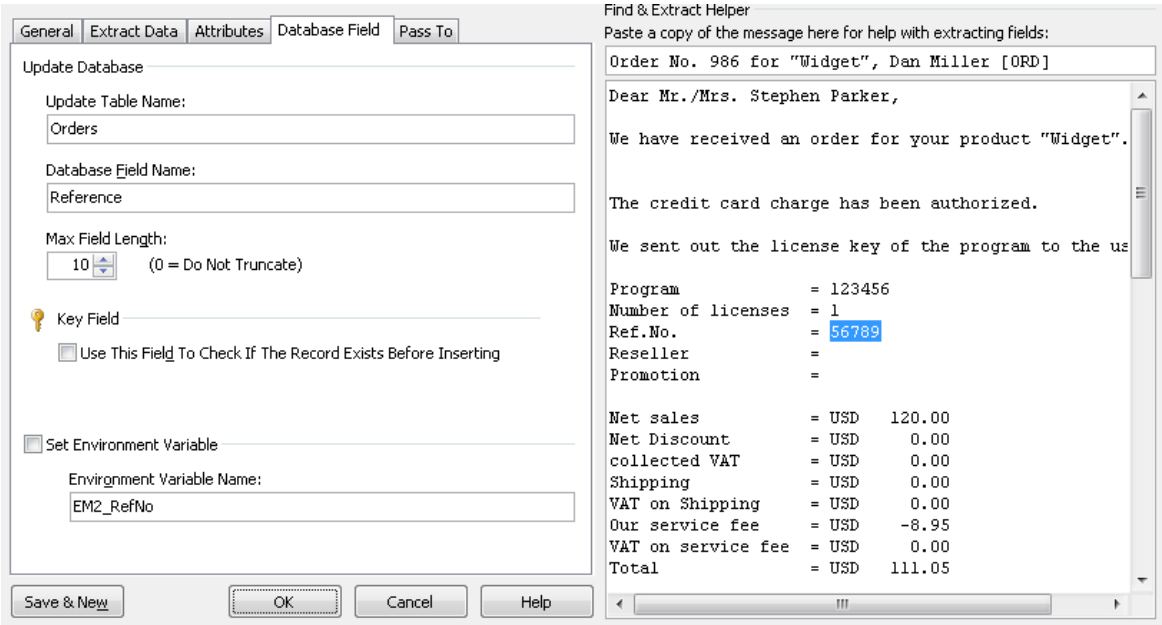
### If Data Is Invalid

Here you specify what Email2DB should do if the extracted field data is invalid. There are two options:

1. Set Field To Default Value - select this option if you want Email2DB to replace the extracted data with the field's default value (or blank if no default it specified).
2. Cancel Trigger - select this option if you want Email2DB to cancel execution of the trigger for the current message.

## Mapping To Database Fields

Select the **Database Field** tab to map the extracted field to a field in your database that you want Email2DB to update.




Enter the **Table Name** that the field will be updated on. This table must already exist in your database. Multiple fields can use different table names if required - but the tables must be part of the same database.

Enter the **Database Field Name** in the table that the Email2DB field will be mapped to.

### Max Field Length

You can optionally specify the maximum allowed field length for the database field. Email2DB will truncate the field before updating the database if the extracted data is greater than the maximum length. Use this option to avoid database errors that will be raised when field data is inserted into

your database that is greater than the defined length. This option applies to String field types only. Set to zero if you do not want Email2DB to truncate the field.

**Key Field**

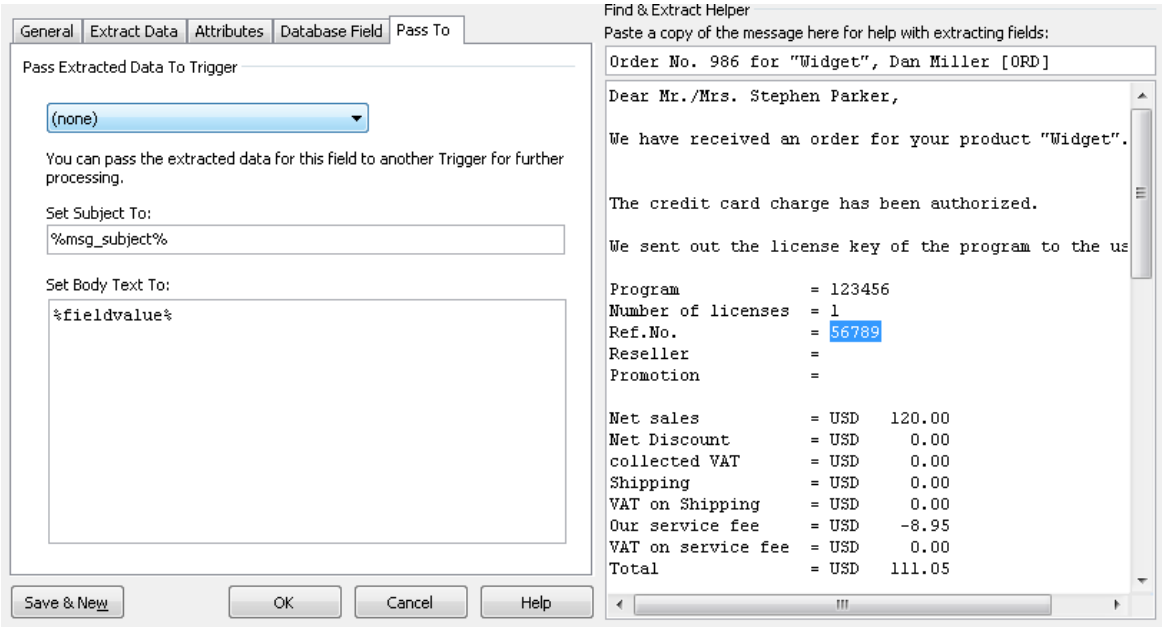
Check this box if this field is a Key Field. Key fields allow you to control how your database is updated. If you create one or more key fields, Email2DB will first check if a record exists in your database using the key field values. If a record already exists then the existing record will be UPDATED otherwise a new record will be INSERTED.

**Environment Variables**

Email2DB can also set environment variables as fields are extracted. This is useful if one of the Actions you are using is 'Run Process'. Email2DB will set the environment variable to the value of the field before the process is run. Your external process can then read the environment variables and act on them.

## Pass To

Select the **Pass To** tab if you want to pass the extracted field value to another trigger for further processing. This is useful for complex messages, allowing you to break up processing.




Select the **Trigger** to pass the field data to from the drop down list.

If you use this option, Email2DB will extract the field data as normal. It will then use the extracted field value as the message body text of a new message and pass it to the selected trigger. This trigger will execute independently as if it had received a new message. The headers of the passed message will be set to the headers of the original.

By default the Subject of the pass to message will be set to the subject of the original message. You can change the subject line in the **Set Subject To** entry.

The Body of the message will be set to the value of the extracted field. You can change this in the **Set Body Text To** entry. The %fieldvalue% placeholder contains the value of the current extracted field. You can enter any other text and use previously extracted field values.

For example:

```
Order No: %orderno%

%fieldvalue%
```

Would pass the previously extracted field 'OrderNo' along with a blank line and then the current extracted field. The Trigger that receives this message can then extract both values just like any other incoming message.

See Also: Repeating Blocks
See Also: Lookup Fields

### 7.1.2 Repeating Blocks

Often messages contain repeating sections. Most of the time you need to run a process or update a database for each individual section of the repeating section. Email2DB allows you to do this by defining a field as a 'repeating block'. When a field is defined as a repeating block, Email2DB will enter a loop and extract each block of the repeating section in turn. This is then used on conjunction with the 'pass to' trigger option - allowing you to pass the individual block to another trigger for processing.

For example:

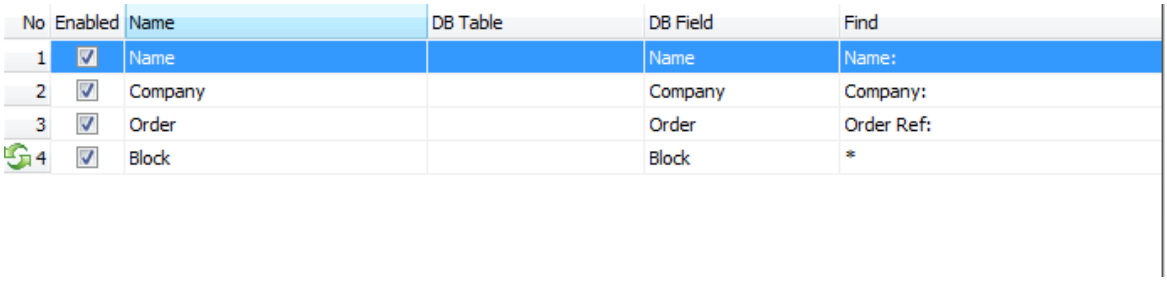
Suppose you receive the following email:

```
Name: Stephen
Company: PSL
Order Ref: 1234

Product: WHO1
Qty: 1

Product: WHO2
Qty: 2
```

The Product and Qty fields can repeat any number of times depending on what the customer has ordered.

We can define the Product & Qty fields as a single field and set them as a repeating block. The repeating block can then be passed to another trigger for processing on it's own.

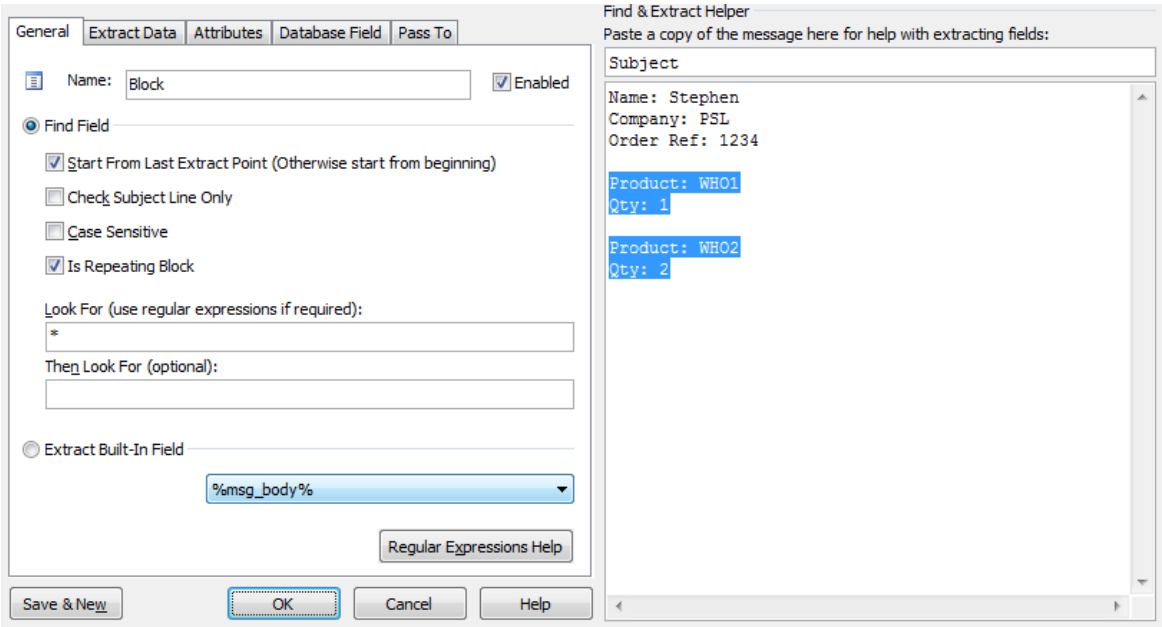
We would define the extraction of the above fields as follows:



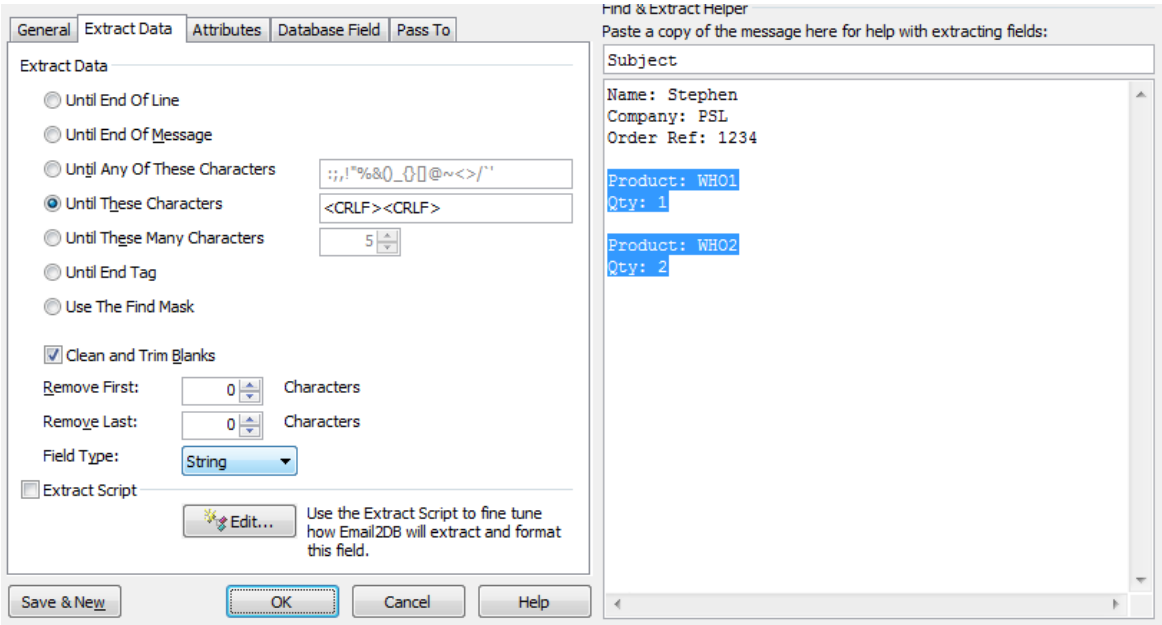
| No | Enabled | Name | DB Table | DB Field | Find |
|---|---|---|---|---|---|
| 1 | ☑ | Name | | Name | Name: |
| 2 | ☑ | Company | | Company | Company: |
| 3 | ☑ | Order | | Order | Order Ref: |
| 4 | ☑ | Block | | Block | * |

The Name, Company and Order fields we extract by looking for the Name:, Company: and Order Ref: field headers and extracting until the end of the line.

For the Block field we set the Look For to '*' - which means start from the next character after the last extract point:



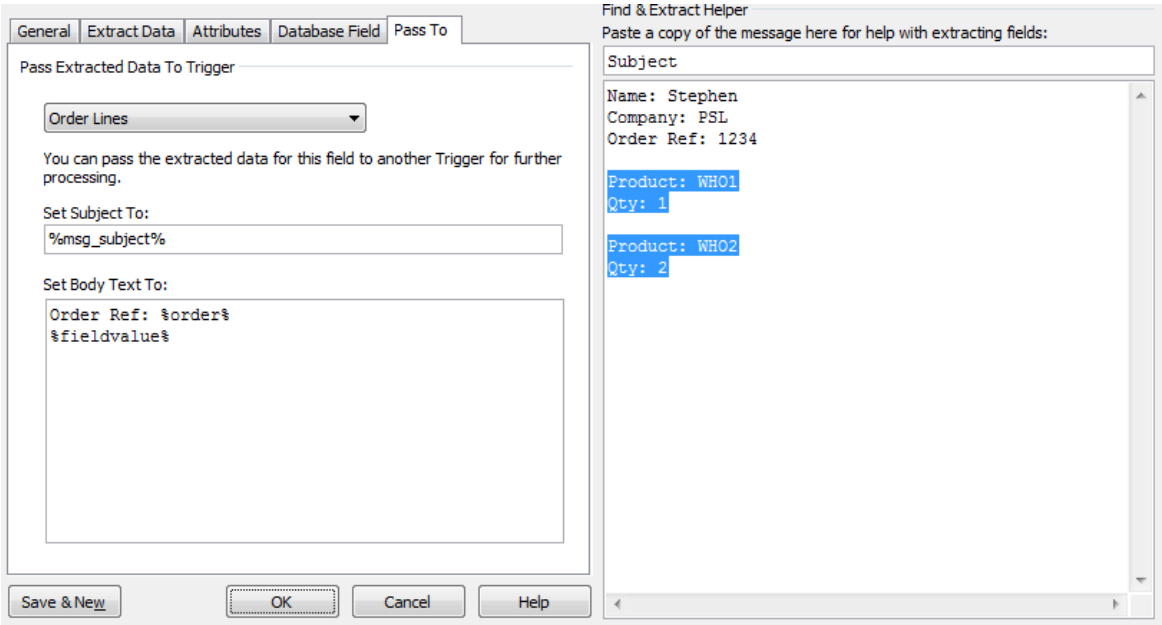

We set the **Extract Data** option to **Until These Characters** - and set this to <CRLF><CRLF>. This basically means 'the next blank line'.




You then enable the **'Is Repeating Block'** option on this field. Email2DB will then repeat the field extraction until it either finds the next field in the extracted fields list or the end of the message.

We then create another Trigger. In this case - called 'Order Lines'. This trigger will receive each section of the repeating block as a new message. This trigger can extract the Product and Qty fields and perform further processing and update a database which each product ordered.

On the **'Pass To'** tab of the repeating block field we select this new trigger:




In the **Set Body Text To** field we also pass the Order Ref: extracted field. This allows us to use this value in addition to the block section in our 'pass to' trigger.

The Order Lines pass to trigger would receive 2 messages in the above example. The messages would be:

```
Order Ref: 1234
Product: WHO1
Qty: 1
```

and then

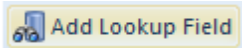
```
Order Ref: 1234
Product: WHO2
Qty: 2
```

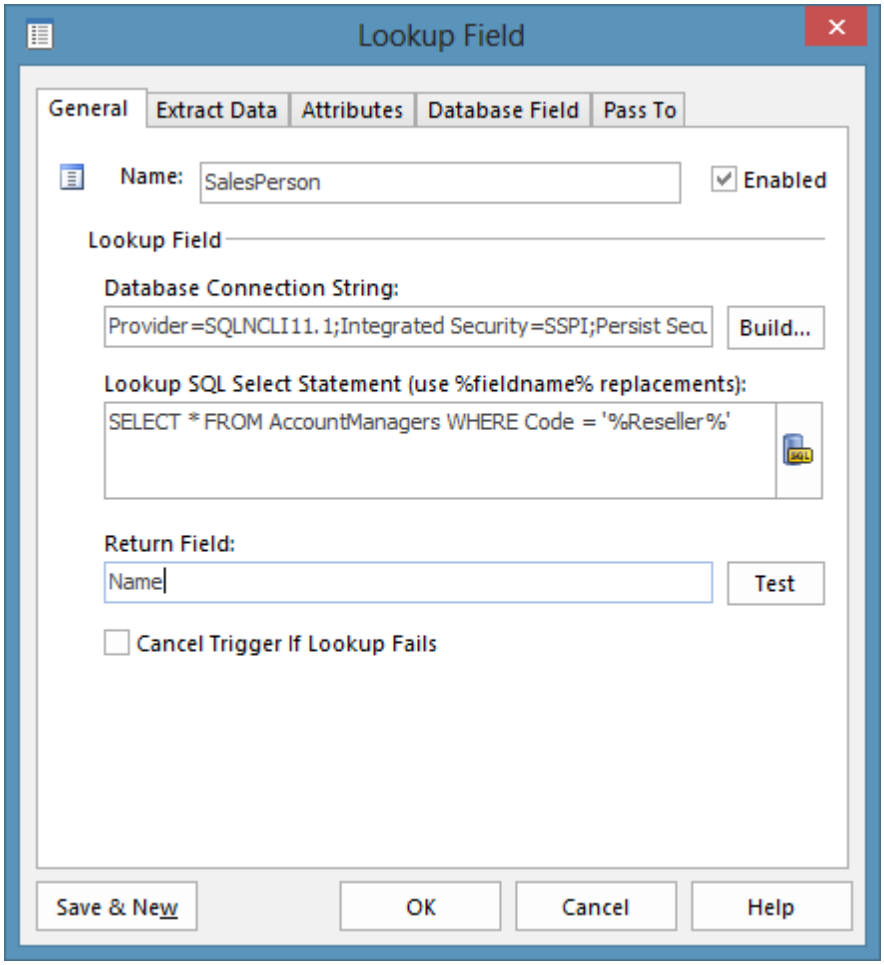
### 7.1.3 Lookup Fields

In addition to extracting fields from the incoming messages, Email2DB allows you to create lookup fields. Lookup fields are used to read data from external databases.

For example, you may have extracted the Field 'Salesman Code' from the incoming message. You could create a lookup field to lookup the Salesman name from a table in your database.

To create a lookup field click the **Add Lookup Field** button on the Ribbon bar:




Enter the Field **Name**.

You must then specify the **Database Connection String** to the database that you want to lookup data from. Click the **Build** button to build a connection string. You can lookup data from any database accessible on your computer that has an ODBC or ADO provider (such as SQL Server, Access, MySQL etc). See Also: Connection Strings

Enter the **SQL Select Statement** to use to retrieve the data. Use %fieldname% replacements in the WHERE clause if you want to find a record using data from previously extracted fields.

In the **Return Field** entry enter the field name to return, or leave blank to return the first field.

When Email2DB executes the lookup statement it will return the field value for the given field from the first record that matches the WHERE clause of the SELECT statement. The value will then be assigned to the Email2DB field.

Enable the **Cancel Trigger If Lookup Fails** option if you want Email2DB to abort processing of the trigger if no record is returned by the lookup. Email2DB will then stop processing the incoming email as if the trigger conditions where not met.

Once the field value is read from the database, you can apply the same Extraction properties, Attributes, Database Field and Pass To properties as with normally extracted fields.

### 7.1.4 Dealing With HTML Only Emails

Messages retrieved in HTML format are fine for viewing, but not really suitable for extracting data. Fortunately most email clients will also create a 'plain text' portion which is a version of the HTML message without any of the HTML tags. Email2DB will automatically use the plain text portion of the message for extracting - if it exists.

#### What If There Is No Plain Text?

If the plain text portion of the message does not exist, Email2DB will create one by removing all the HTML tags itself. However the resulting plain text portion may need additional extraction options depending on how the original message was formatted.

To be able to set up your extraction fields properly, use the **Find & Extract Helper** field in the **Field Extraction** form. Paste the complete HTML source code of the message into this entry. Email2DB will then automatically convert it to plain text. The resulting plain text will be how Email2DB will 'see' it during field extraction.

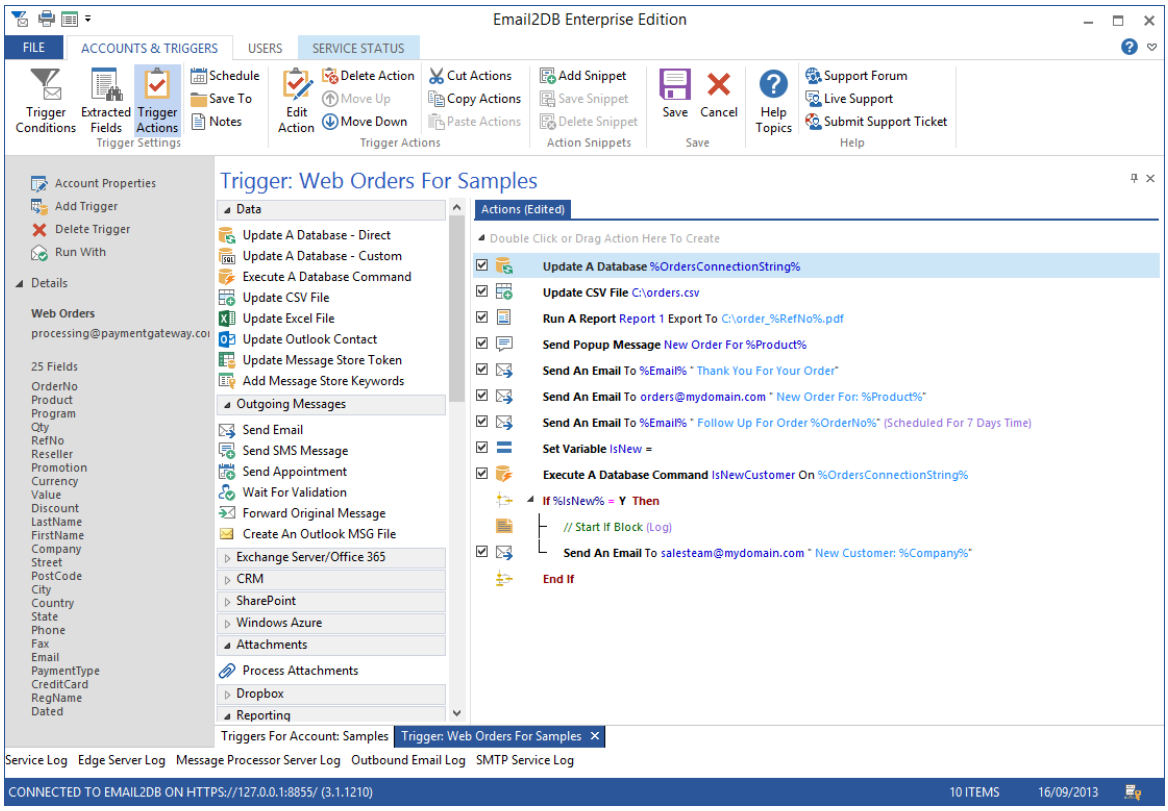
#### Extracting From HTML Manually Using Scripts

The field MSG_Html will contain the HTML portion of the message (whilst the MSG_Body field will contain the plain text). You can use this field in extraction scripts if you want to extract using specific HTML tags.

# Part VIII

# 8 Trigger Actions

Click the **Trigger Actions** option on the Ribbon bar to define the additional actions that you want the Trigger to perform after it has extracted your fields.




Each of the Actions are optional. You can run one or more of the actions depending on your requirements. Each Trigger can contain any number of Actions. You can add multiple instances of each Action type - so for example, you can send multiple Outgoing emails by adding multiple 'Send Email' actions.

To add an Action double click an action in the **Toolbox** or drag and drop an action to the **Actions List**. When you add an Action the corresponding property page for the Action type will be displayed. To edit an existing action simply double-click it or press Enter.

When a Message is processed by Email2DB the Actions of the matching Trigger will be executed in the order they appear in the Actions list. You can re-order actions using **Move Up**/**Move Down** buttons.

See Also: Using Field Replacements
See Also: Conditional Execution
See Also: Available Action Types

# 8.1 Using Field Replacements

All actions that have parameters allow you to use %fieldname% replacements. This means you can specify an Email2DB field name, variable or constant in the parameter that will be replaced at run time with the value of the field. Field names must be enclosed with % symbols.

For example, outgoing emails can contain values extracted from the incoming message by simply inserting %fieldname% (where 'fieldname' is the name of your field) into the email text (or any other fields).

Another example: The 'Attachments' action lets you save attachments from the incoming message to folders on your system. If you had created a field called 'OrderNumber' you could save the attachment as '%OrderNumber%.pdf'.

## Built-In Fields

In addition to extracted fields or variables that you create yourself, you can also use any of the following built-in fields:

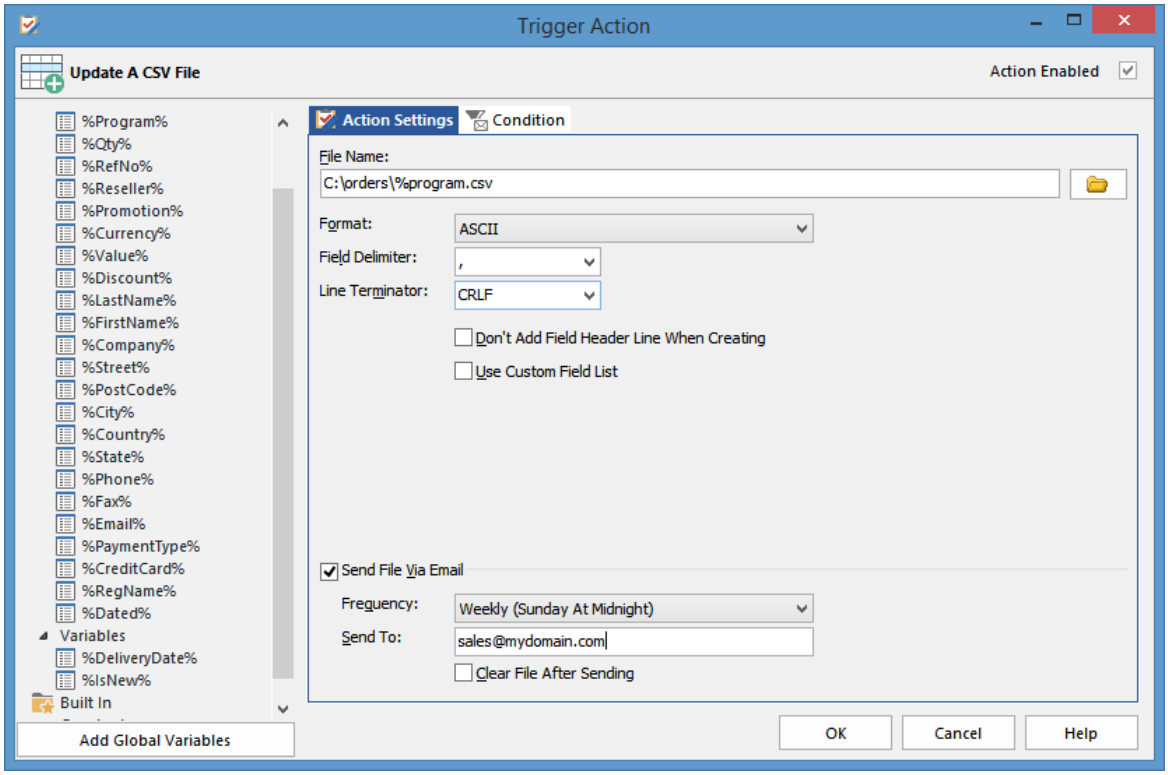
| | |
|---|---|
| %msg_body% | The original plain text message. (If the message is HTML and has no plain text, Email2DB will create a plain text version and store it here). |
| %msg_html% | The HTML version of the original message (if it has one). |
| %msg_mimetext% | The complete mime text of the message (including attachments). |
| %msg_subject% | The original message subject. |
| %msg_to% | The 'To' email address(s). Just the email addresses separated by commas. Name portions are removed. |
| %msg_towithnames% | The To addresses including names (if available). |
| %msg_from% | The 'From' address. |
| %msg_fromname% | The 'From Name' if available. |
| %msg_fromip% | The IP address of the sender. |
| %msg_cc% | The 'CC' addresses. Just the email addresses separated by commas. Name portions are removed. |
| %msg_ccwithnames% | The 'CC' addresses including names (if available). |
| %msg_bcc% | The 'BCC' addresses. Just the email addresses separated by commas. Name portions are removed. |
| %msg_bccwithnames% | The 'BCC' addresses including names (if available). |
| %msg_replyto% | The 'Reply To' address. |
| %msg_inreplyto% | The 'In Reply To' header. |
| %msg_references% | The 'References' header. |
| %msg_return-path% | The 'Return-Path header. |
| %msg_sender% | The 'Sender' header field. |
| %msg_date% | The date of the message. |
| %msg_datesent% | The date the message was originally sent. |
| %msg_size% | The total size of the message including attachments (in bytes). |
| %msg_headers% | The complete headers section of the email. |
| %msg_attachments% | A string containing each attachment file name (separated by CrLf). |
| %msg_attachmentcount% | The number of attachments. |
| %msg_attachmentinlin | The number of inline attachments. |

| | |
|---|---|
| ecount% | |
| %msg_attachmentssavedto% | A string containing the file system location of each saved attachment (if Attachment saving action used). |
| %msg_attachmentlistwithsizes% | A string containing each attachment & size separated by commas eg: 'document1.pdf (2mb), document2.pdf (500kb)' |
| %msg_charset% | The character set used on the email. |
| %msg_messageid% | The Unique message identifier. |
| %msg_geoipcity% | The GEO-IP City data for the senders IP address. |
| %msg_geoipcountry% | The GEO-IP Country data for the senders IP address. |
| %msg_geoiporganization% | The GEO-IP Organization data for the senders IP address. |
| %msg_wordindex% | Unique keywords extracted from the message body (space separated excluding common words). |
| %msg_wordindexsorted% | Unique keywords sorted by word count (highest first). |
| %msg_wordindexwithcount% | Unique keywords with word count. |
| %msg_wordindexwithcountsorted% | Unique keywords with word count sorted (highest first). |
| %msg_digest% | The first 250 characters of the plaintext body with all whitespace removed. |
| %msg_viewurl% | A link to the Email2DB web services viewmessage URL. |
| %msg_validationurl% | A link to the validation web page for the message - used by the Wait For Validation action. |
| %date% | The current date. |
| %time% | The current time. |
| %hour% | The current hour (0-23). |
| %minute% | The current minute (0-59). |
| %datetime% | The current date & time. |
| %sqldate% | The current date in yyyy-mm-dd format. |
| %sqldatetime% | The current date & time in yyyy-mm-dd hh:mm:ss format. |
| %dayofweek% | The current day of week string. |
| %weekdaynumber% | The current day of the week number (1=Sunday, 2=Monday etc). |
| %monthnumber% | The current month number. |
| %monthname% | The current month name. |
| %year% | The current year number. |
| %lasterrornumber% | This will contain a numeric error number returned from an Action that has failed for some reason. |
| %lasterrortext% | This will contain a description of the last error. |
| %freediskspace% | The current system drive free space in bytes. |
| %root% | The root folder of the logged in Email2DB users file system (hosted edition). |
| %triggerlog% | The current trigger execution log for the message. |
| %accountname% | The name of the Email2DB Account associated with the current message. |
| %triggername% | The name of the Email2DB Trigger associated with the current message. |
| %organizationname% | The name of your Organization. |
| %organizationemail% | The default email address of your Organization. |

You can assign any of the above fields to fields you create in Email2DB. You can also use them directly in any Action parameters.

## Dropping Field Names Into Action Parameters

Each Action form includes the Fields Tree:



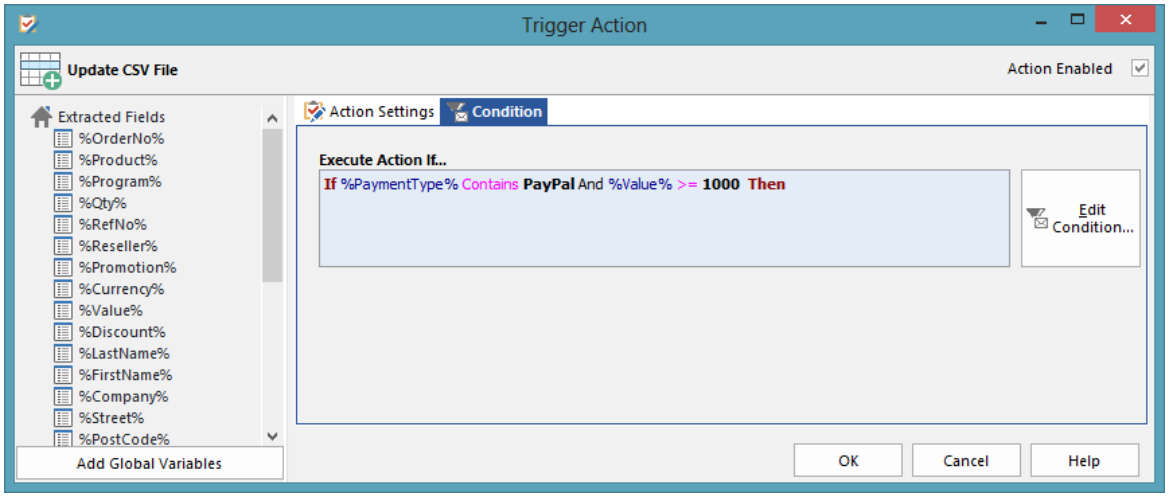

Drag and drop any field/variable or built-in field to any of the editable Action fields to use that field in the entry. At run time the %field% will be replaced with its value. You can also directly type any %field% or %variable% into any entry.

See Also: Organization Constants

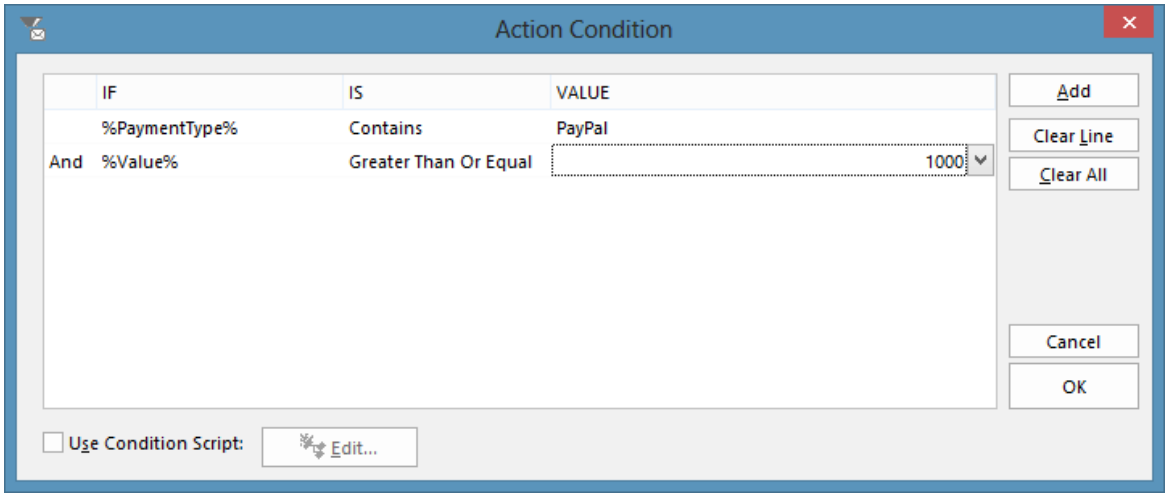
## 8.2 Conditional Execution

Against each Action you can define a Condition that controls execution of the Action.

On each Action form the **Condition** tab is used to define the condition:




Click the **Edit Condition** button to open the **Condition Builder**.




Using the Condition Builder you can construct an IF statement.

In the **If** column you select a message property or one of your Extracted Fields or Variables.

In the **Is** column you select one of the following:

- Equal To
- Not Equal To
- Less Than
- Greater Than
- Less Than Or Equal To
- Greater Than Or Equal To
- Is Blank
- Is Not Blank
- Contains
- Does Not Contain

- Starts With
- Length Equal
- Length Less Than
- Length Greater Than
- Is A Valid Email Address

In the **Value** column you can type a value to compare against or select one of your Extracted Fields or Variables.

Click the **Add** button to add another line. The new line can be assigned as an AND or OR clause.

## Condition Scripts

For even finer control over the conditional execution you can use a Condition Script. Click the **Use Condition Script** option and then click **Edit**. You can then create a script to control execution of the action. The script must set the variable Email2DBAction to True if you want the Action to execute. Both the Condition Builder statement AND the script must return True for the Action to be executed.

## 8.3 Fields & Variables

This topic describes the difference between Email2DB Extracted Fields and Email2DB Variables. Both Extracted Fields and Variables can be used in Action properties using the %fieldorvariablename% replacement syntax.

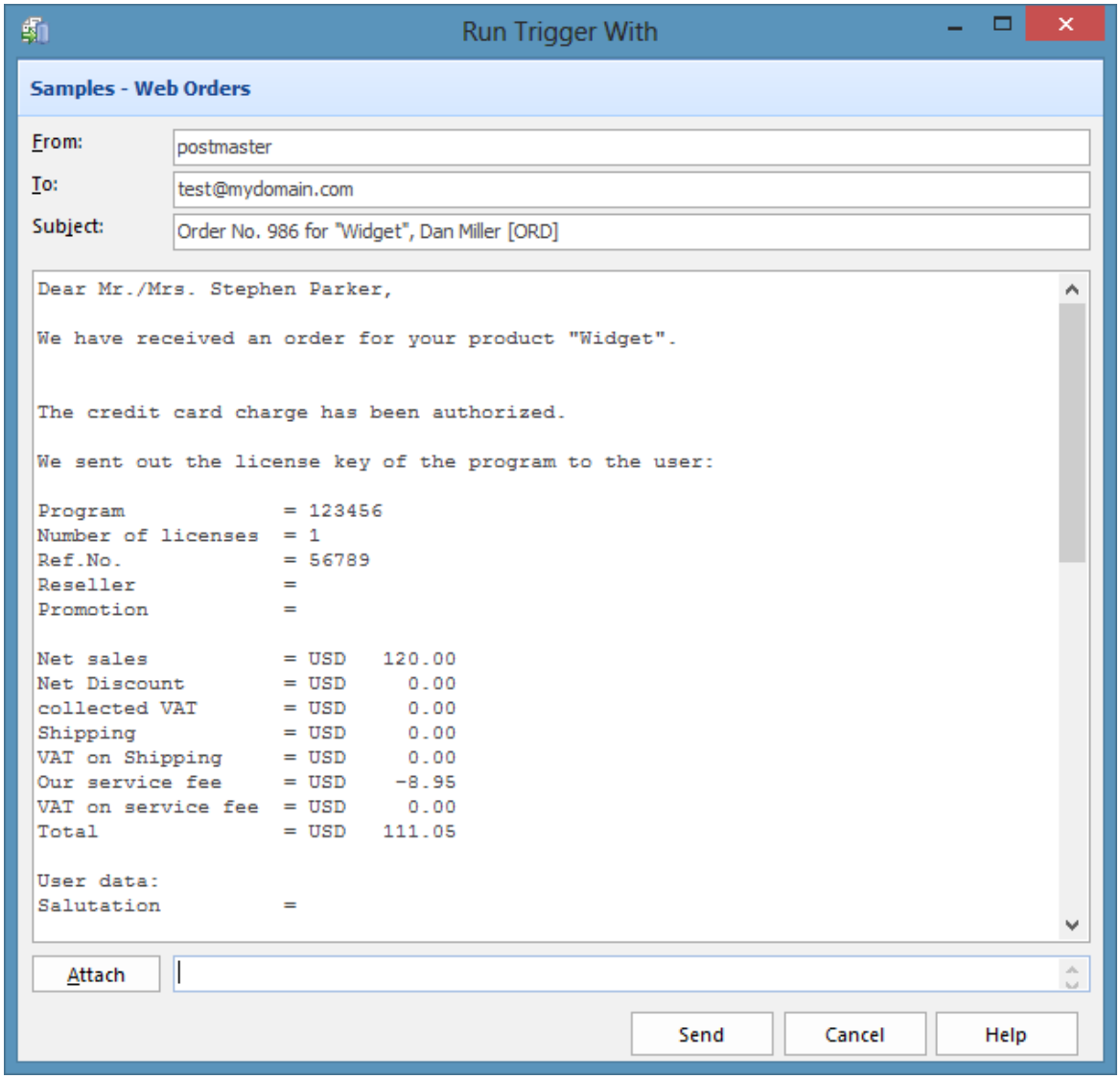
Extracted Fields are created to extract distinct data from the incoming message. At the start of the extraction process each Extracted Field is reset. Extracted Fields can be set to other values in **Script** Actions and using the **Set Variable** action. Extracted Fields are private to the current message being processed.

Variables are created using the **Set Variable** Action. Variables are shared with other Triggers in the same Account. So if you create a Variable called 'Counter' on two separate Triggers in the same account they will both share the same value. Variables also keep their value from message to message.

## 8.4 Running Tests

Once you have configured your Trigger you can send a test message to Email2DB to test Trigger processing.

Select your Trigger and click the **Run With** button on the Ribbon bar, or right-click a Trigger and select **Run With** from the popup menu.




The Run With form will be displayed.

All entries are optional - you can send a blank message if your Trigger does not need to extract anything.

Enter a **From** & **To** address. The From address will default to your email address entered on your Email2DB user details.

Enter a **Subject** and **Body**. These entries will default to the text entered in the Field Helper entries.

Click the **Attach** button to attach any files to the message.

Click the **Send** button to send the message to Email2DB.

The message will be processed immediately, allowing you to monitor the logs to check Trigger execution. The logging level will be set to Detailed for messages sent this way - regardless of the logging option set in the program options.

## 8.5 Client Side Database Updates

This topic describes the use of Database Update At Client options of Email2DB. This option is used in contrast to database updates that are from the Email2DB server to the target, for instance a database that is only accessible to the Remote Administrator's machine.

The Update At Client functionality grants the Administrator the ability to control the location of the procedure that will fire an update. This is ideal if you wish to use a database that is only available to your local machine, or you wish to only have the Email2DB product update your system when you have the Remote Administrator open. The connectivity is made by the Remote Administrator that creates that particular Action within the Trigger of Email2DB.

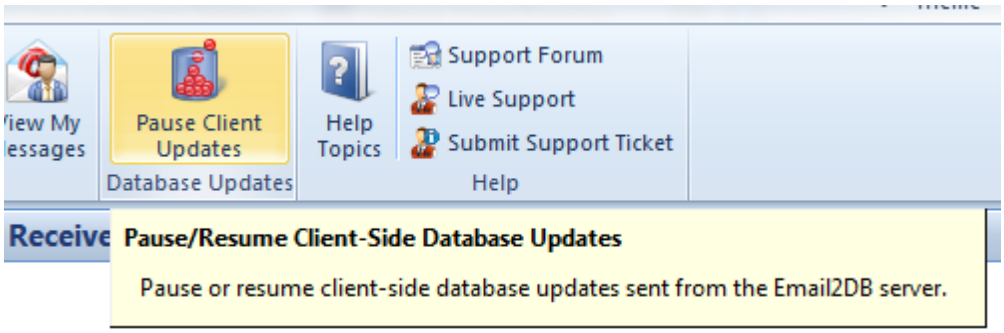
This then opens the ability to use any Connection Strings specific to your desktop or existing Data Source configurations, adding to this database drivers that are only available from your Remote Administrator machine.

The process does add a small amount of overhead in comparison to a typical Database Update Action from the Email2DB Server. This is due to the fact that any information you wish to send to the target database is passed through to your Remote Administrator in an XML packet via HTTPS following message parsing and then to the target database from your machine.

Data is only sent to the Remote Administrator when the Email2DB Server detects that the configured Computer Name's Remote Administrator has connected. When the specific Remote Administrator that built the connection is not connected, the data is queued until the next login from that particular machine.

Each time that the Remote Administrator performs a Client Update a pop-up message is displayed in the bottom right corner of the connected machine where the update was performed. Along with this, if an error occurs, the error message is displayed within a pop-up and is also available within the Message Processor Server Log for you to review.

### Pause Client Updates




For times when you wish to perform modifications to the Client Updates of your machine or stop processing temporarily due to maintenance, you are able to Pause Client Updates from the ribbon menu of the Remote Administrator at all levels Email2DB (Account, Triggers and Trigger Properties). The Remote Administrator will not request data from the Email2DB Server until the Resume Client Updates has been selected by a user.

### Trigger Actions With Update At Client

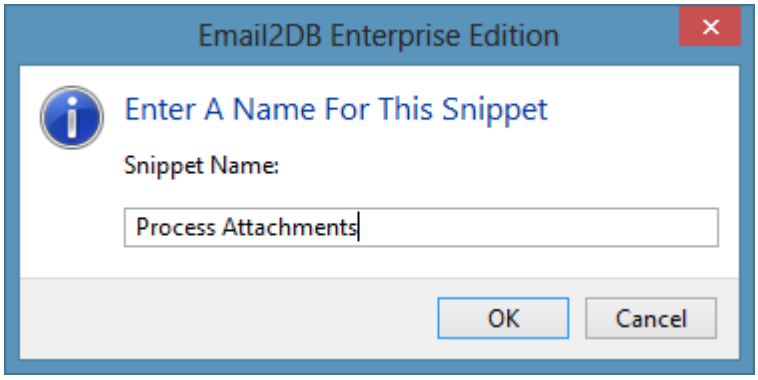
Update A Database
Update A Database Using Custom SQL

## 8.6 Action Snippets

You can save a set of Actions as a 'Snippet'. The Snippet can then be used on any of your other Triggers. Editing the Snippet will change the Actions on all of the Triggers that make use of it.

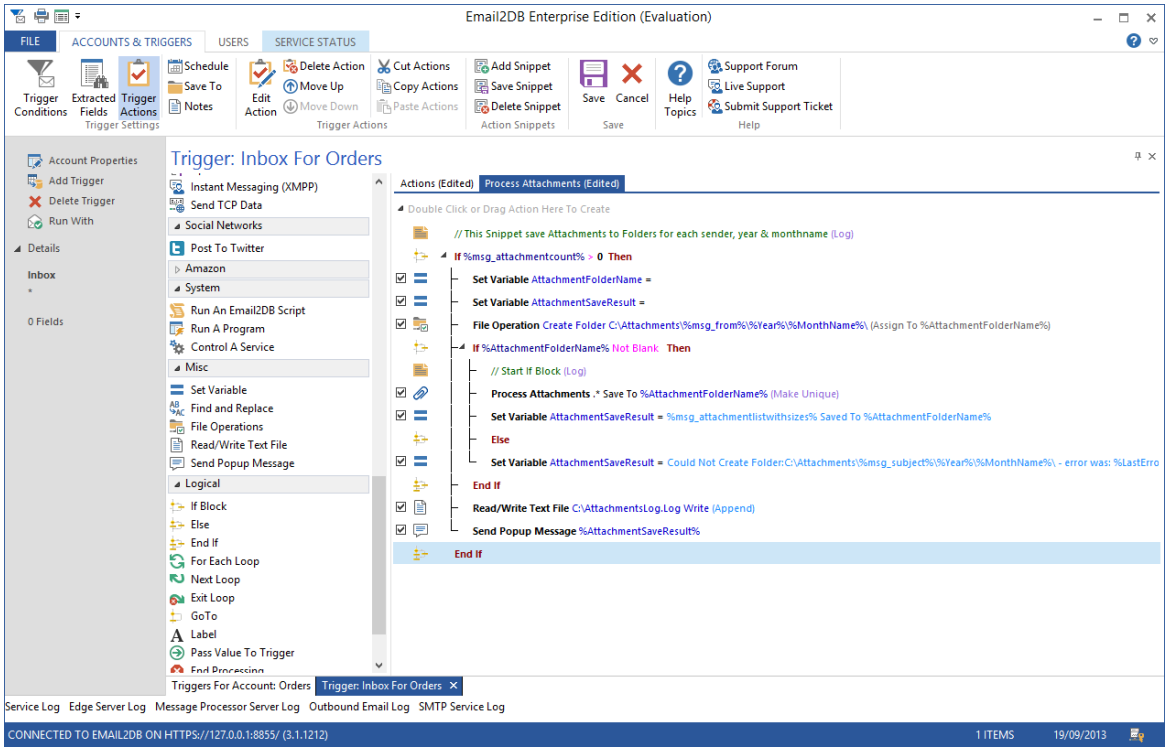
Whilst Editing Trigger Actions click the **Add Snippet** button to create a new Snippet.

You must the give the Snippet a name:

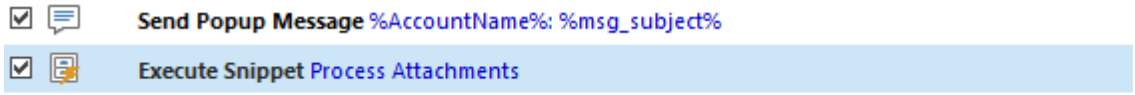



Snippet Names must be unique.

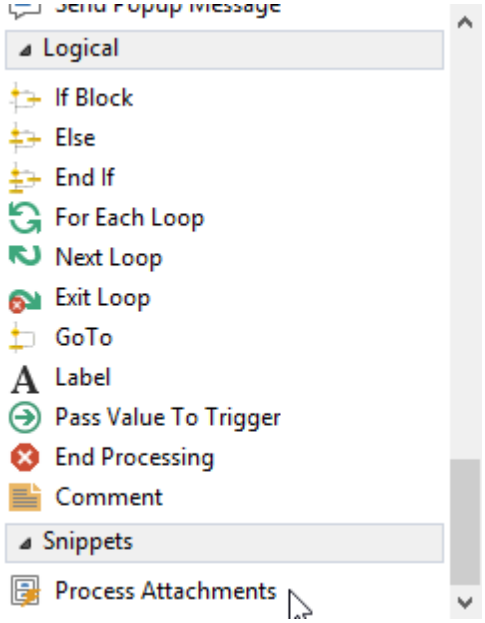
A new tab for the Snippet will be created in the Actions pane. You can then create your Actions as normal:




In your main Actions list an **Execute** action will have been added that points to the new Snippet:




The Snippet will have also been added to the Actions **Toolbox**:




The Snippet will appear in the Toolbox for all your Triggers. You can simply drag it onto the Actions pane to include it. When the Message Processor Executes the Trigger then Snippet Actions will be execute at the point where the Execute Snippet Action is placed.

Whenever you include a Snippet onto one of your Triggers the Snippet tab will be shown in the Actions Pane. You can edit the Snippet here. Any changes you make will affect all Triggers that include the Snippet.

Any variables defined inside the Snippet are visible to the parent Actions and vise versa.

Snippets themselves cannot contain further Snippets.

## 8.7 Reverting To A Previous Revision

If you have enabled the **Keep Copy Of All Trigger Edits** in the main program options then you can revert any Trigger to a previous revision.

Open the Trigger form for any Trigger. Click the **Revert** button on the Ribbon bar. All the revisions for the Trigger will be listed. Select the revision you want to revert to.

## 8.8 Available Action Types

Each of the Actions are optional. You can run one or more of the actions depending on your requirements. Each Trigger can contain any number of Actions. You can add multiple instances of each Action type - so for example, you can send multiple Outgoing emails by adding multiple 'Send Email' actions.

To add an Action double click an action in the **Toolbox** or drag and drop an action to the **Actions List**. When you add an Action the corresponding property page for the Action type will be displayed. To edit an existing action simply double-click it or press Enter.

When a Message is processed by Email2DB the Actions of the matching Trigger will be executed in the order they appear in the Actions list. You can re-order actions using **Move Up**/**Move Down** buttons.

The table below shows the Action Types available with each edition of Email2DB.

| Action Type | Details | SB | Ent/ DC | Hosted |
|---|---|---|---|---|
| **Update Databases** | | | | |
| Update A Database - Direct | Update a SQL database. Add new records or update existing ones. | ✓ | ✓ | ✓ |
| Update A Database - Custom | Update a SQL database using custom SQL statements. | ✓ | ✓ | ✓ |
| Execute A Database Command | Execute a SQL command or store procedure and return results. | | ✓ | ✓ |
| Lookup From A Database | Reads a record from a database and assigns results to multiple Email2DB fields/variables | | ✓ | ✓ |
| Update A CSV File | Update a CSV (comma separated values) file with auto periodic emailing. | ✓ | ✓ | ✓ |
| Update Excel Spreadsheet | Update a Microsoft Excel spreadsheet file. | ✓ | ✓ | ✓ |
| Update Outlook Contact | Update Outlook Contact records directly. | ✓ | ✓ | |
| Update Message Store Token | Updates a counter value in the Message Store that can be later reported on | | ✓ | ✓ |
| Add Message Store Keywords | Adds keywords to the Message Store record to aid searching. | | ✓ | ✓ |
| **Outgoing Messages** | | | | |
| Send Email | Send outgoing emails. With optional future scheduled sending. | ✓ | ✓ | ✓ |
| Send SMS Message | Send outgoing SMS text messages. * | ✓ | ✓ | ✓ |
| Send Appointment | Create appointments on any iCal compatible calendar server. | ✓ | ✓ | ✓ |
| Wait For Validation | Send an email request for someone to validate before remaining actions execute. | | ✓ | ✓ |
| Forward Original Message | Send the incoming message to new recipients. | ✓ | ✓ | ✓ |
| Create An Outlook MSG File | Create an Outlook compatible MSG file. | | ✓ | ✓ |
| **Exchange Server/ Office 365** | | | | |
| Update Contact | Add or update an Exchange Server Contact record. | ✓ | ✓ | ✓ |
| Create Appointment | Add an Exchange Server Appointment record. | | ✓ | ✓ |
| Create Task | Add an Exchange Task. | | ✓ | ✓ |
| Create Note | Add an Exchange Note item. | | ✓ | ✓ |

| | | | | |
|---|---|---|---|---|
| Set/Get Out Of Office Status | Sets or gets the Out Of Office status for a given user. | | ✓ | ✓ |
| Move Message To Folder | Moves the current message to a folder within Exchange/Office 365. | | ✓ | ✓ |
| Add Message To Exchange | Creates a new message on any Exchange/Office 365 server using the current message. | | ✓ | ✓ |
| Mark Message As Junk | Marks the current message as Junk/Not Junk and optionally moves the message to the Junk folder. | | ✓ | ✓ |
| Create Lync Group | Creates a new Lync contacts group. | | ✓ | ✓ |
| Add Contact To Lync Group | Adds or updates a contact in a Lync group. | | | |

**CRM**

| | | | | |
|---|---|---|---|---|
| Get CRM Item | Gets a Microsoft Dynamics or Salesforce CRM Entity value and assigns to a Field or Variable. | | ✓ | ✓ |
| Update A CRM System | Add or update an Salesforce or Microsoft Dynamics entity. | | ✓ | ✓ |
| Upload To CRM | Uploads one or more files or attachments to Microsoft Dynamics or Salesforce CRM. | | ✓ | ✓ |

**SharePoint**

| | | | | |
|---|---|---|---|---|
| Get SharePoint Item | Gets a SharePoint List Item value and assigns it to an Email2DB Field to Variable. | | ✓ | ✓ |
| Update SharePoint List Items | Updates Items in a SharePoint List. | | ✓ | ✓ |
| Upload Files To SharePoint | Uploads one or more files or attachments to a SharePoint site. | | ✓ | ✓ |

**Windows Azure**

| | | | | |
|---|---|---|---|---|
| Update SQL Azure Database | Update an Azure Cloud-Based SQL database. | | ✓ | ✓ |
| Get/Update Azure Table | Read or Update a Windows Azure Table Entity. | | ✓ | ✓ |
| Get/Update Azure Blob | Read or Update a Windows Azure Blob. | | ✓ | ✓ |

**Attachments**

| | | | | |
|---|---|---|---|---|
| Save Attachments | Save attachments of specific types to folders on your file system. | ✓ | ✓ | ✓ |
| Save Attachments To A Database | Save attachments to binary (blob) fields in an external database. | | ✓ | ✓ |

**Online Storage**

| | | | | |
|---|---|---|---|---|
| Upload To DropBox | Save files or attachments to Dropbox. | | ✓ | ✓ |
| Upload To SkyDrive | Save files or attachments to Microsoft SkyDrive. | | ✓ | ✓ |
| Upload To Amazon S3 | Save files or attachments to Amazon Simple Storage Service. | | ✓ | ✓ |
| Upload To Google Drive | Save files or attachments to Google Drive. | | ✓ | ✓ |
| Download From DropBox | Download a file from DropBox to the local file system. | | ✓ | ✓ |
| Download From SkyDrive | Download a file from SkyDrive to the local file system. | | ✓ | ✓ |
| Download From Google Drive | Download a file from Google Drive to the local file system. | | ✓ | ✓ |

**Reporting**

| | | | | |
|---|---|---|---|---|
| Print Message | Print the incoming message and/or a report of extracted fields. | ✓ | ✓ | ✓ |

| | | | | |
|---|---|---|---|---|
| Run A Report | Create a custom report with the report designer. Send PDF reports with outgoing emails. | ✓ | ✓ | ✓ |
| Send A Fax | Send a fax document. | | ✓ | |
| Save As PDF | Renders the incoming message or URL as a PDF file. | | ✓ | ✓ |
| **Web** | | | | |
| Posting To A Web Page | Post extracted field data to any web page, ASP, PHP or CGI script. | ✓ | ✓ | ✓ |
| Read A Web Page | Lookup a web page and add the returned text to an Email2DB field or variable. | ✓ | ✓ | ✓ |
| Download File | Download a file via HTTP from a given URL. | | ✓ | ✓ |
| Call A Web Service | Calls any SOAP or .NET Web Service and returns the results to Email2DB fields or variables. | | ✓ | ✓ |
| DNS Lookup | Lookup a DNS record and add the returned value to an Email2DB field or variable. | | ✓ | ✓ |
| Update WebDav Resource | Upload files, Post Data or Set Properties on WebDav resources. | | ✓ | ✓ |
| Upload To FTP Server | Upload attachments to an FTP Server. | | ✓ | ✓ |
| Instant Messaging (XMPP) | Post a message or get presence with an XMPP compatible instant messaging service. | | ✓ | ✓ |
| Send TCP Data | Open a TCP socket, send data and optionally wait for a response. | | ✓ | ✓ |
| **Social Networks** | | | | |
| Post To Twitter | Post a tweet to Twitter for any authorized user. | ✓ | ✓ | ✓ |
| Post To LinkedIn | Post a share to LinkedIn for any authorized user. | | ✓ | ✓ |
| **Language/ Location** | | | | |
| GeoIP Lookup | Finds the Geo-Location for any IP address or domain name/ email address. | | ✓ | ✓ |
| Translate Text | Translate text from one language to another and return the result to a field or variable. * | | ✓ | ✓ |
| Speak Text | Returns a wave or mp3 stream of the passed-in text being spoken in the desired language. * | | ✓ | ✓ |
| Detect Language | Returns the language code for any text. * | | ✓ | ✓ |
| **System** | | | | |
| Run An Email2DB Script | Execute Visual Basic compatible Email2DB Scripts for custom Trigger actions. | ✓ | ✓ | ** |
| Run A Program | Execute an external application. | ✓ | ✓ | |
| Control A Service | Control a Windows service running on the Email2DB Server computer. | ✓ | ✓ | |
| **Misc** | | | | |
| Set Variable | Create and set an Email2DB variable. | ✓ | ✓ | ✓ |
| Find and Replace | Search and replace text in any Email2DB extracted field or variable. | ✓ | ✓ | ✓ |
| File Operations | Create new folders, copy, move, rename & delete files. | ✓ | ✓ | ✓ |
| Read/Write Text File | Write data to any text file and/or read a text file and assign to a field or variable. | ✓ | ✓ | ✓ |
| Send Popup Message | Send a popup message to Email2DB Administrator or Client users. | ✓ | ✓ | ✓ |
| **Logical** | | | | |
| If Block | Start a logical If block to conditionally control Action flow. | ✓ | ✓ | ✓ |
| Else | Start a logical Else block to execute action if the previous If Block condition fails. | ✓ | ✓ | ✓ |

| | | | | |
|---|---|---|---|---|
| End If | End a logical If block. | ☑ | ☑ | ☑ |
| Go To | Move Action execution to a label. | ☑ | ☑ | ☑ |
| For Each | Start a For Each loop. | ☑ | ☑ | ☑ |
| Next Loop | Move Action execution to the start of the loop unless no more items. | ☑ | ☑ | ☑ |
| Exit Loop | Exit the current loop block (use with a Condition - eg: If X Then Exit Loop). | ☑ | ☑ | ☑ |
| Label | Set a label. | ☑ | ☑ | ☑ |
| Pass Value To Trigger | Pass a given field or variable or custom text to another Trigger for processing. | ☑ | ☑ | ☑ |
| End Processing | Stop Action execution. | ☑ | ☑ | ☑ |
| Comment | Add a comment. | ☑ | ☑ | ☑ |

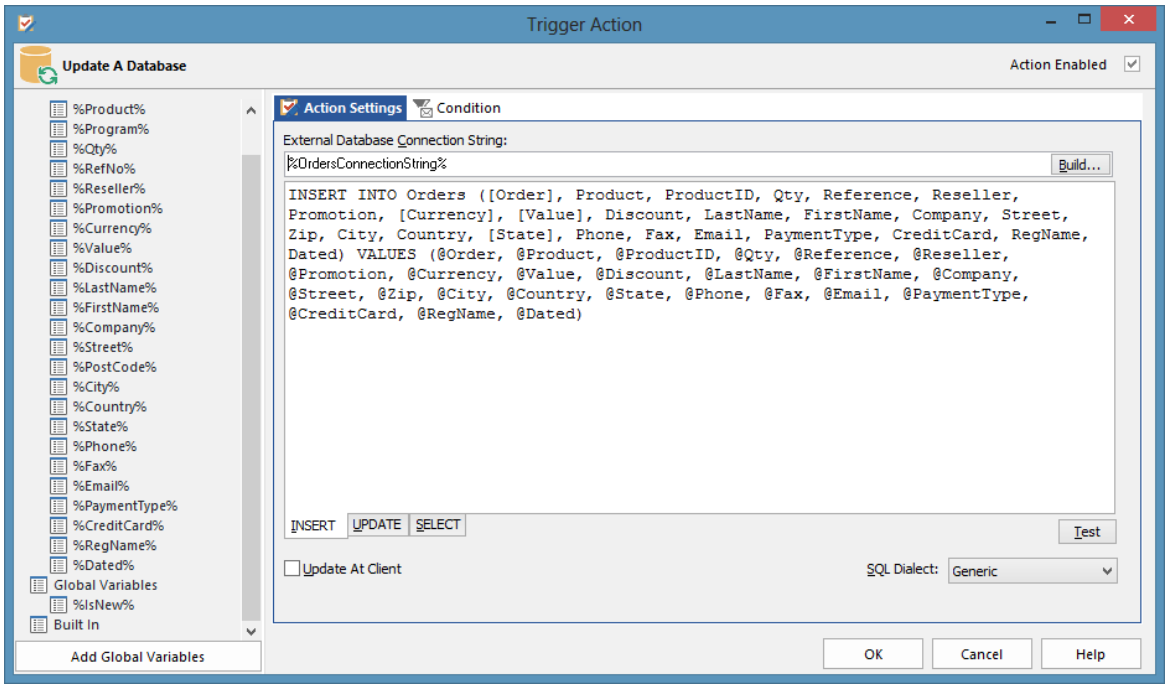
* Requires additional credits.
** Via Professional Services only.

See Also: Using Field Replacements

## 8.8.1 Data

### 8.8.1.1 Update A Database

Updates any data source available on your computer with fields extracted from the incoming message.




The tables and field names used in the SQL statements are specified on the **'Insert Into Database Field'** tab on each individual field. Email2DB will then create the SQL INSERT statement automatically.

You must specify a **Connection String** that Email2DB will use to open the data source. Click the **Build** button to build the connection string using any of the data source providers that are installed on your PC. Click the **Test** button to verify that Email2DB can login to the data source. See Also: Connection Strings

The UPDATE and SELECT statements will only be created if you have defined one or more of your fields as 'Key Fields'. Email2DB will then first check if a record exists with the key field values by issuing a SELECT * From... statement. It will then issue the Update statement if a record is found or the INSERT statement otherwise.

You can only have one Update A Database Action in your Actions List. If you want to update multiple tables within the same database you can specify the different table names against each field in the **Update Table Name** entry - Email2DB will then create separate SQL statements for each separate table being updated.

If you need to update multiple separate databases within the same Trigger then you can use the Update A Database Using Custom SQL Action type. You can have any number of these Actions within your Trigger.

#### Update At Client

Enable this option to allow your Remote Administrator the ability to perform your database update, instead of the Email2DB Server. See: Client Side Database Updates

## Using The Internal Database

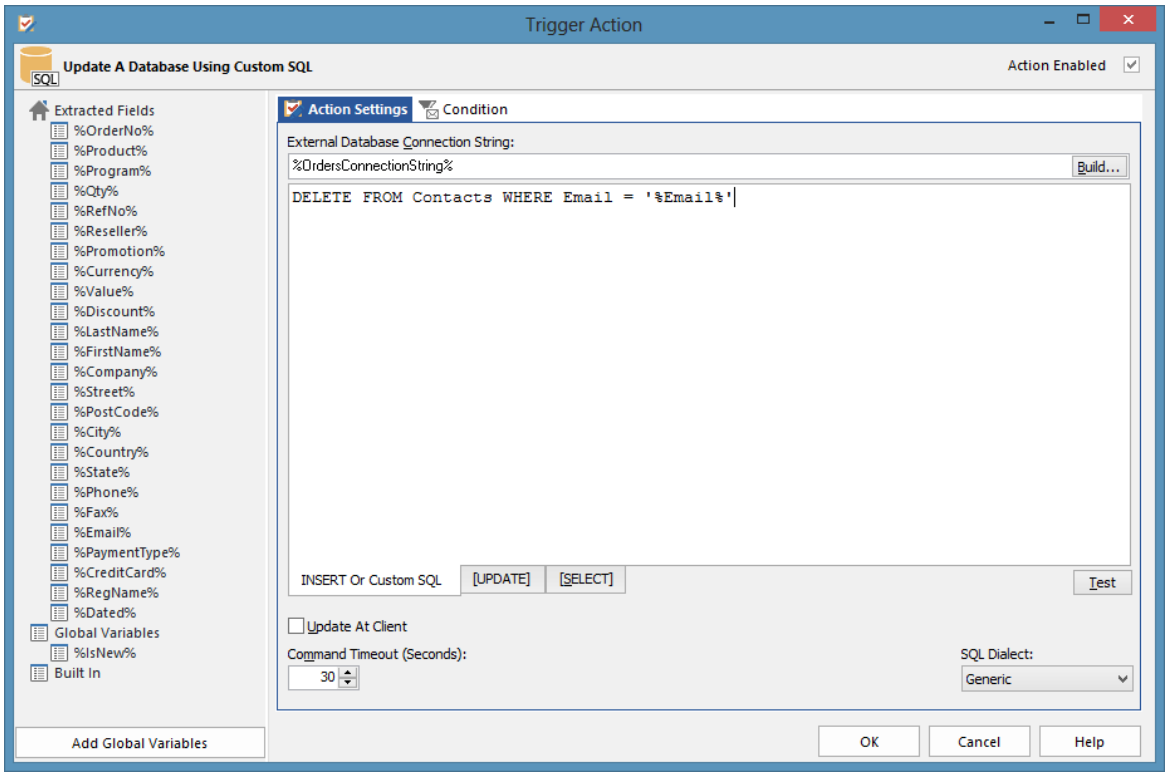
This option will be available if you are using SQL Server for the Email2DB Message Store. See: Configuring The Email2DB Message Store Database.

When this option is selected, Email2DB will use its own database. Tables will be created automatically. Fields will be created for each Email2DB field you have created.

*Note: The Email2DB Hosted Edition can only update the Internal Database.*

### 8.8.1.2 Update A Database Using Custom SQL

Updates any data source available on your computer with custom SQL statements.




This Action allows you to execute any valid SQL Statements against any data source available on your computer. You can include any number of these action types in your Triggers.

You must specify a **Connection String** that Email2DB will use to open the data source. Click the **Build** button to build the connection string using any of the data source providers that are installed on your PC. Click the **Test** button to verify that Email2DB can login to the data source. See Also: Connection Strings

Select the **SQL Dialect** from the drop down box depending on your database type. If the database is not listed select 'Generic'.

The **INSERT** tab is used to enter any valid SQL statement or stored procedure call. You can also optionally enter statements in the **UPDATE** and **SELECT** tabs.

If a SELECT statement is entered in the SELECT tab, then the SQL entered in the UPDATE tab is executed if the SELECT returns one or more records. If no records are returned then the SQL entered in the INSERT tab is executed.

You can also specify a **Command Timeout**. This is the number of seconds that Email2DB will wait for

the SQL statements to execute before canceling the Action with an error. This defaults to 30 seconds. Specify a higher value if you are executing a store procedure that will take longer.

#### @Field or %field% In The SQL Statement

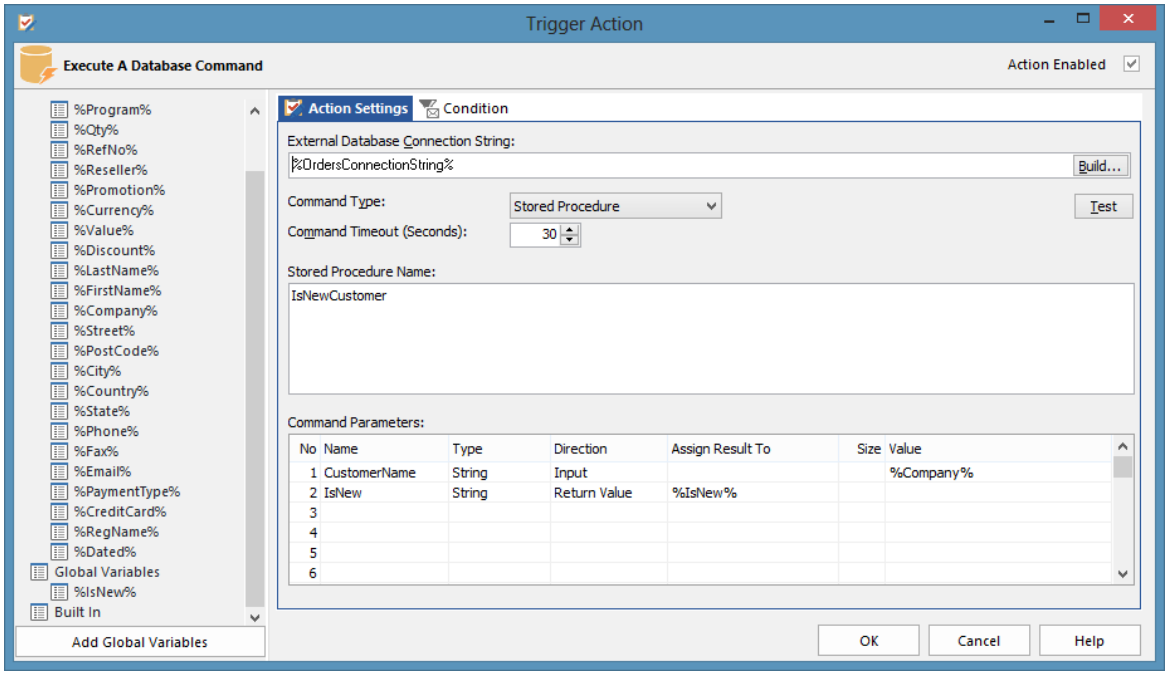
You can use %fieldname% replacements in SQL Statements or your can use the @fieldname format. The difference is that @fieldname tags will be replaced correctly depending on the field type. So strings will include quote marks, dates will be formatted, numeric fields will not contain quotes etc. If %fieldname% tags are used then the field will be replaced as is - you must include quote marks for strings etc.

### Update At Client

Enable this option to allow your Remote Administrator the ability to perform your database update, instead of the Email2DB Server. See: Client Side Database Updates

### 8.8.1.3 Execute A Database Command

Executes a SQL Command or Stored Procedure with optional parameters and returns multiple return values.




This Action allows you to execute a SQL Command or Stored Procedure. You can pass any number of parameters and assign output parameters to variables.

You must specify a **Connection String** that Email2DB will use to open the data source. Click the **Build** button to build the connection string using any of the data source providers that are installed on your PC. Click the **Test** button to verify that Email2DB can login to the data source. See Also: Connection Strings

Select the **Command Type**. This will be a SQL Statement or Stored Procedure.

Specify the **SQL statement** or **Stored Procedure Name** depending on the command type. You can substitute parameter values with ? in SQL statements.

You can pass multiple parameters to your command.

For each parameter you can specify the **Name**, Data **Type**, **Direction** & **Size**. These must match your stored procedure parameters.

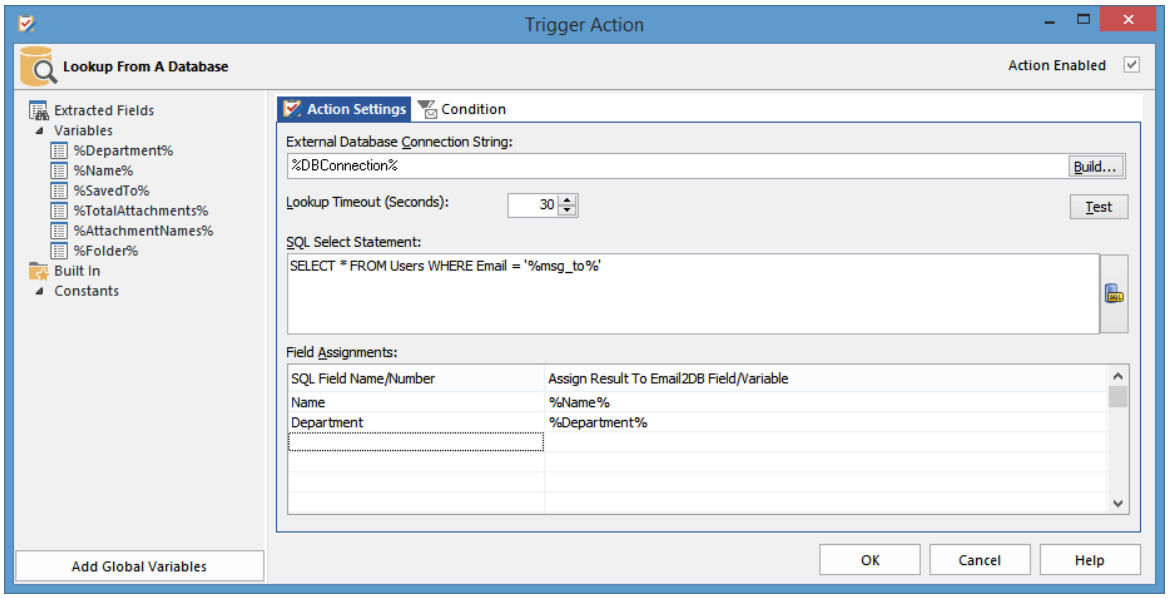
For Output Parameters you can specify the variable to **Assign Result To**.

For Input Parameters you set the **Value** - this can be fixed or a %field%.

Up to 99 parameters can be defined.

### 8.8.1.4 Lookup From A Database

Reads a record from any external database and assigns field results to multiple Email2DB fields or variables.




You must specify a **Connection String** that Email2DB will use to open the data source. Click the **Build** button to build the connection string using any of the data source providers that are installed on your PC. Click the **Test** button to verify that Email2DB can login to the data source. See Also: Connection Strings

Enter the **SQL Select Statement** to use to lookup a record from your database.

In the **Field Assignments** grid you can map database fields used on your lookup table to Email2DB fields or variables.

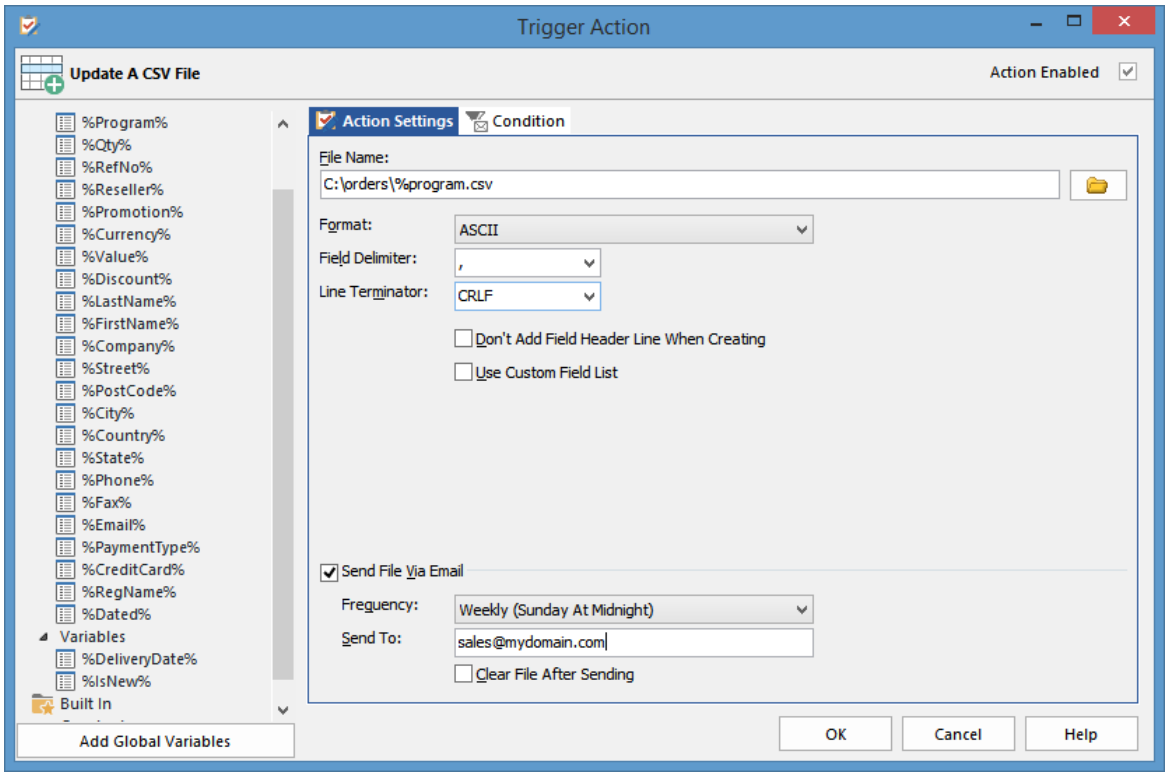
In the **SQL Field Name/Number** column specify a database field name or position number from the SELECT statement.

In the **Assign Result To** column select an Email2DB field or Variable that you want the database field value assigned to.

When Email2DB performs the lookup it will open a readonly recordset using the supplied SQL statement. It will then get the first record from this recordset.

### 8.8.1.5 Update A CSV File

Updates A CSV file with fields extracted from the incoming message.




Enter or select the CSV **File Name** to update.

Email2DB will create the file if it does not already exist. Field names will be written to the first line of the file.

Select the file **Format**. This can be ASCII, Unicode or UTF8. Select Unicode or UTF8 if you expect any fields to contain non-ASCII characters.

Select the **Field Delimiter**. This is normally a comma, but can be any character. You can select select a | (pipe) character or TAB from the drop down.

Select the **Line Terminator**. This can be CRLF (Carriage Return/Line Feed), LF (Line Feed Only), CR (Carriage Return Only), TAB, or Escape characters. CRLF is the default.

Select **Don't Add Field Header Line When Creating** if you want the CSV file to contain data lines only. By default, when the CSV file is first created, the first line will contain the field headers.

Select **Use Custom Field List** if you want to define your own fields to include in the CSV file. If not selected then just the extracted fields will be used. If you choose to use a custom field list then you can then select the field name/variable or constant to include in each column. You can also specify the field header names.

Email2DB can send you the CSV file via email at predefined intervals. To enable this, enable the **Send CSV File Via Email** option and select the **Frequency**. Email2DB can send the CSV file each time a record is added, or on a daily, weekly or monthly basis. If you select daily,weekly or monthly, then Email2DB will send you the file regardless of if new records have been added or not.

In the **Send To** entry specify the email address you want the file sent to. Separate multiple

addresses with semi-colons.

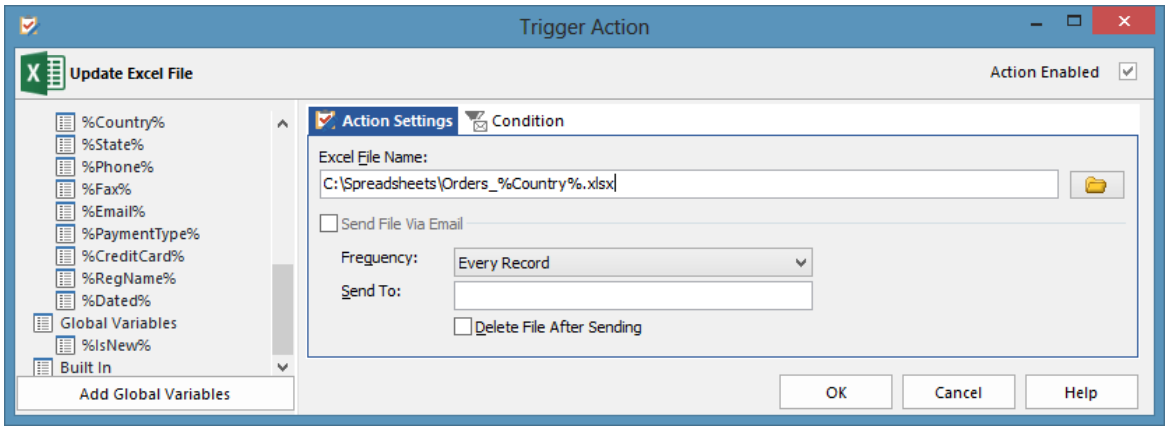
Enable the **Clear File After Sending** option if you want Email2DB to delete the CSV file after it has sent it to you. A new file will be created when the next record is added.

*Note: If the CSV file is located on a network drive the Email2DB Service may not have permission to write to it. You must run the service under a different user.*

*Note: Do not save the CSV file to your desktop. The Email2DB Services runs under the SYSTEM account, which won't have access to your desktop. Instead, save the CSV file to a regular folder on your C:\ drive.*

#### 8.8.1.6 Update An Excel Spreadsheet

Updates a Microsoft Excel 2010 Spreadsheet file with fields extracted from the incoming message.




Enter or select the Excel **File Name** to update.

Email2DB will create the file if it does not already exist. Field names will be written to the first line of the file. Excel does not need to be installed on the Email2DB computer for this Action to work.

Email2DB can send you the Excel file via email at predefined intervals. To enable this, enable the **Send File Via Email** option and select the **Frequency**. Email2DB can send the Excel file each time a record is added, or on a daily, weekly or monthly basis. If you select daily,weekly or monthly, then Email2DB will send you the file regardless of if new records have been added or not.

In the **Send To** entry specify the email address you want the file sent to. Separate multiple addresses with semi-colons.

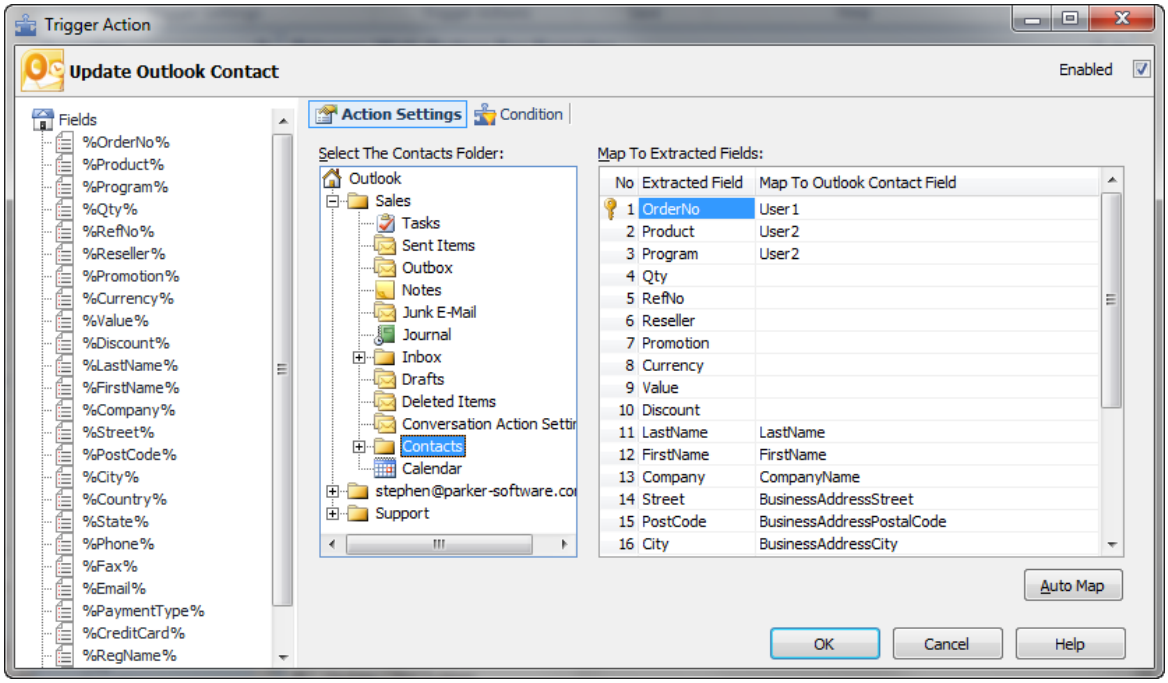
Enable the **Delete File After Sending** option if you want Email2DB to delete the Excel file after it has sent it to you. A new file will be created when the next record is added.

*Note: If the file is located on a network drive the Email2DB Service may not have permission to write to it. You must run the service under a different user.*

*Note: Do not save the file to your desktop. The Email2DB Services runs under the SYSTEM account, which won't have access to your desktop. Instead, save the file to a regular folder on your C:\ drive.*

#### 8.8.1.7 Update Outlook Contact

Updates an Outlook Contact record. This Action type can only be used on the Email2DB Server computer with Outlook 2003/2007/2010 installed (32 bit).




Email2DB will list all your Outlook folders. Select the **Contacts Folder** that contains the Contacts that you want to update.

You must then map your extracted fields to the Outlook contacts fields. Against each extracted field select the Outlook Contact field from the drop down list that matches it (if any). Click the **Auto Map** button to automatically map fields with the same or similar names.

As with the Update A Database, Email2DB will use the Key Fields to look for existing contact records. If an existing contact exists it will be updated, otherwise a new contact will be created.

This option uses the Email2DB Client to update the Outlook contact records. Outlook must be running on the same PC as Email2DB and the connector add-in active for this to work.

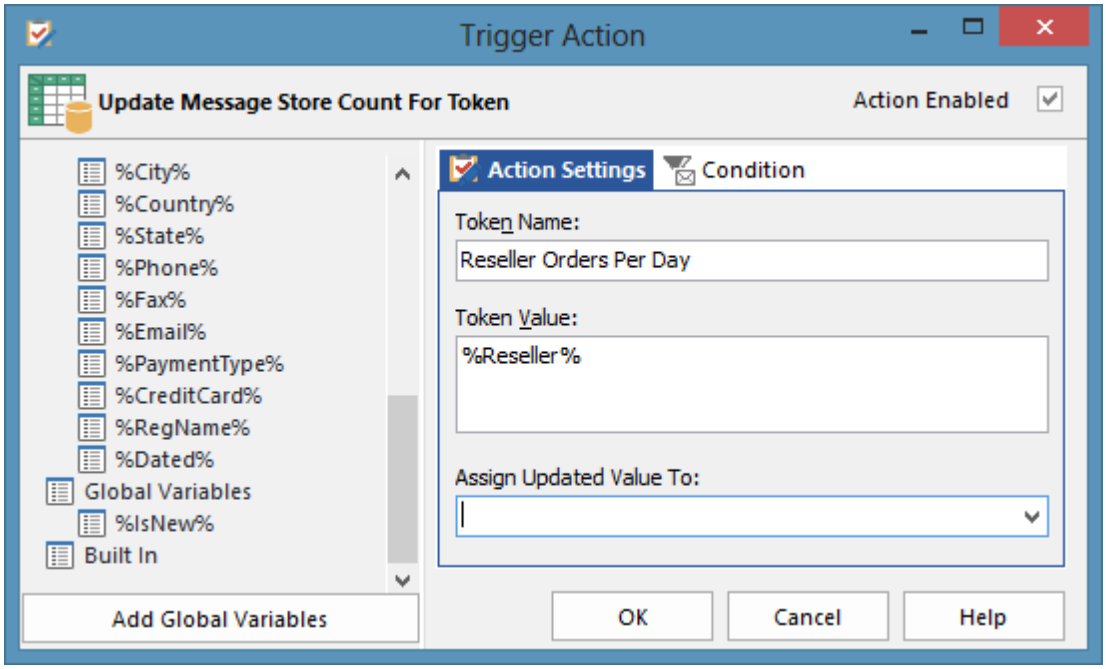
See Also: The Email2DB Client Application
See Also: Integrating With Microsoft Outlook

*Note: Currently the Email2DB Client can only update 32 bit Outlook. A 64 bit version is in development.*

You can also add contacts directly to your Exchange Server without requiring Outlook. See: Update Exchange Contact.

#### 8.8.1.8 Update Message Store Token

Updates a counter value in the Message Store that can be later reported on.




This Action can be used to update a daily counter value that is stored in the Email2DB Message Store for each Account. These counters can be reported on via the Email2DB Dashboard.

Enter a **Token Name** and **Token Value**.

When a message is processed, Email2DB will lookup the Token Name/Value pair in the Message Store against the Message Date. It will create the Token Name/Value pair if a record does not exist, otherwise it will increment the counter for the existing Token Name/Value pair.

You can assign the updated counter value to an Email2DB Field or Variable for use on subsequent actions. Select from the **Assign Updated Value** To list.

Any number of unique Token Names can be counted against each Email2DB Account.

Email2DB counts the following Tokens by default:

- Sender Domain
- Sender Country
- Attachment File Type

So you do not need to create tokens for these.

Any arbitrary data item can be counted. For example:

- Messages by day of week (set the Token Value to %DayOfWeek%).
- Messages containing certain keywords.
- Messages by any of your Extracted Field values (payment types, currencies etc).

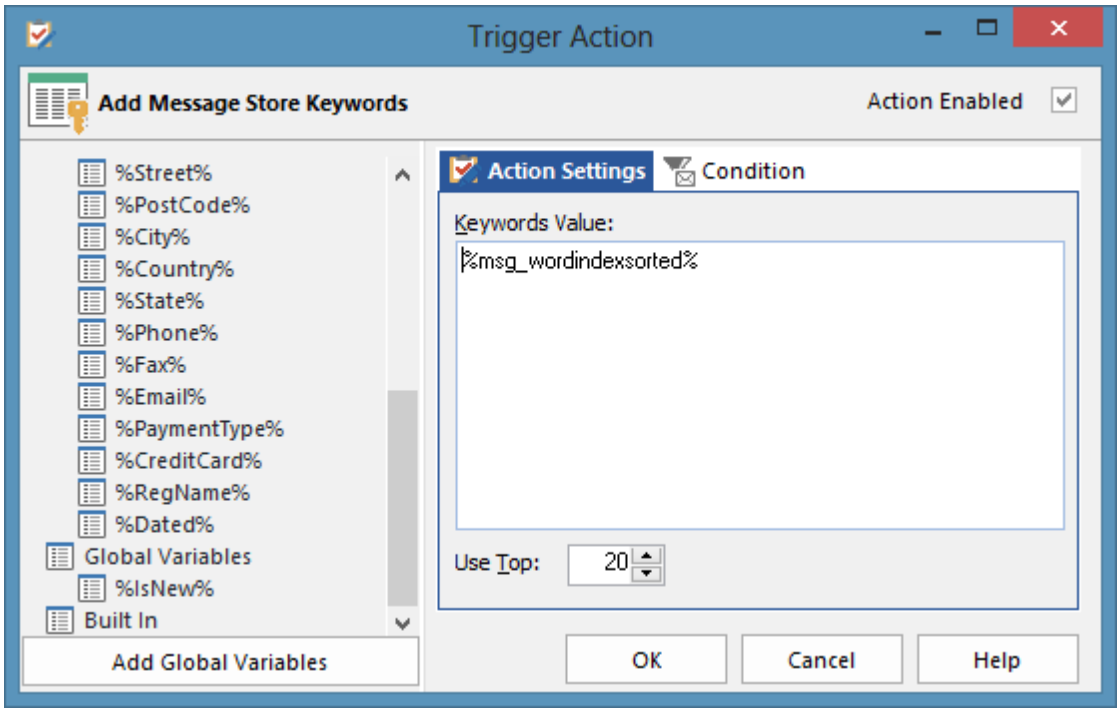
The counted Token Value has a maximum size of 200 characters. Values larger than this will be truncated before being counted.

The Email2DB Statistics viewer will then show a list of Token Names to view. You can then view a chart for any given date range.

*Note: Changing the name of an existing Token will result in a new counter being started.*

### 8.8.1.9 Add Message Store Keywords

Adds keywords to the Message Store record for the current message.




The MessageStore table in the Email2DB Message Store contains a 'Keywords' field. When you view the message store you can search the Subject & From/To addresses when searching for messages. If you add this Action then the Keywords field will be populated and included in any searches.

The **Keywords Value** entry can be set to any text. It defaults to the built-in field **%msg_wordindexsorted%**. This built-in field returns a sorted list of unique words found in the message. This list is sorted by word count (the number of times each word appears in the message). Each word is separated by a space. Common English words, numbers, symbols & tags are excluded.

You can add your own words or extracted field values to the Keywords Value entry. For example: If you have an extracted field called *OrderNumber* and you want the keywords just to contain the Order Number value then set the **Keywords Value** to %OrderNumber%. Then when searching you could enter an order number in the search box to instantly find the relevant message(s). You can include the Order Number AND top keywords by setting the Keywords Value to '%OrderNumber% %msg_wordindexsorted%'.

If the **Use Top** value is specified then the Top x words are used. If this entry is zero then all words are added.

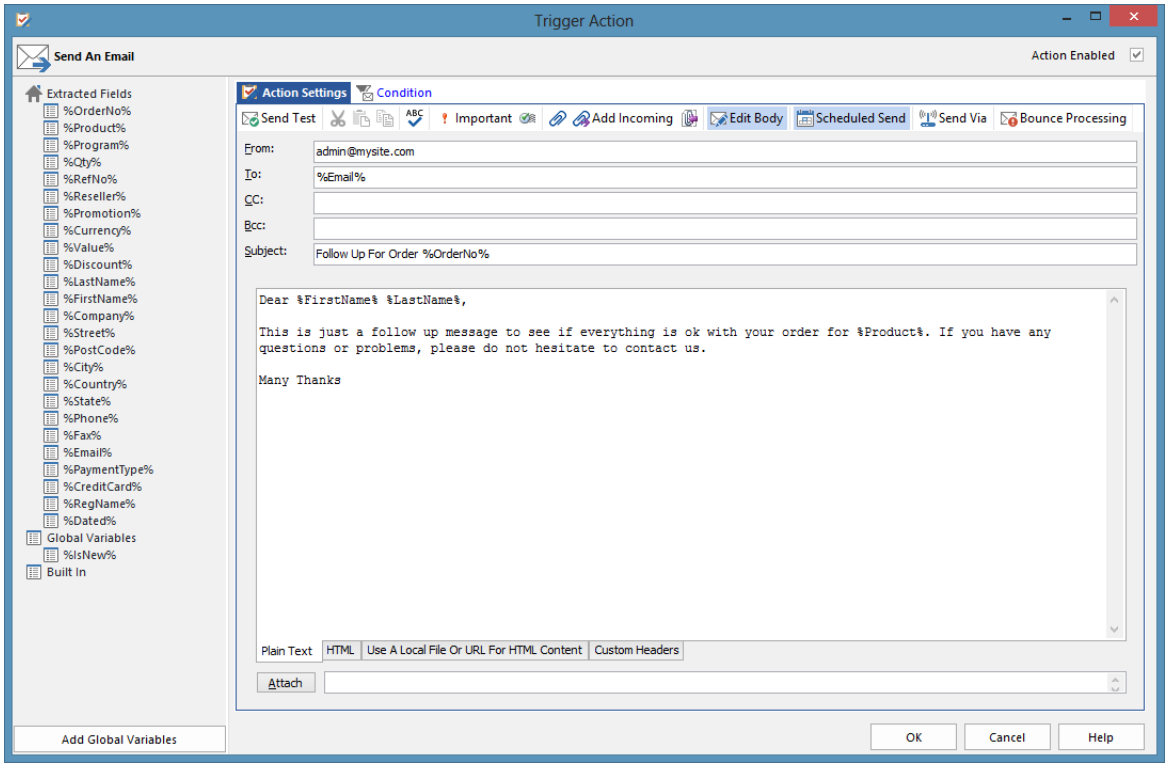
*Note: Using this option will increase the storage requirements for each message in the Message Store - especially if no 'Top' value is specified.*

## 8.8.2 Outgoing Messages

### 8.8.2.1 Send An Email

Sends new outgoing email messages. This can be used, for example, to send a confirmation email to the customer that has just placed an order and to send an email to the order processing department to process it.

You can send multiple outgoing emails to different recipients. Emails can be sent immediately or you can 'Schedule Send' on future dates. Emails can be sent in plain text and/or HTML format. You can add attachments to outgoing emails and you have the option of including the incoming attachments with outgoing emails.




In the **From** box enter the From email address. Leave it blank to use the system default from address.

In the **To** box enter the email address or enter a field name. You can specify multiple addresses by separating each with a semi-colon. Specify optional addresses in the **CC** and **Bcc** entries. If you simply want to reply to the sender of the incoming email, use the built-in field %MSG_From%

*Note: Field names are specified enclosed in % characters.*

Enter a **Subject** - again you can use %fieldname% replacements.

The Fields extracted are listed. Double click a field to add to the currently selected entry.

For the Body of the email click the **Plain Text** tab to enter the plain text portion of the email, click the **HTML** to compose the HTML portion (you can use either or both).

The **Use A Local File Or URL For HTML Content** tab allows you to specify an optional HTML file or URL that will be used when the email is sent. If you do specify a HTML file you should also fill in the plain body text. This will allow the email to still be read on email clients that block HTML content. When sending HTML files you have the option to Embed External Images. If this option is selected,

then any externally referenced images will be downloaded and embedded into the email itself. If you specify a URL for the HTML file, then Email2DB will download the complete web page from the URL specified.

## Attachments

You can attach files to outgoing messages. You can also include all the attachments that were attached to the incoming message.

Click the **Attach** button to add any number of file attachments to the email. The attachments field can use %field% replacements allowing you to use fields or variables containing files saved or created during previous actions.

Click the **Add Incoming** button to automatically send any incoming attachments with the outgoing email. This will be in addition to any other attachments you want to send.

The  button is used to attach incoming inline attachments. Inline attachments are images and other types of file that are included in the body of the incoming email. If this option is enabled then Email2DB will add the inline attachments are regular attachments to the outgoing message.

## Options

Click the **!** button to mark the email as Important.

Click the **Request Read Receipt** button to request a read receipt to be sent to the From address when the recipient has read the email.

Email2DB includes a built-in spell checker. Click the **Spelling** button to check your spelling. All %fieldname% embedded fields will be automatically excluded from the spell check.

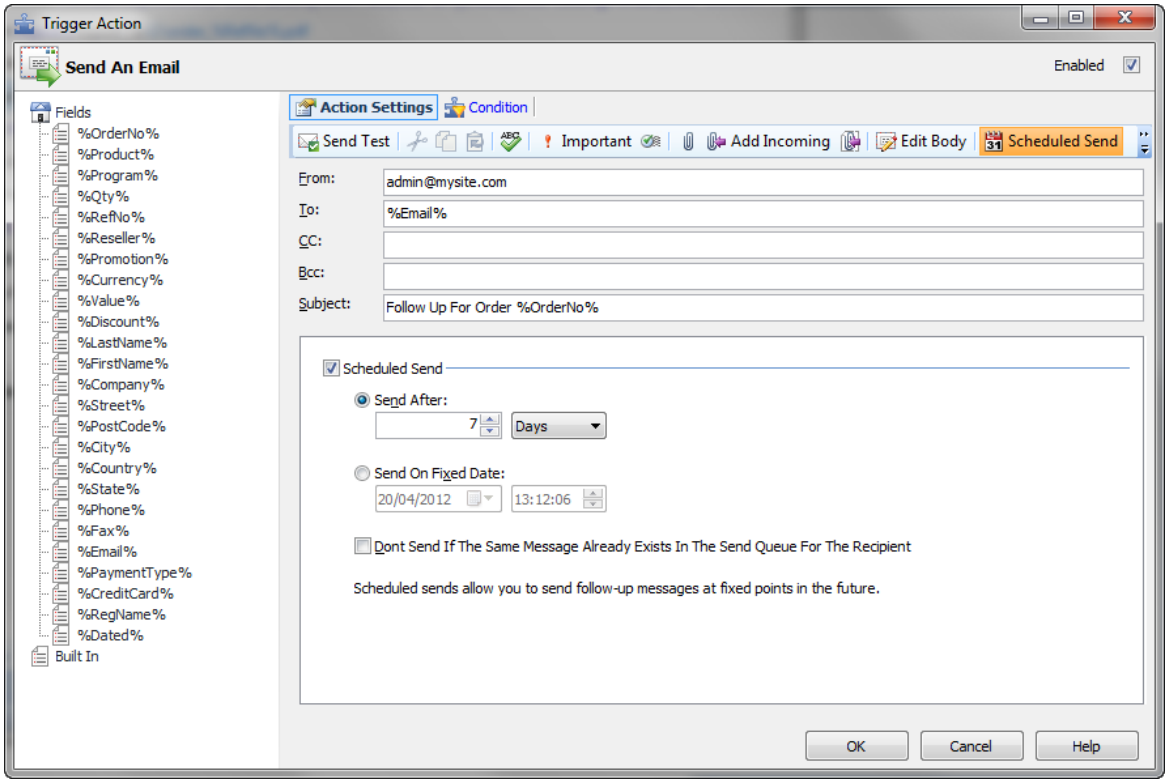
## Scheduling Messages

Email messages can be sent on future dates using **Scheduled Send** option.

For example, suppose you have a Trigger that responds to a sales order email. The Trigger sends the customer a 'thank you' email when the order is received. You could then use the Scheduled Email option to send a follow up email in 30 days time to see how the customer is getting on with their new product.

Any number of scheduled email responses can be setup per trigger. Email2DB records the email in it's database along with the scheduled date and time that the message should be delivered. It checks this database every few minutes and automatically sends out pending messages.

On the Edit Message Body form, click the **Scheduled Send** button:




Click the **Scheduled Send** check box to enable the scheduled send option.

Click **Send After** to enter the number of days after the message is processed that you want the outgoing email to be sent.

Or, click the **Send On Fixed Date** to enter a specific date and time in the future that you want the outgoing email to be sent.

Click **Edit Body** to save the scheduled send options and return to editing the email.

*Note: You can view the list of pending outgoing emails by right-clicking the Account and selecting View Scheduled Messages. From here you can delete a pending message, or clear the complete list.*

*Note: You can also add scheduled emails in scripts using the SendScheduledEmail command.*

See Also: Viewing Scheduled Emails

## Send Via

By default, Email2DB will send outgoing emails directly to the recipients mail server (unless a 'Smart Host' is specified in the Email2DB Mail Server Options). Email2DB also allows you to specify a specific mail server to use on an individual message basis. This is useful when you want to route specific messages via a specific mail server. For example, suppose you have two mail servers, one located in the USA and one in the UK. You could create 2 versions of the outgoing message, one that is sent via your UK mail server for UK recipients and one that is sent via the USA server for everyone else. You could then use the Conditional Execution option to define which email is sent based on a Country field. This will help your messages being blocked as spam - since it's always better to send email via a mail server that is closest to the recipient.

Click the **Send Via** button on the toolbar.

Now specify the **SMTP Server** details.

In each of the SMTP Server entries you can make use of %fieldname% replacements. This allows you to create a mail server field which you could set via Field Extraction options to define the mail server to use.

### Bounce Processing

You can define how bounces are handled for this outgoing message. Click the **Bounce Processing** button on the Toolbar.

You can then choose between 3 options:

1. Send Non Delivery Report To Sender. The NDR will be sent to the email address specified in the From field.
2. Send Non Delivery Report To: - you can then specify a specific address.
3. Don't Send Non Delivery Report.

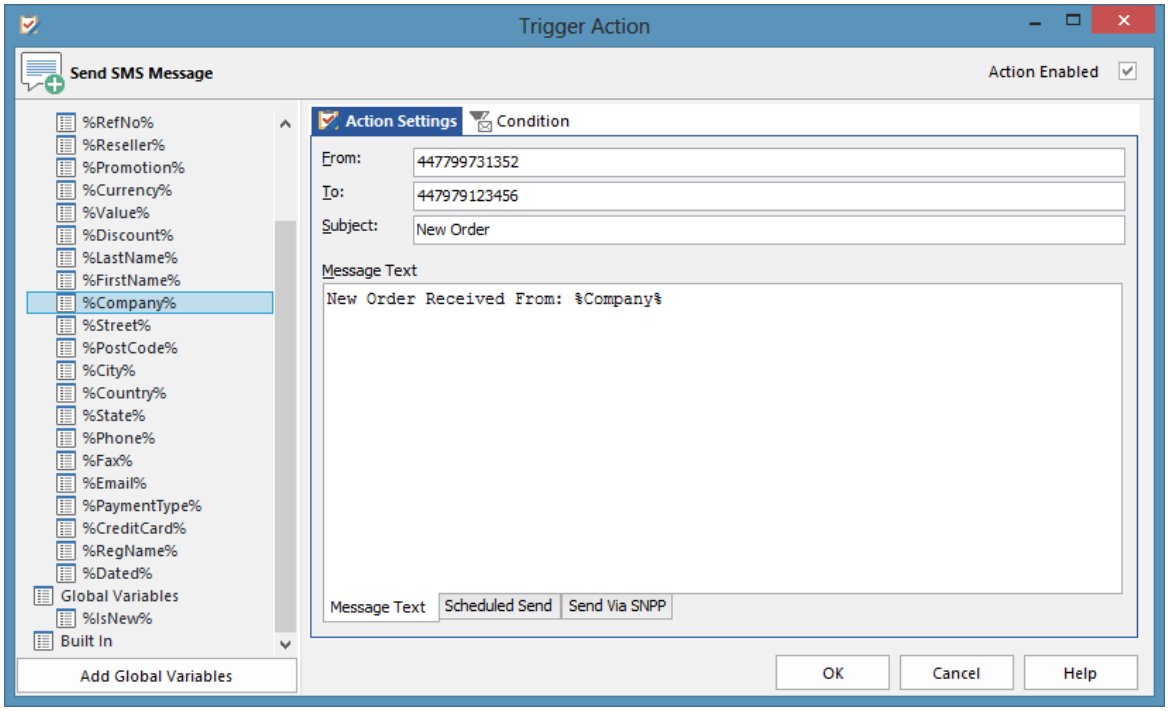
You can also pass the Non Delivery Report email to another Trigger for further processing (for example, to remove the email address from a subscribers database). Select the **Then Pass Non Delivery Report To Trigger** option and select the Trigger from the drop down menu.

#### 8.8.2.2 Send An SMS Message

Sends outgoing SMS text messages to different recipients. Messages can be sent immediately or you can 'Schedule Send' on future dates. SMS messages can be sent to any mobile device in most counties and using most networks.

The actual sending of SMS messages is done by our SMS Server Gateway. Before sending SMS messages you need to setup an SMS Server Account and purchase 'credits'.

SMS messages cost between $.12 and $.05 each depending on the country and network of the receiver. The receiver of the SMS is not charged anything. Email2DB can send messages to almost all countries and networks. For full details of coverage and pricing see http://www.email2db.com/ordersms.aspx




**From** No# (optional).
Enter the phone number that the message will appear to be from. The recipient of the message can then reply to this number.

**To** No#.
Enter the recipient's phone number. You can specify multiple recipients by separating the numbers with commas.

Both numbers can use %fieldname% replacements.

*Note: Phone numbers must be entered in international format. Country code first - no space - then full number with no leading zero.*

Enter the **Message Text**. This can be up to 160 characters in length.

## Scheduling SMS Messages

SMS messages can be sent on future dates using **Scheduled Send** option.

On the Message Text form, click the **Scheduled Send** tab.

Click the **Scheduled Send** check box to enable the scheduled send option.

Click **Send After** to enter the number of days after the message is processed that you want the message to be sent.

Or, click the **Send On Fixed Date** to enter a specific date and time in the future that you want the message to be sent.

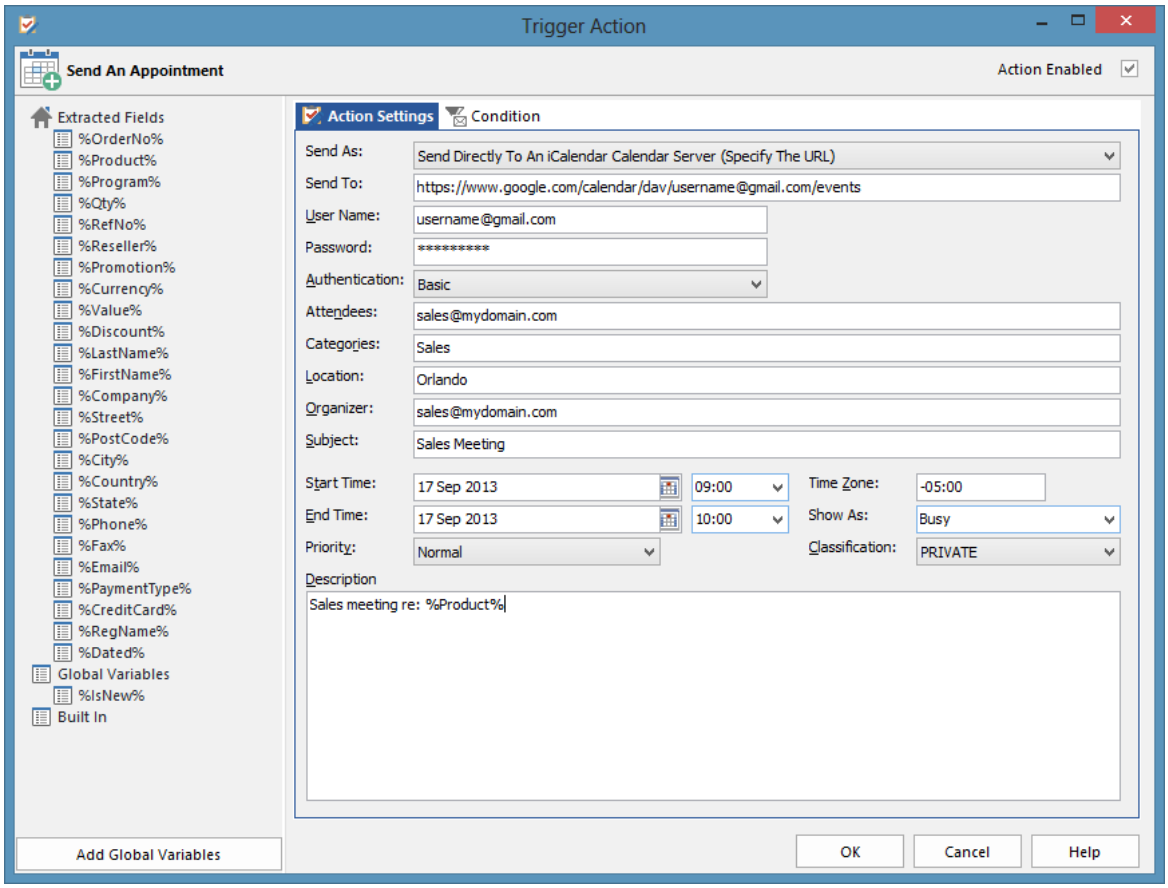
## SMS Server Settings

Before SMS messages can be sent you must configure the SMS Server settings in the Program Options. See: Options - SMS

### 8.8.2.3 Send An Appointment

Creates an appointment in any iCalendar compatible Calendar Server or sends an iCalendar compatible appointment request as an email attachment.

iCalendar is used and supported by a large number of products, including Google Calendar, Apple iCal, Lotus Notes, Yahoo Calendar, Microsoft Outlook and others.




The Send As drop down has two options:

**1. Send Directly To An iCalendar Server**

This option allows you to post directly to an iCalendar Server using the CalDav protocol. You need to specify the URL of the server to post to. This URL will depend on the Calendar Server you are using.

For Google Calendars use: https://www.google.com/calendar/dav/{googlemailaddress}/events/ - and use your Google user name/password.
For Yahoo Calendars use: https://caldav.calendar.yahoo.com/dav/{user}/Calendar/{calendarname}/ - the {user} would be your Yahoo user name (email address) and {calendarname} is the name of the Calendar to update. For example: https://caldav.calendar.yahoo.com/dav/myname@yahoo.com/My%20Calendar/ (the %20 represents a space).

You may need to specify a **User Name** and **Password** and **Authentication** mode if the iCalendar Server requires a login first. Login to your iCalendar Server to obtain the iCal address.

**2. Send To An Email Recipient As An iCalendar Attachment**

With this option you specify an email address. The appointment will be sent as an attachment with the .ics extension. Most Calendar applications will import this directly when the recipient opens the

attachment.

All remaining text fields can use %field% replacements.

You must specify the **Attendees** as email addresses. Multiple addresses can be separated by commas.

The **Categories**, **Location, Subject** & **Description** fields are optional.

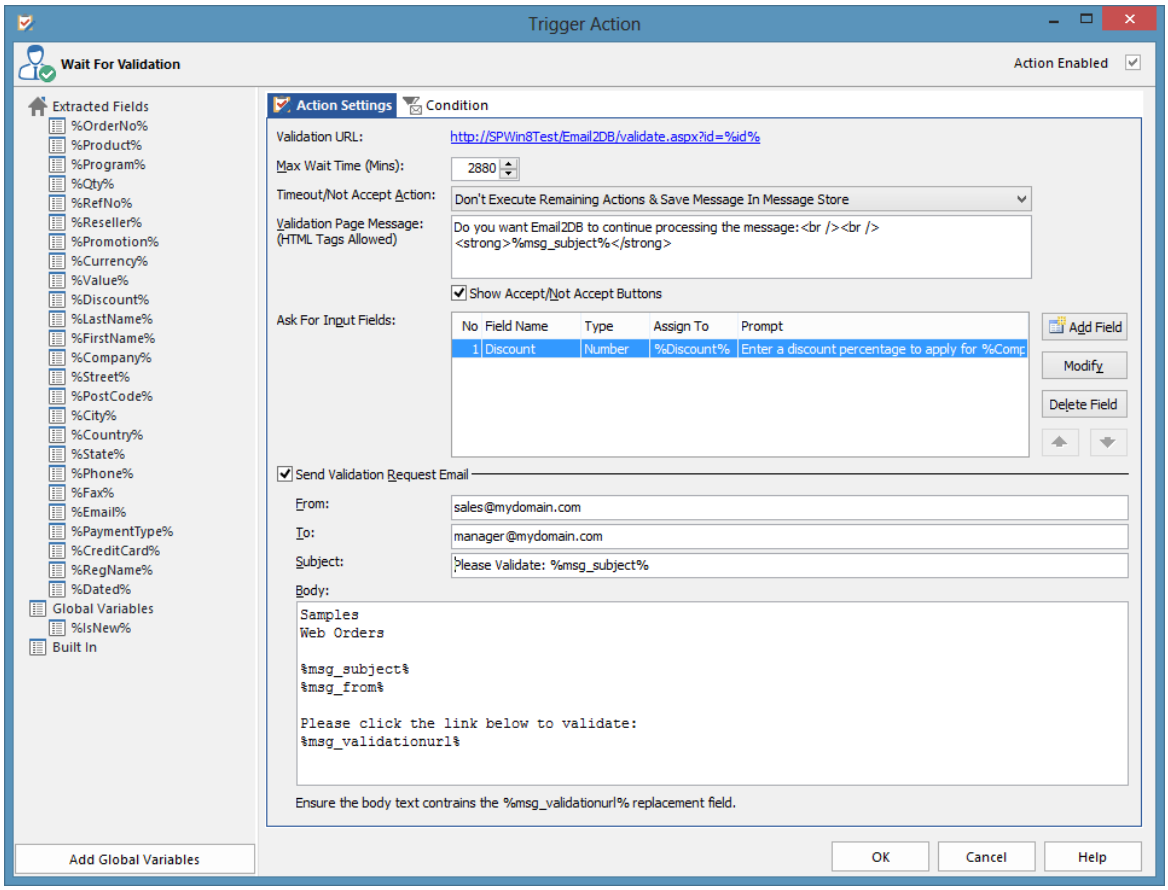
The **Organizer** must be specified as an email address.

The **Start Date/Time** and **End Date/Time** can use %field% replacements if required. If a Start/End Date is given as a %field% replacement then the field contents must be able to be interpreted as a date.

#### 8.8.2.4 Wait For Validation

Waits for a validation URL to be accessed before proceeding with the remaining actions for the Trigger.

The Action is used to provide a human validation response in the form of a unique URL that must be accessed. Email2DB will pause execution of the Actions for a message until the validation is performed. It will then move on to process the next message. A single Trigger can contain multiple Validation actions.




For each Validation request Email2DB creates a unique Validation URL. You must send an email or SMS containing the %msg_validationurl% field replacement to someone or provide the URL in some other manner (write it to a HTML file for example). When the recipient receives this email they click the URL to validate the message. Once validated the remaining actions setup on the Trigger are then

executed for that specific message. Validation URLs are secured using a one-way hash, so a user could not validate the wrong message by manually changing the URL.

You can either use the Send An Email or Send An SMS Message actions to send you own message containing the %msg_validationurl% or enable the **Send Validation Request Email** option to include the email sending within the Validation action itself. You then need to specify the **From**, **To**, **Subject** and **Body**. The Body text must contain %msg_validationurl% somewhere.

**Max Wait Time (Mins)**

This is the maximum number of minutes that Email2DB should wait for the validation. It defaults to 2880 (48 hours).

**Timeout/No Accept Action**

You must specify what Email2DB should do if a validation response is not received before the Max Wait Time or the user clicks the 'Not Accept' button if the **Show Accept/Not Accept Buttons** option is enabled.

The options are:
1. Don't Execute Remaining Actions & Save Message In Message Store.
2. Execute Remaining Actions & Save Message In Message Store.
3. Don't Execute Remaining Actions & Drop Message From Message Store.
4. Pass To Another Trigger & Drop Message From Message Store.

You can use the Pass To option to execute another Trigger in the event that the Validation is rejected or expires.

**Validation Page Message**

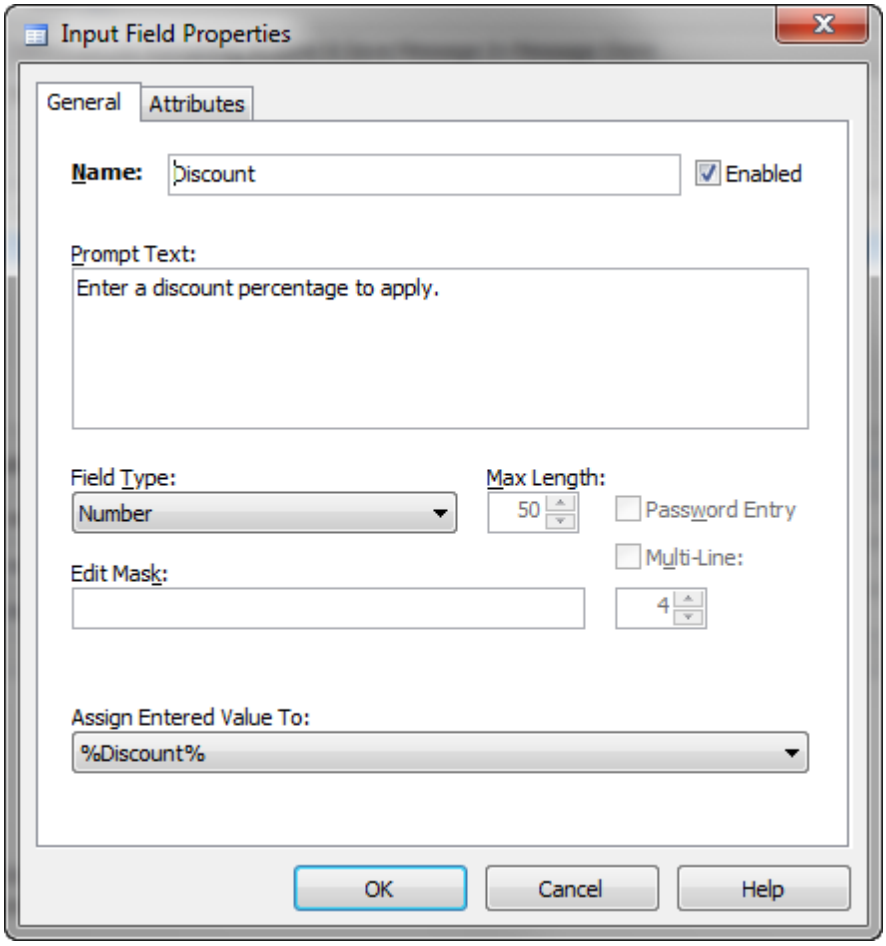
This is the text that will be displayed in the users browser when they click on the Validation URL to validate a message. It can contain %field% replacements and HTML tags.

**Show Accept/Not Accept Buttons**

If this option is enabled then the Validation page will show Accept & Not Accept buttons below the Validation Page Message. The user must click the Accept button to validate the message. If the Not Accept button is clicked then the message is rejected and the Timeout/Not Accepted Action performed. If this option is not enabled then the user only needs to click the Validation URL to validate the message (unless input fields are used - see below). You should adjust the Validation Page Message accordingly.

**Ask For Input Fields**

In addition to displaying a message, Email2DB can optionally show a form containing any number of input fields. If any fields are specified then the user must complete the form before validating the message. The results of the form are passed back to the Trigger and can be used in subsequent actions. Click **Add Field** to add a new input field.




Enter a **Name** and **Prompt Text**. The **Field Type** can be Text, Number, Date, Time or Boolean. For text field types you can specify the **Max Length**. The **Multi-Line** option allows you to define the maximum lines. The **Password Entry** option allows you to mask the input.

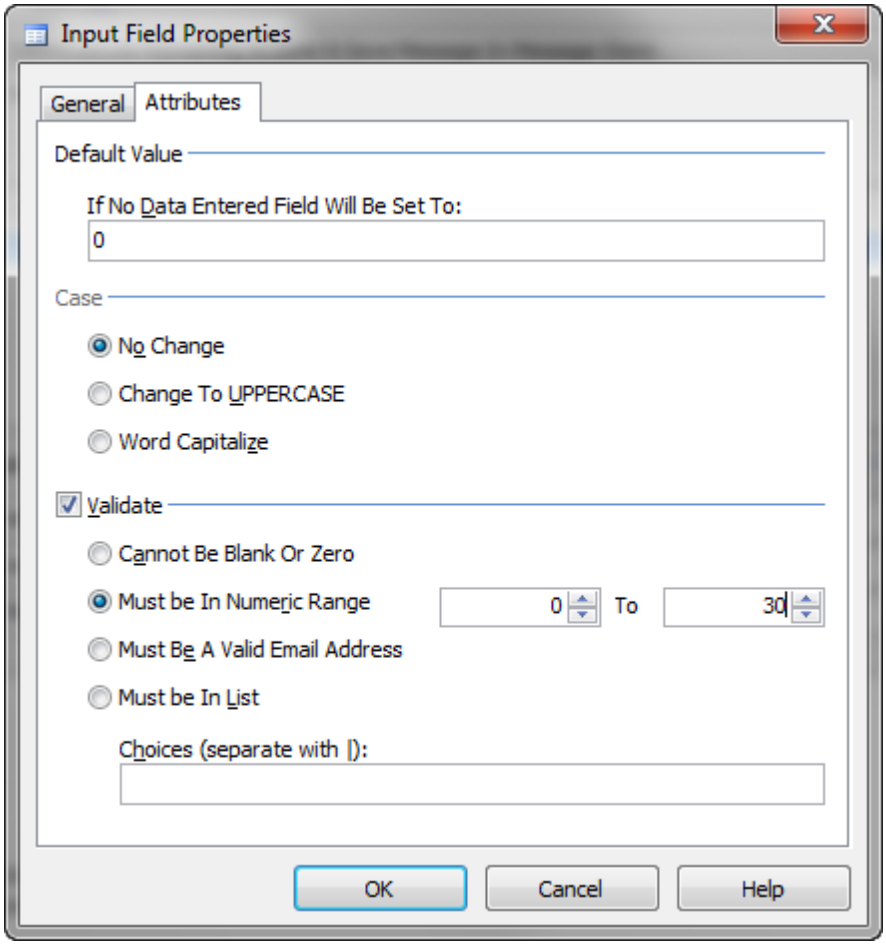
The **Edit Mask** can be used to supply a regular expression to use as an input mask to format and limit what can be entered.

**Assigning To A Field**

You must then select an Email2DB field or variable to assign the entered value to. Select the field/variable from the **Assign Entered Value To** drop down. Once a message is validated and the Trigger continues execution you will be able to make use of input values in subsequent actions.

A simple example would be a single field that asks 'Do you want to process this message' - with a Y/N choice. The value can be assigned to a Variable called 'Continue'. After the Validation Action you can add a conditional End Processing Action: *'If %Continue% Is Not Equal To Y End Processing'*

The **Attributes** tab enables you to set additional attributes for the field:




You can specify a **Default value** that will be assigned to the field if the visitor does not enter anything.

The **Case** option enables you to change the Case of the field value.

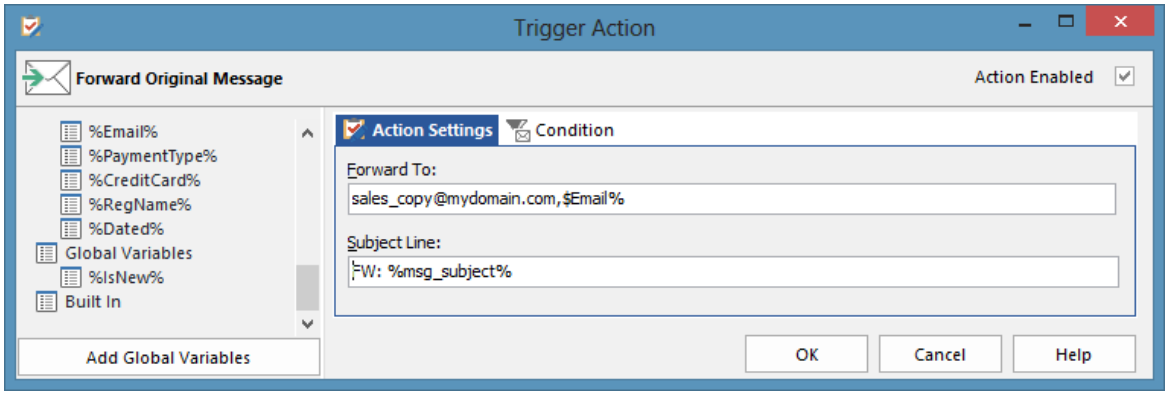
The **Validate** option enables you to apply validation against the field value:
- Cannot Be Blank Or Zero - the visitor must enter something.
- Must Be In The Numeric Range - for number fields the value must be in the specified range.
- Must Be A Valid Email Address - you can specify that the field will only accept a valid email address.
- Must Be In List - you can specify a list of possible values that are allowed. Separate the choices with a | character.

*Notes: The Email2DB Web Services must be configured and the Web Services URL correctly specified in the program options for the validation URL to work correctly.*

See Also: Customizing The Validation Response Page

#### 8.8.2.5 Forward Original Message

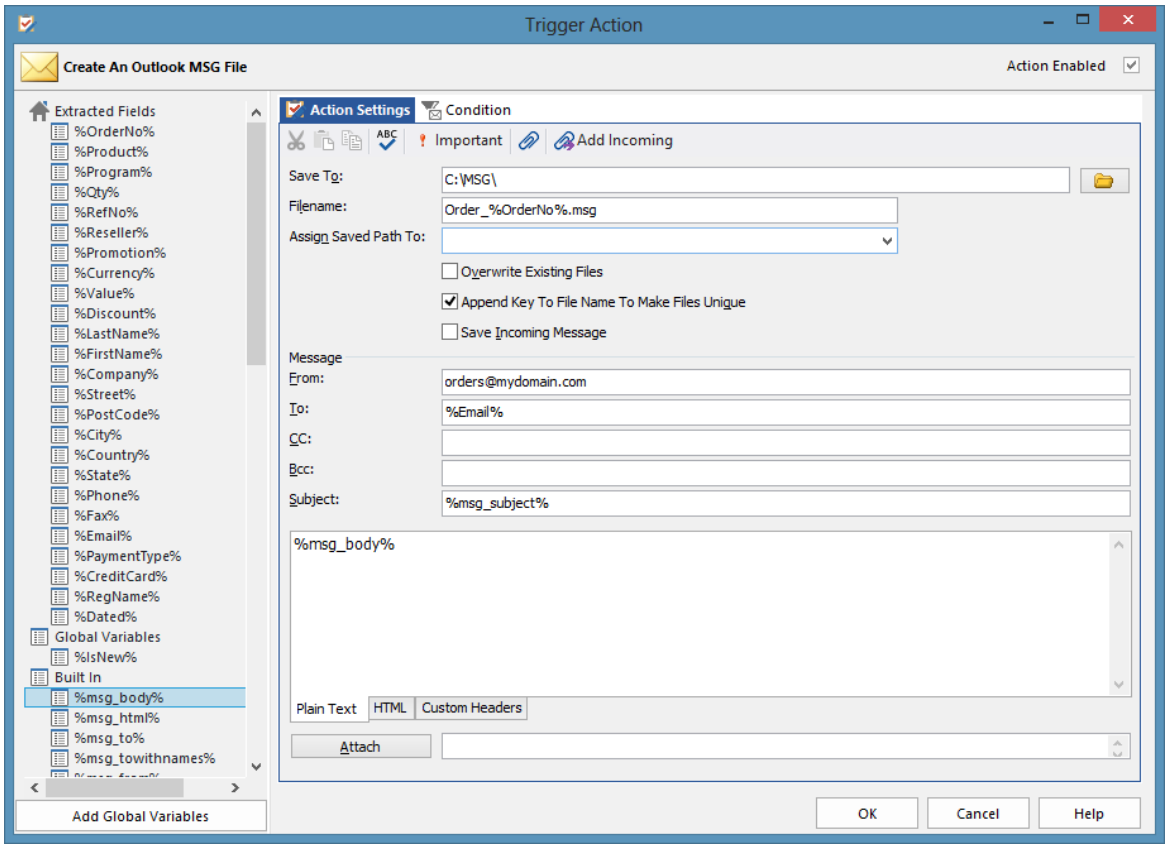
Forwards the incoming message to new recipients.




Enter the **Forward To** address and **Subject** Line.

### 8.8.2.6 Create An Outlook MSG File

Saves the current message or a custom message as a Microsoft Outlook compatible MSG file to a folder on your file system.




Select the **Save To location** and enter a **Filename**.

You can select a field/variable to **Assign Saved Path To**.

To ensure the filename is unique enable the **Append Key To File Name To Make Files Unique** option.

Enable the **Save Incoming Message** if you want the incoming message saved as a MSG file. If not

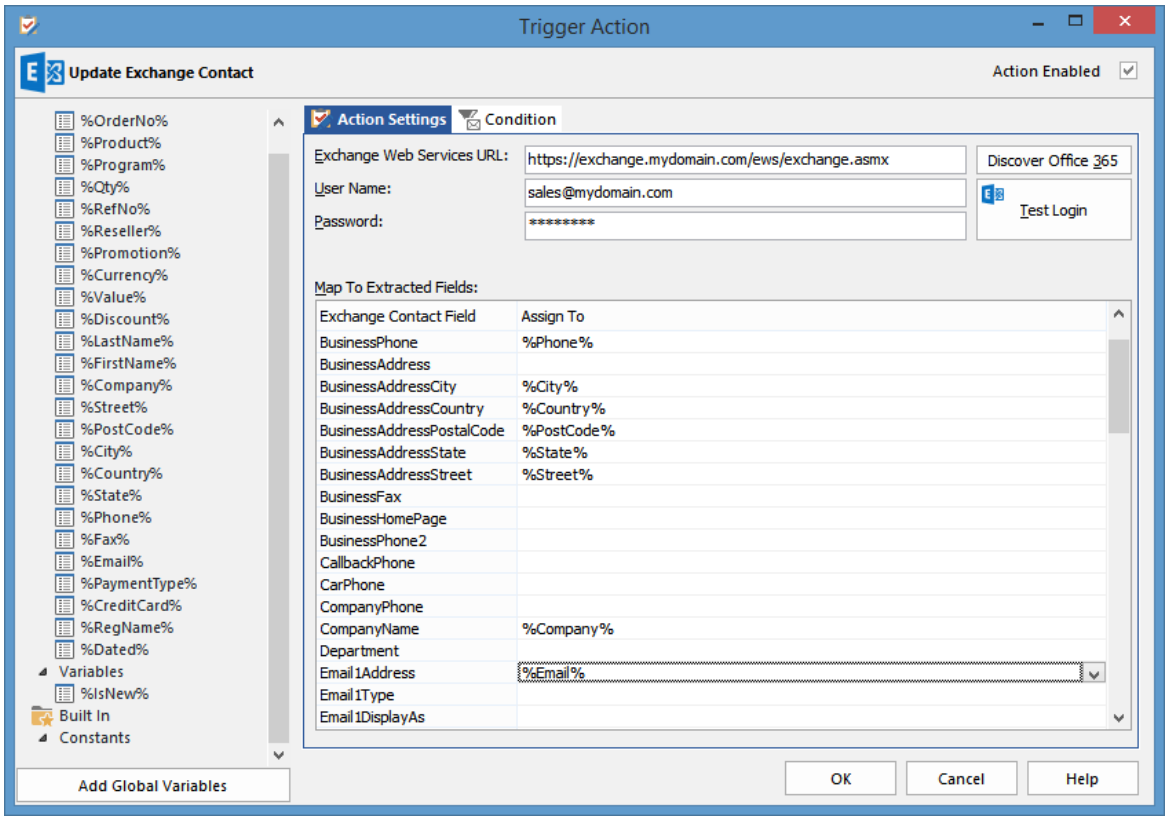
enabled you can create a custom message by completing the **From**, **To**, **CC**, **Bcc**, **Subject** and **Body** fields and adding any **Attachments**.

The message will then be saved as an Outlook MSG file. This can be opened directly by Outlook 2007 and higher.

## 8.8.3 Exchange/Office 365

### 8.8.3.1 Update Exchange Contact

Adds or updates an Exchange Contact record for a given user on a specified Exchange 2007/2010/2013 server. This includes Office 365.




Enter the **Exchange Web Services URL**. The normally takes the form https://{exchangeserver}/ews/exchange.asmx

You will need to specify a **User Name** and **Password**. This will be the user on the Exchange Server who you want to add a contact record to.

Click the **Text Login** button to verify the connection details.

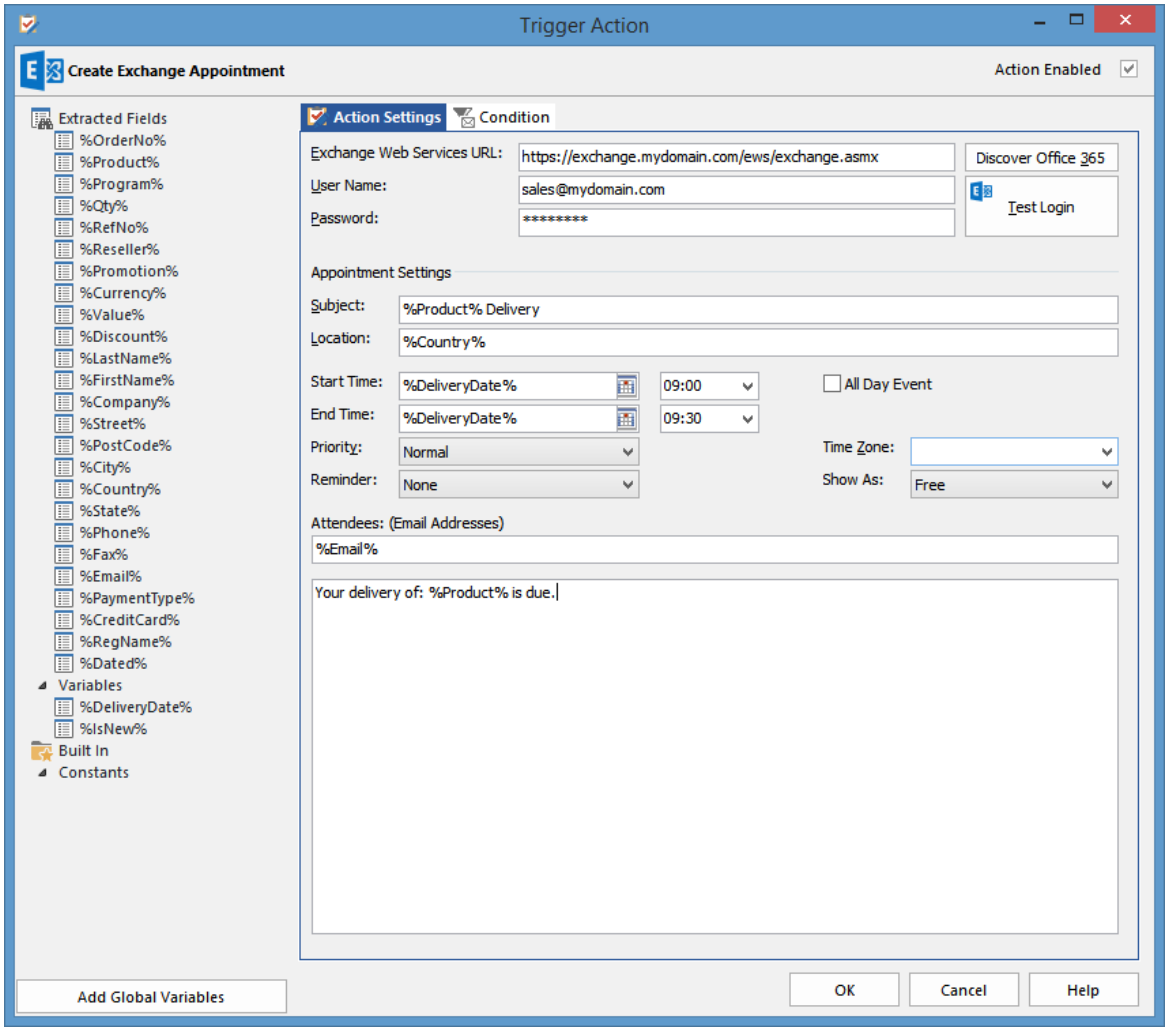
Email2DB will first lookup an existing contact using the Email1Address field (if specified). If an existing contact is found then the contact is updated, otherwise a new contact record is created.

#### Discovering Your Office 365 Web Services URL

If you are using **Office 365** enter your User Name & Password and click the **Discover Office 365** button. This will use the Office 365 auto discover service to find the correct web services URL for your user.

### 8.8.3.2 Create Exchange Appointment

Creates an Exchange Appointment record for a given user on a specified Exchange 2007/2010/2013 server. This includes Office 365.




Enter the **Exchange Web Services URL**. The normally takes the form https://{exchangeserver}/ews/exchange.asmx

You will need to specify a **User Name** and **Password**. This will be the user on the Exchange Server who you want to add a contact record to.

Click the **Text Login** button to verify the connection details.

You must then default the Appointment Settings.

The **Attendees** field must use email addresses.

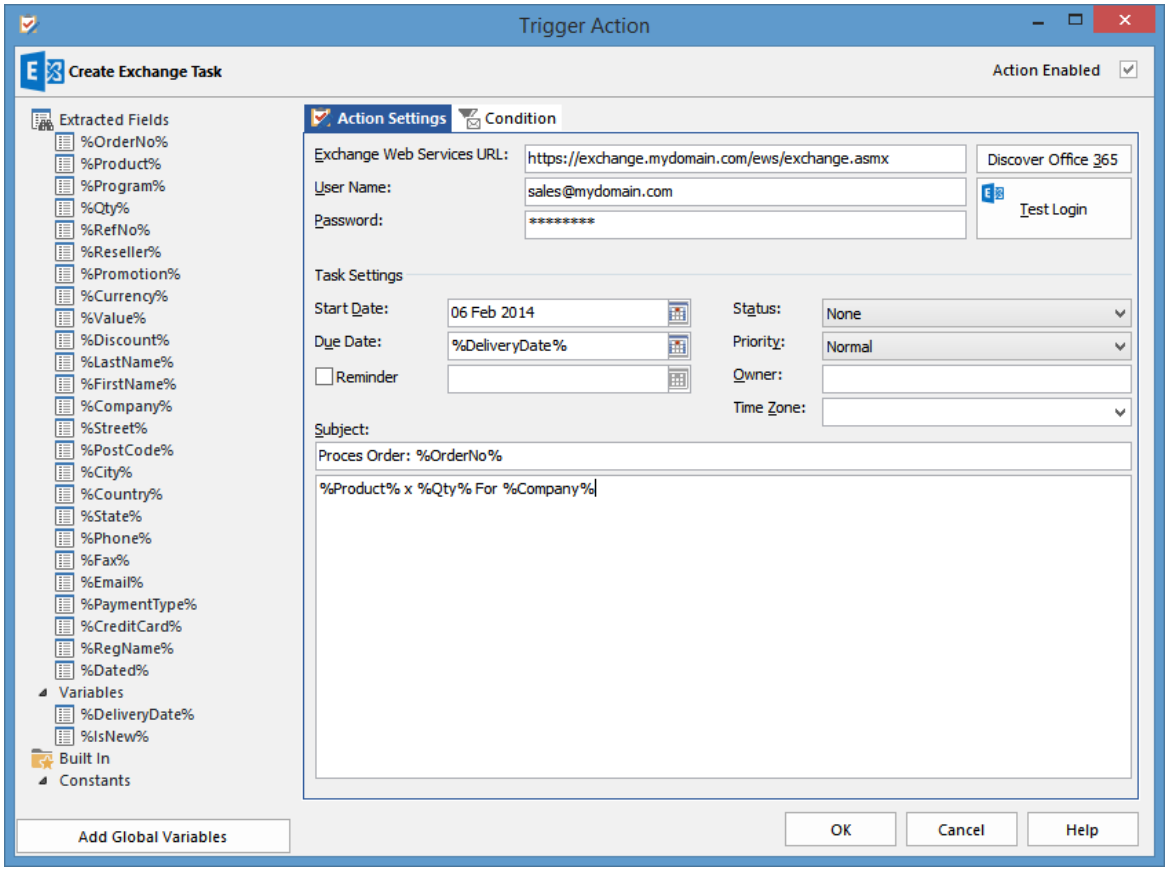
If %field% replacements are used for the **Start Date/Time** or **End Date/Time** then you need to ensure that the fields contain valid dates or times.

### Discovering Your Office 365 Web Services URL

If you are using **Office 365** enter your User Name & Password and click the **Discover Office 365** button. This will use the Office 365 auto discover service to find the correct web services URL for your user.

### 8.8.3.3 Create Exchange Task

Creates an Exchange Task record for a given user on a specified Exchange 2007/2010/2013 server. This includes Office 365.




Enter the **Exchange Web Services URL**. The normally takes the form https://{exchangeserver}/ews/exchange.asmx

You will need to specify a **User Name** and **Password**. This will be the user on the Exchange Server who you want to add a contact record to.

Click the **Text Login** button to verify the connection details.

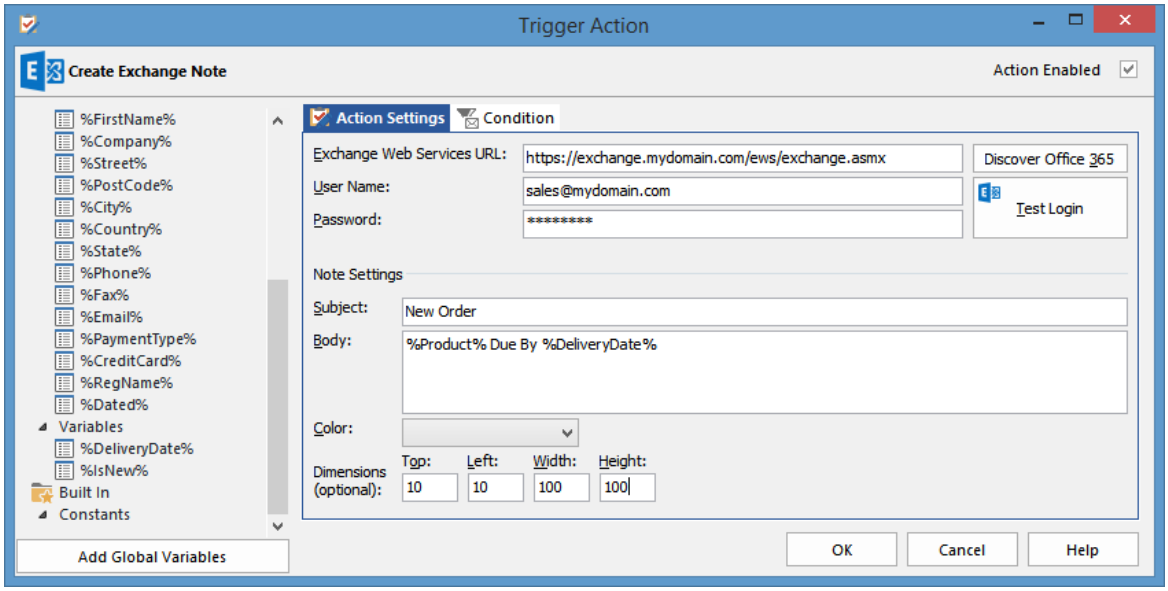
The **Start Date** and **Due Date** can use %field% replacements. If field replacements are used then the data must represent a valid date.

#### Discovering Your Office 365 Web Services URL

If you are using **Office 365** enter your User Name & Password and click the **Discover Office 365** button. This will use the Office 365 auto discover service to find the correct web services URL for your user.

### 8.8.3.4 Create Exchange Note

Creates an Exchange Note item for a given user on a specified Exchange 2007/2010/2013 server. This includes Office 365.




Enter the **Exchange Web Services URL**. The normally takes the form [https://{exchangeserver}/ews/exchange.asmx](https://{exchangeserver}/ews/exchange.asmx)

You will need to specify a **User Name** and **Password**. This will be the user on the Exchange Server who you want to add a contact record to.

Click the **Text Login** button to verify the connection details.

Enter the **Subject** & **Body**.

The **Color** of the note can be selected.

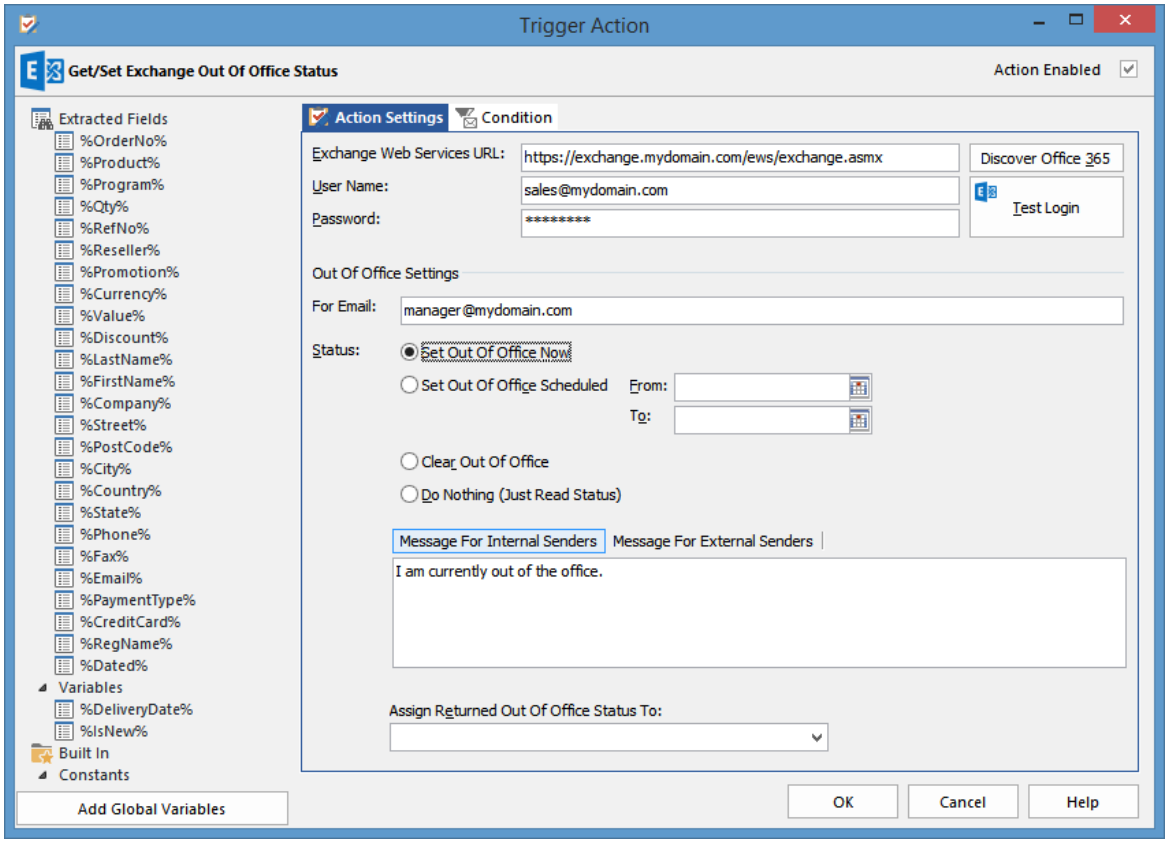
The **Dimensions** can also be specified. These are optional.

#### Discovering Your Office 365 Web Services URL

If you are using **Office 365** enter your User Name & Password and click the **Discover Office 365** button. This will use the Office 365 auto discover service to find the correct web services URL for your user.

### 8.8.3.5 Set/Get Exchange Out Of Office Status

Sets or gets the Out Of Office status for a given Microsoft Exchange 2007/2010/2013 server. This includes Office 365.




Enter the **Exchange Web Services URL**. The normally takes the form [https://{exchangeserver}/ews/exchange.asmx](https://{exchangeserver}/ews/exchange.asmx)

You will need to specify a **User Name** and **Password**. This will be the user on the Exchange Server who you want to add a contact record to.

Click the **Text Login** button to verify the connection details.

Specify the **Email** address of the user you need to set.

For the Status option select:

- Set Out Of Office Now - the user will be set to out of office immediately.
- Set Out Of Office Scheduled - the user will be set out of office between the given **From** and **To** dates.
- Clear Out Of Office - clears any currently set out of office status for the given user.
- Do Nothing - select this option if you just want to read the Out Of Office status for the given user.

When setting Out Of Office status you can define the **Message For Internal Senders** and **Message For External Senders** text. These are the messages that will be sent to anyone who sends the user a message during out of office periods.

The **Assign Returned Out Of Office Status** To drop down is used to select an Email2DB Field or Variable that you want the returned status assigned to.

The status is returned as a block of XML text - which you can parse or pass to another Trigger for further processing.

```
<Oofs>
	<Email>someone@mydomain.com</Email>
```

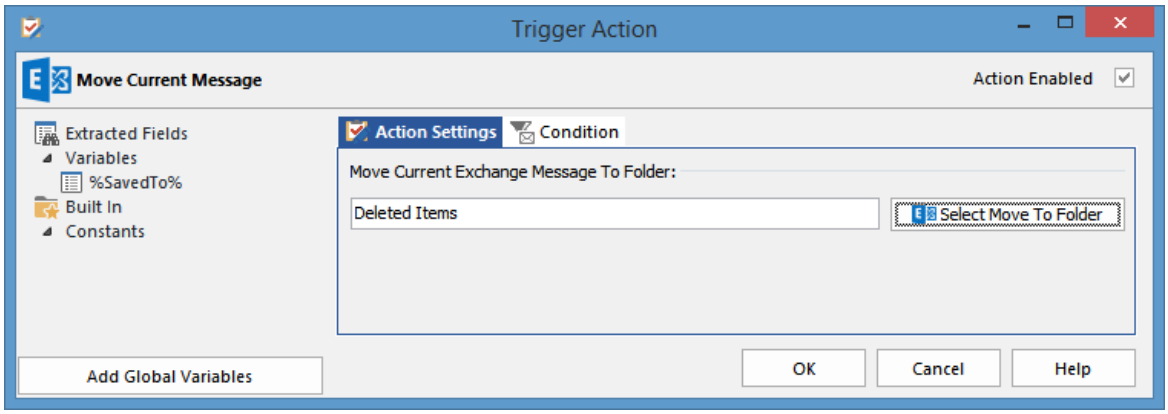
```
        <State>Enabled/Disabled/Scheduled</State>
        <StartDate>[start date/time]</StartDate>
        <EndDate>[end date/time]</EndDate>
        <InternalReply>internal reply message</InternalReply>
        <ExternalReply>external reply message</ExternalReply>
</Oofs>
```

#### Discovering Your Office 365 Web Services URL

If you are using **Office 365** enter your User Name & Password and click the **Discover Office 365** button. This will use the Office 365 auto discover service to find the correct web services URL for your user.

### 8.8.3.6 Move Message To Exchange Folder

Moves the current message to a different folder on a specified Exchange 2007/2010/2013 server. This includes Office 365.




Using the Move Message To Exchange Folder within your Trigger Actions grants you the ability to conditionally move the current message being processed to a specific folder on your Exchange/Office 365 server.

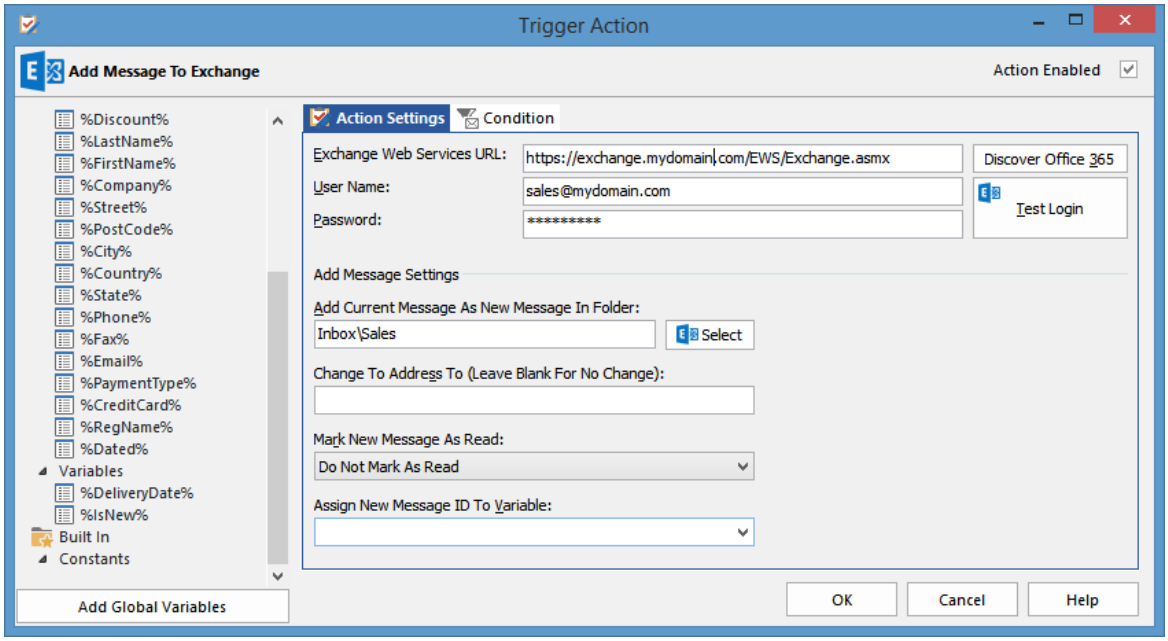
To use the action simply pick the target folder for the move from **Select Move To Folder**.

If you wish to conditionally move the item then please review the Logical Operators available within the Trigger Actions. See: Logical Operators

*Notes: This action is only available when you are using an Account that is connected to an Exchange Web Services mail folder. If this is not the Account type then you will not be able to use this Trigger Action.*

### 8.8.3.7 Add Message To Exchange

Creates a new message using the current message on a specified Exchange 2007/2010/2013 server. This includes Office 365.




This Action will create a new message in the selected folder. Any type of source message can be used (IMAP, POP3, Exchange etc). This allows you to use this action to migrate messages for separate mailboxes to an Exchange/Office 365 mailbox.

Enter the **Exchange Web Services URL**. The normally takes the form https://{exchangeserver}/ews/exchange.asmx

You will need to specify a **User Name** and **Password**. This will be the user on the Exchange Server who you want to add a contact record to.

Click the **Text Login** button to verify the connection details.

Click the **Select** button to choose a folder where the new message will be added.

By default the new message will be an exact copy of the current message being processed. You can change the To address of the message in the **Change To Address To** entry. If you specify an address here it must exist as a mailbox on the Exchange server.

New messages will be marked as Unread by default. You can change the Mark As Read status using the **Mark New Message As Read** drop down.

When the new messages is created the Message ID can be returned to a variable. This could be used to store the ID in a database or used on subsequent actions.

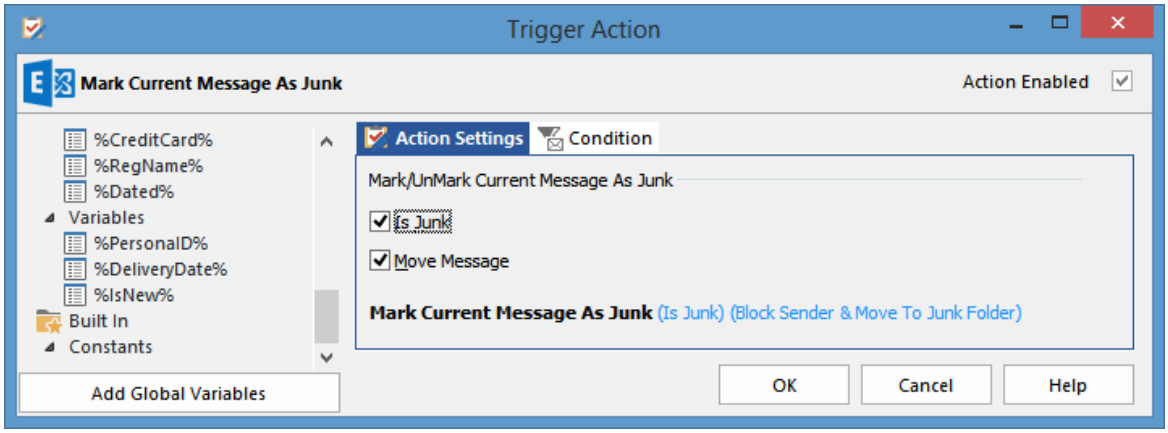
*Note: If you do not want to include attachments on the new message use the Attachments Action before the Add Message action and enable the 'Remove From Incoming Message' option. Attachments will be dropped from the message.*

### Discovering Your Office 365 Web Services URL

If you are using **Office 365** enter your User Name & Password and click the **Discover Office 365** button. This will use the Office 365 auto discover service to find the correct web services URL for your user.

### 8.8.3.8 Mark Current Message As Junk

Marks the current message as junk or not junk and optionally moves the message to the Junk folder on a specified Exchange 2007/2010/2013 server. This includes Office 365.




Using the Mark Current Message As Junk action within your Trigger Actions grants you the ability to conditionally mark the current message being processed as junk or not junk.

Select **Is Junk** to mark the current message as junk. If selected the current message will be marked as junk and the sender added to the blocked senders list. If not selected the sender will be unblocked.

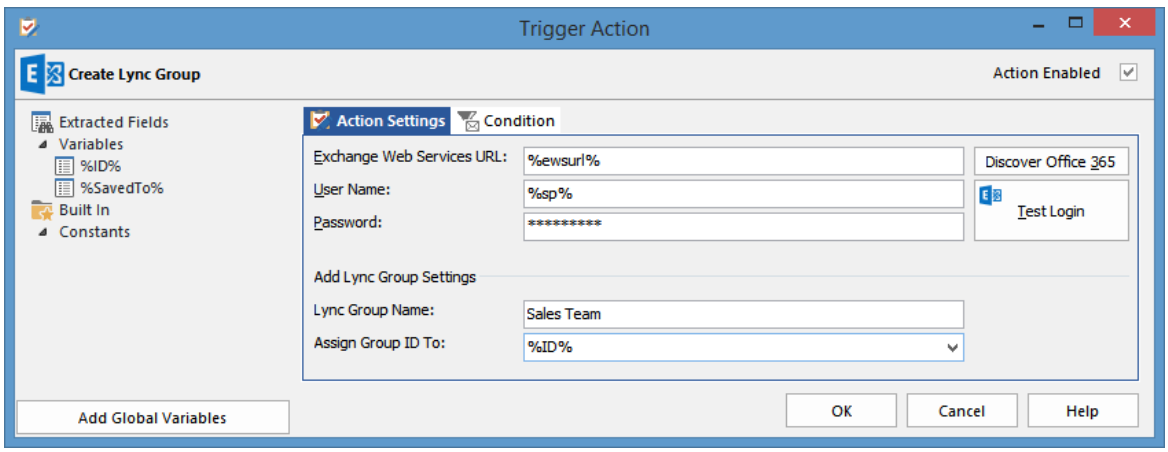
Select **Move Message** if you want to move the message to the Junk folder or Inbox depending on the Is Junk setting.

If you wish to conditionally mark an item as junk then please review the Logical Operators available within the Trigger Actions. See: Logical Operators

*Notes: This action is only available when you are using an Account that is connected to an Exchange Web Services mail folder. If this is not the Account type then you will not be able to use this Trigger Action.*

### 8.8.3.9 Create Lync Group

Creates a new Lync Contact Group on a specified Exchange 2007/2010/2013 server. This includes Office 365. Lync is the Instant Messaging service provided as part of Exchange & Office 365.



Enter the **Exchange Web Services URL**. The normally takes the form https://{exchangeserver}/ews/exchange.asmx

You will need to specify a **User Name** and **Password**. This will be the user on the Exchange Server who you want to add a contact record to.

Click the **Text Login** button to verify the connection details.

Enter the **Lync Group Name**. This is used as the display name for the group. If an existing group exists with the same name then the action will do nothing and just return the existing group ID.

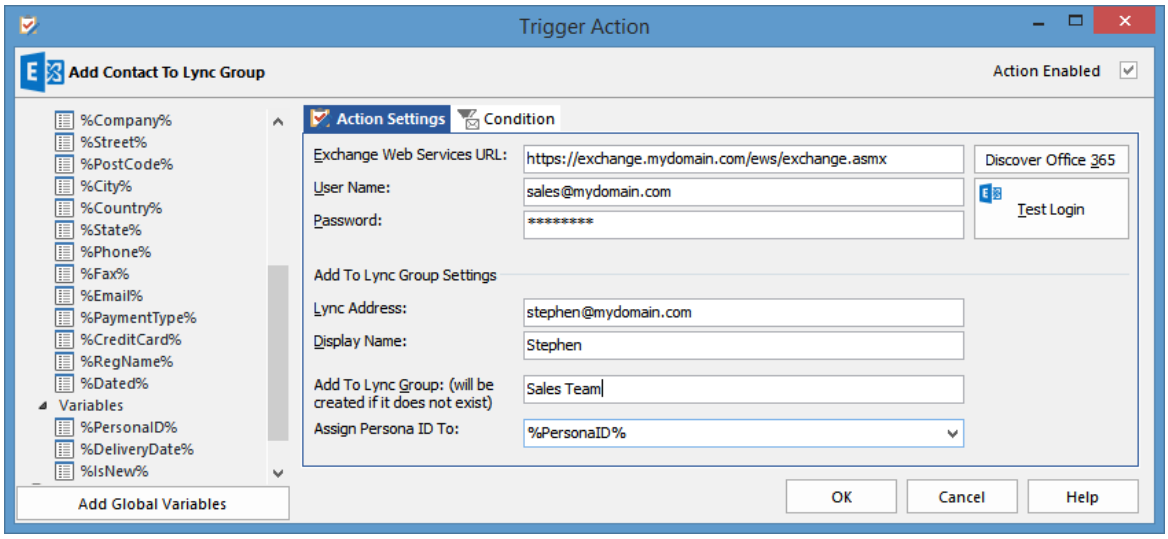
The group ID can be returned to a field or variable. Select from the **Assign Group ID** To list.

**Discovering Your Office 365 Web Services URL**

If you are using **Office 365** enter your User Name & Password and click the **Discover Office 365** button. This will use the Office 365 auto discover service to find the correct web services URL for your user.

### 8.8.3.10 Add Contact To Lync Group

Adds a Contact to a Contact Group on a specified Exchange 2007/2010/2013 server. This includes Office 365. Lync is the Instant Messaging service provided as part of Exchange & Office 365.




Enter the **Exchange Web Services URL**. The normally takes the form https://{exchangeserver}/ews/exchange.asmx

You will need to specify a **User Name** and **Password**. This will be the user on the Exchange Server who you want to add a contact record to.

Click the **Text Login** button to verify the connection details.

Enter the **Lync Address** of the contact to add. This is the IM address of the existing contact in your global address book. This would normally be the primary email address.

Enter the Display Name of the contact.

In the **Add To Lync Group** entry, enter the display name of the contact group you want to add the

contact to. If the group does not exist it will be created.

The Persona ID of the contact can be returned to the field/variable specified in **Assign Persona ID To**.

### Discovering Your Office 365 Web Services URL

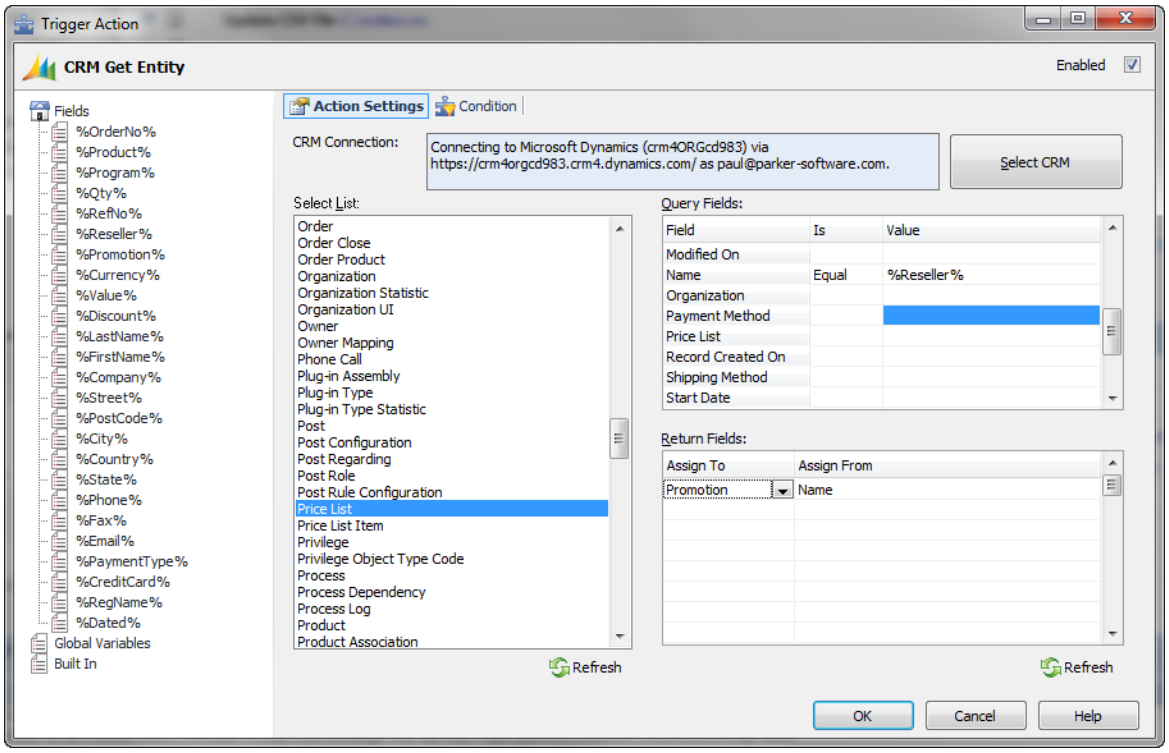
If you are using **Office 365** enter your User Name & Password and click the **Discover Office 365** button. This will use the Office 365 auto discover service to find the correct web services URL for your user.

## 8.8.4 Microsoft Dynamics & Salesforce CRM

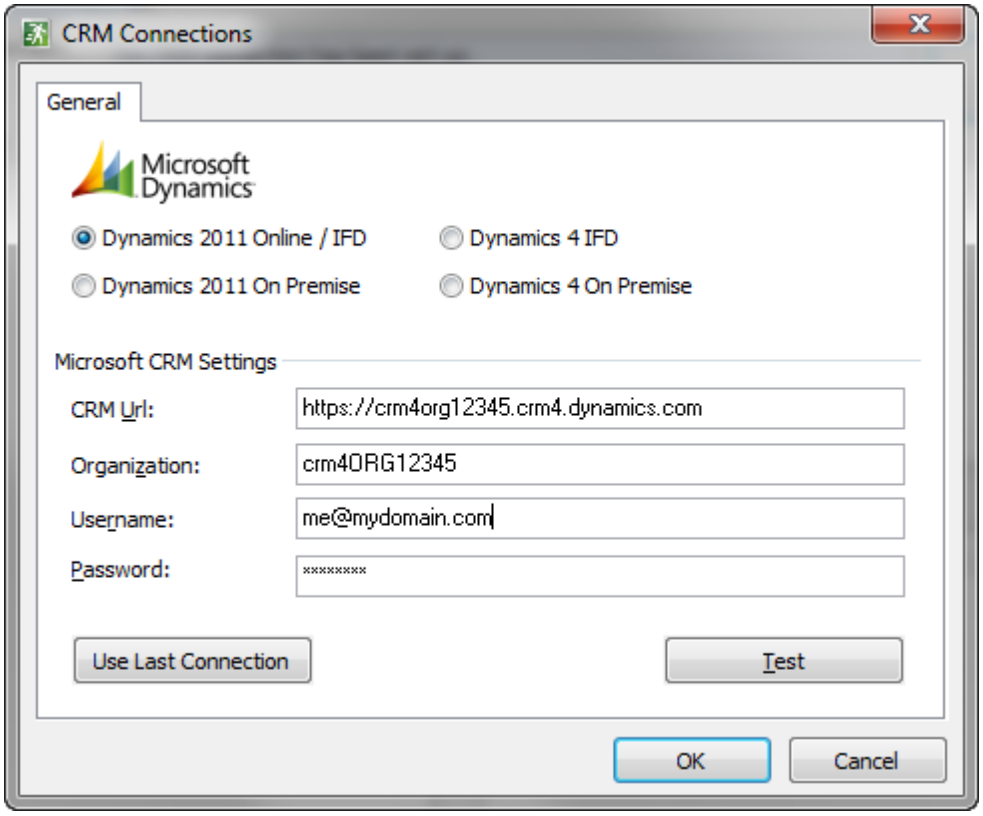
Enter topic text here.

### 8.8.4.1 Get CRM Entity

Reads an item from a Microsoft Dynamics or Salesforce CRM Entity assigns entity fields to Email2DB Fields or Variables.




Click the **Select CRM** button to connect to your CRM system.




Email2DB supports Microsoft **Dynamics 2011** (On Line or On-Premise), **Dynamics 4** and **Salesforce**

Select the CRM type that you want to connect to and enter your connection details - these will depend on the CRM type selected.

Click **Test** to verify the connection. If you have connected to your CRM system once already in the current Trigger - click the **Use Last Connection** button to use the same settings as before.

Click **OK** to save the connection details.

Email2DB will then connect to your CRM system and list the available Entity types. Select the entity you wish to read from from the **Select** list.

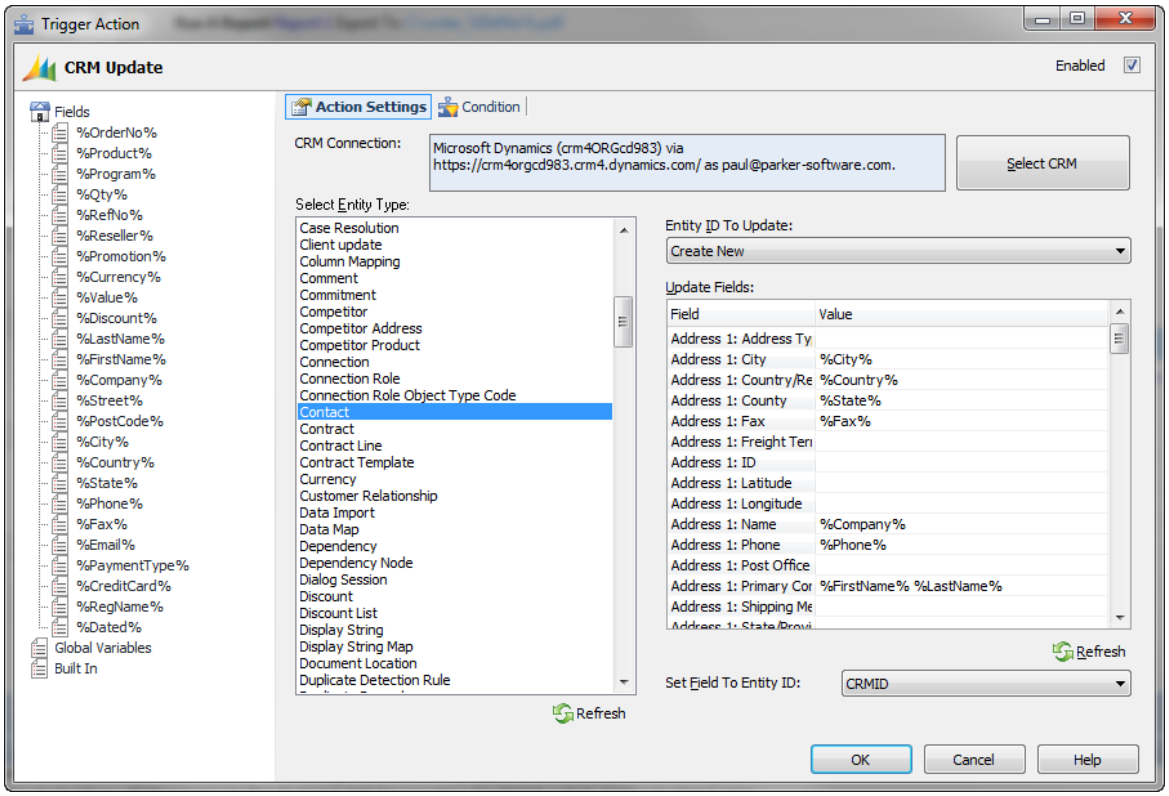
When you select an Entity the available fields for the Entity will be shown in the **Query Fields** list. Here you can choose how to read an item. Against each field the '**Is**' column can be assigned an operator (Equal To, Not Equal To, Greater Than, Less Than etc). In the **Value** column you can enter a fixed value or select an Email2DB extracted field or variable (or use a combination). Repeat for any of the Entity fields. When Email2DB executes the Trigger it will then query the CRM Entity and read the first item that matches the Query Fields.

In the **Return Fields** list you can assign any of the returned Entity fields to your Email2DB fields or variables. These can then be used on subsequent actions.

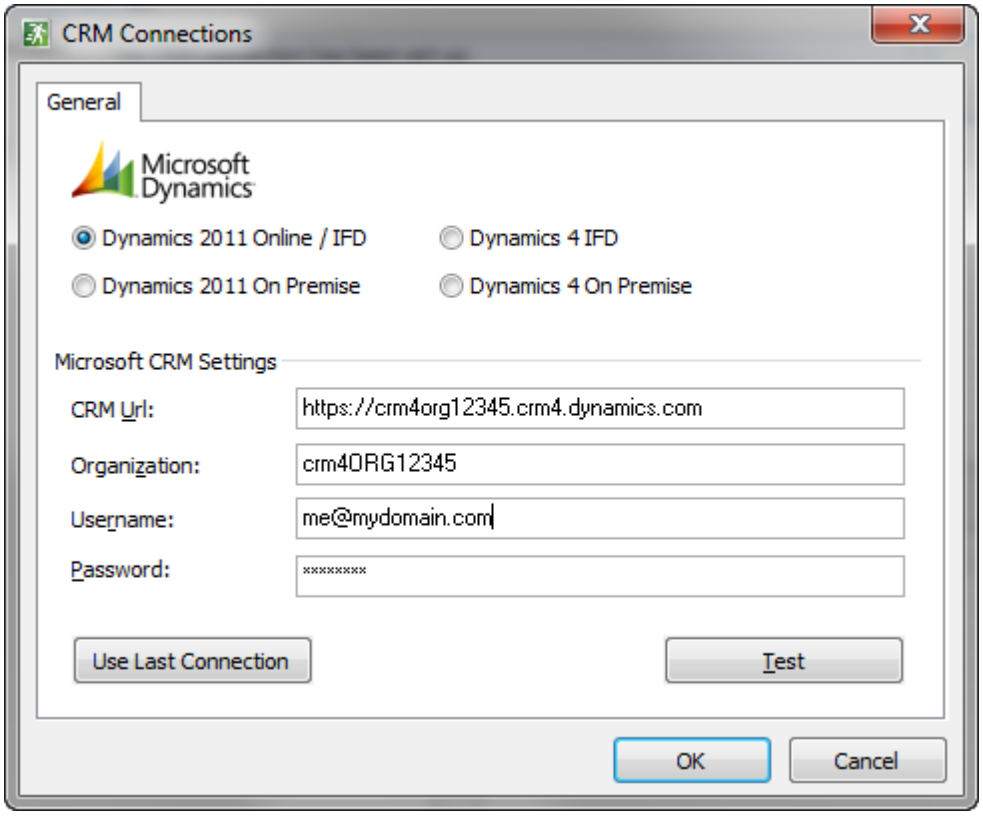
If an Entity item is not found based on the Query Fields then the Return Fields will be blank. To verify if an Entity Item is returned you can assign the 'ID' field to an Email2DB variable and then check if this is blank - as all items will have an ID.

### 8.8.4.2 Update CRM System

Creates or updates a Microsoft Dynamics or Salesforce CRM Entity Item.




Click the **Select CRM** button to connect to your CRM system.



Email2DB supports Microsoft **Dynamics 2011** (On Line or On-Premise), **Dynamics 4** and **Salesforce**

Select the CRM type that you want to connect to and enter your connection details - these will depend on the CRM type selected.

Click **Test** to verify the connection. If you have connected to your CRM system once already in the current Trigger - click the **Use Last Connection** button to use the same settings as before.

Click **OK** to save the connection details.

The Entity Types for your CRM system will then be listed. Select the **Entity Type** that you want to update.

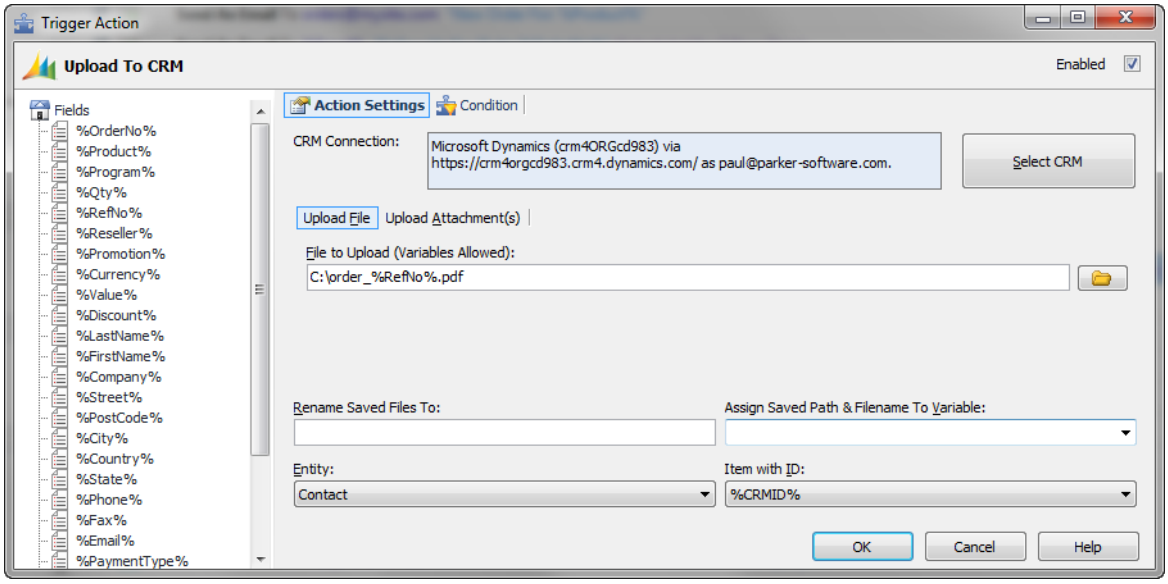
You can update an existing Entity item or create a new one for the selected Entity Type. To update an existing entity item you must first get the item ID using the Get CRM Entity action and store the ID in an Email2DB Field or Variable. You can the use this in the **Entity ID To Update** drop down. If you do not a field containing the an ID then a new item will be created.

All the fields for the selected Entity Type will be listed in the **Update Fields** list. You can then assign fixed values or Email2DB Field or Variable (or combinations) to one or more of the Entity fields.

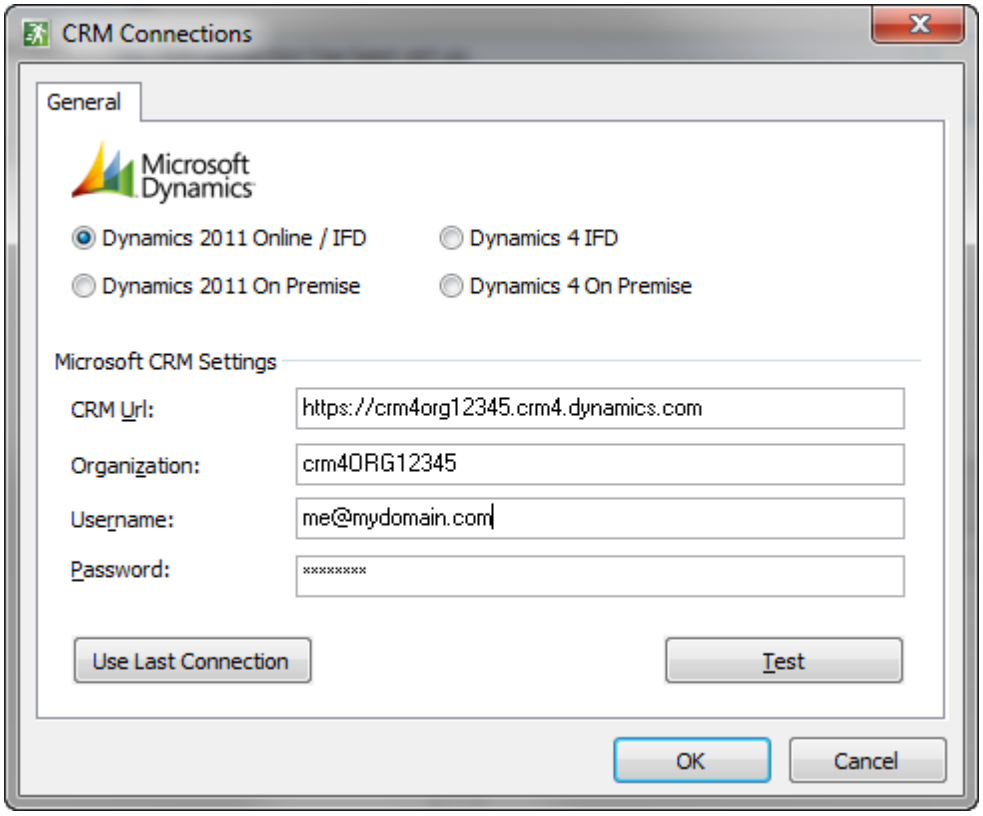
When the CRM Entity is updated or created Email2DB can assign the existing or new Entity ID to an Email2DB Field or Variable. Select from the **Set Field To Entity ID** drop down. This can then be used on subsequent actions.

### 8.8.4.3 Upload To CRM

Uploads files to Attachments to Microsoft Dynamics or Salesforce CRM Entity Item.




Click the **Select CRM** button to connect to your CRM system.




Email2DB supports Microsoft **Dynamics 2011** (On Line or On-Premise), **Dynamics 4** and **Salesforce**

Select the CRM type that you want to connect to and enter your connection details - these will depend on the CRM type selected.

Click **Test** to verify the connection. If you have connected to your CRM system once already in the current Trigger - click the **Use Last Connection** button to use the same settings as before.

Click **OK** to save the connection details.

You then have the choice of uploading a single file or uploading attachments attached to the incoming message. Select **Upload File** or **Upload Attachments(s)**

For uploading a file enter the File To Upload. This can use field %replacements% if you want to upload a file created on a previous action (such as an export report PDF).

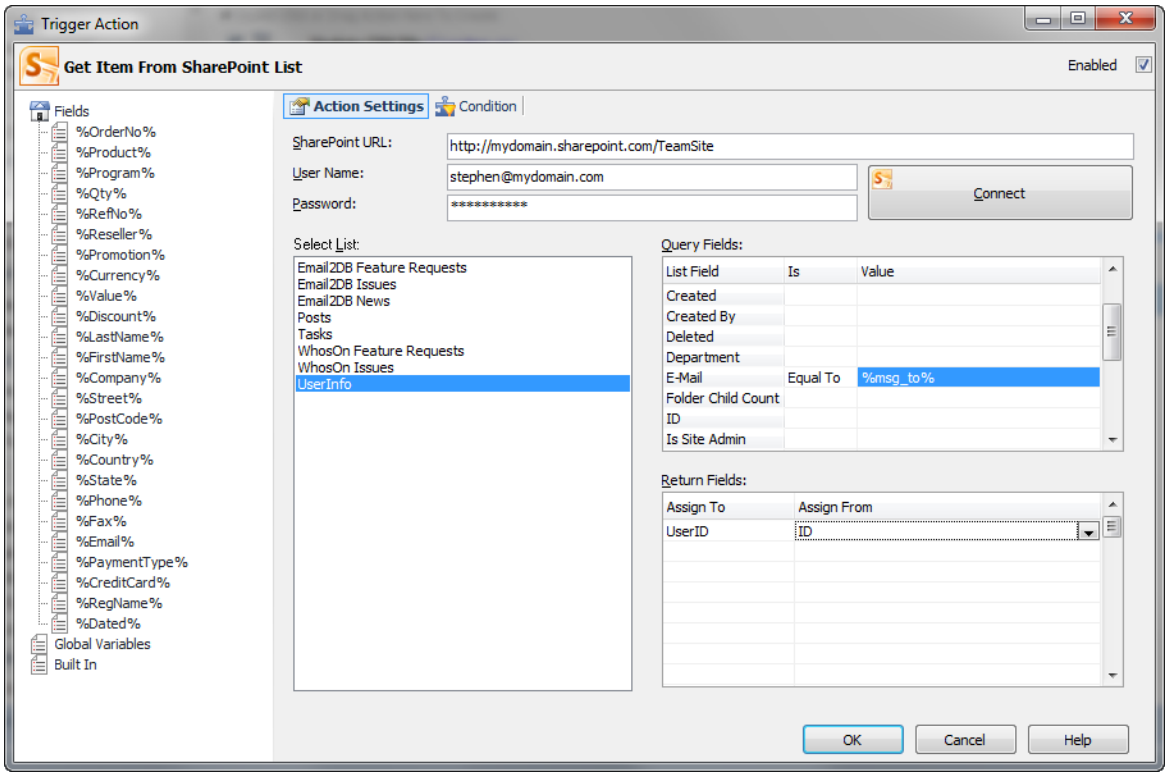
For uploading attachments enter the File Mask (eg *.PDF).

Uploaded files must be assigned to an Entity Item. Select the **Entity** and **Item With ID**. For the ID you must choose an Email2DB Field or Variable containing the CRM Entity Item ID. You can read this using the Get CRM Entity action.

## 8.8.5 Microsoft SharePoint

### 8.8.5.1 Get SharePoint Item

Reads an item from a Microsoft SharePoint list and assigns item fields to Email2DB Fields or Variables.




This Action enables you to make a query against any SharePoint list items. The query can use fixed values or values from extracted fields & variables. The data returned can be assigned to one or more Email2DB fields or variables to be used on subsequent actions.

Enter your **SharePoint URL**, **User Name** & **Password** and click the **Connect** button to connect to your SharePoint server.

The available SharePoint Lists will be shown. Select the List you want to query.

In the **Query Fields** grid you can then assign values against one or more fields in the list.

Select the **List Field** you want to query against. In the **Is** column drop down you can select:

- Equal To
- Not Equal To
- Less Than
- Greater Than
- Less Than Or Equal
- Greater Than Or Equal
- Is Blank
- Is Not Blank
- Contains, Starts With
- Membership Of

In the **Value** column enter a fixed value, or select an Email2DB Extracted Field or Variable.

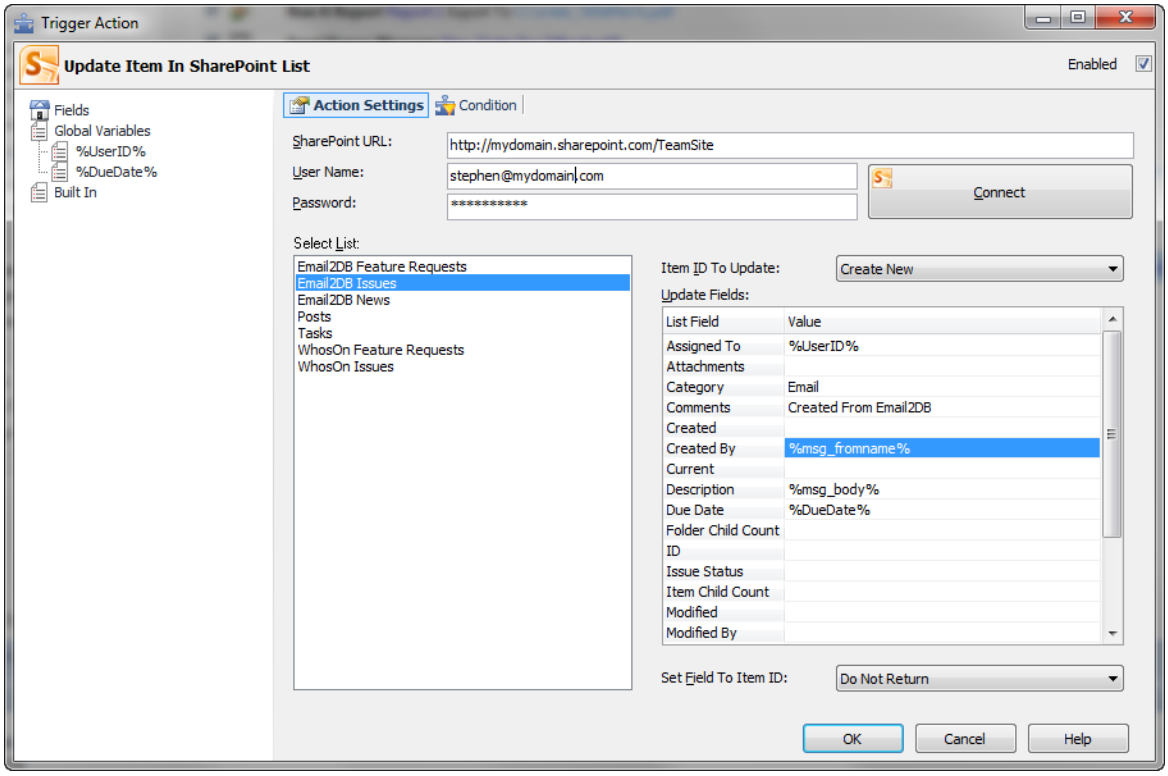
Repeat this for any of the List Fields. The Query Fields will be combined when the List is read from SharePoint. All of the Query fields must match.

In the **Return Fields** grid you select the Email2DB Fields or Variables which will be assigned to the returned List item fields.

In the **Assign To** column select the Email2DB Extracted Field or Variable you want to assign the data to. In the **Assign From** column select the SharePoint List Item field you want to assign.

### 8.8.5.2 Update SharePoint List Item

Creates a new Microsoft SharePoint List item or updates an existing one.




Enter your **SharePoint URL**, **User Name** & **Password** and click the **Connect** button to connect to your SharePoint server.

The available SharePoint Lists will be shown. Select the **List** you want to update.

By default a new List item will be created. If you want to update an existing List item select the Email2DB Field or Variable that contains the **Item ID To Update**.

All of the List fields will be shown on the **Update Fields** grid.

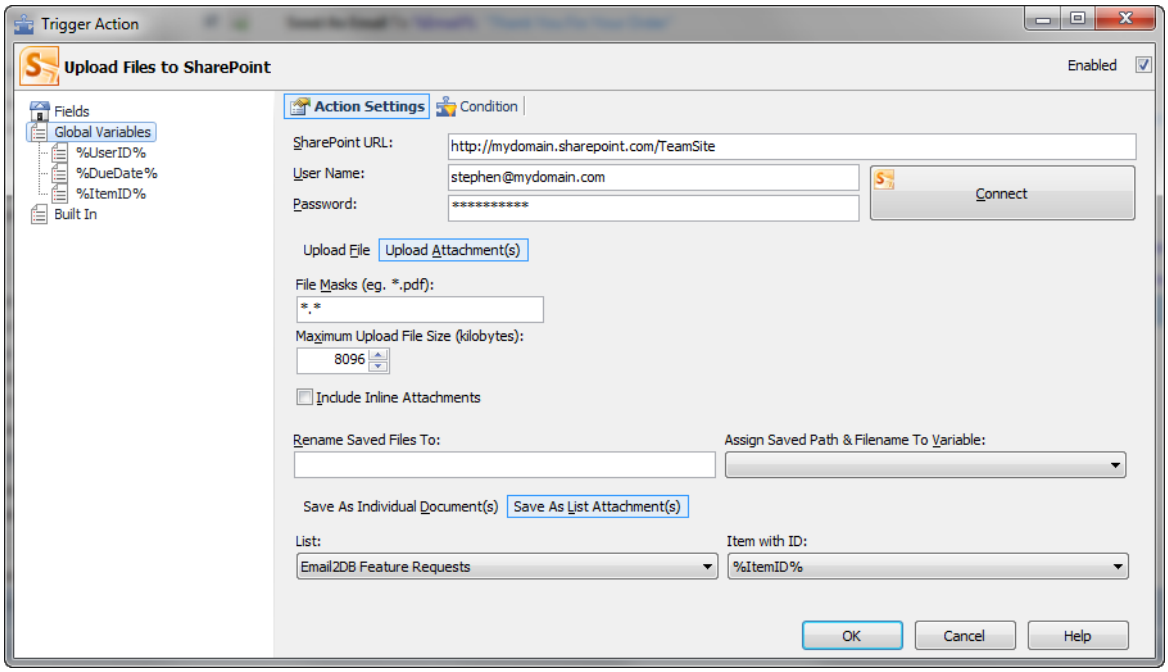
Against each field you can type a fixed value or select an Email2DB Extracted Field or Variable which will be assigned to the field.

When the update is performed the new or existing List item ID will be returned to Email2DB. You can

assign this value to an Email2DB Field or variable by selecting from the **Set Field To Item ID** list.

### 8.8.5.3 Upload To SharePoint

Uploads files or Attachments to a Microsoft Sharepoint site.




Enter your **SharePoint URL**, **User Name** & **Password** and click the **Connect** button to connect to your SharePoint server.

You can upload any file available on your file system or you can upload Attachments that were attached to the incoming message.

Click Upload File to choose a file on your system or Upload Attachment(s) to upload attached files.

In the **File Mask** entry enter the type of attachments to upload. For example, entering *.pdf will upload all PDF files. Enter *.* or just * to upload all attachments.

You can set a **Maximum Upload File Size** to stop large attachments being uploaded. Enter the value in Kilobytes.

Check **Include Inline Attachments** if you want to upload embedded images and other files that are included in HTML emails.

In the **Rename Saved Files To** entry you can optionally specify a new name for the file. You can use %fieldname% replacements in the Rename to - for example: order%orderno%.csv would rename the attachment order1234.csv if the OrderNo field contained 1234.

You can use the special field replacement '%filename%' to use the original filename as part of the renamed file. For example, suppose the incoming attachment was called "orderdata.csv" and the order number field was set to 1234 - renaming to: %filename%_No_%orderno%.csv would rename the file 'orderdata_No_1234.csv'.

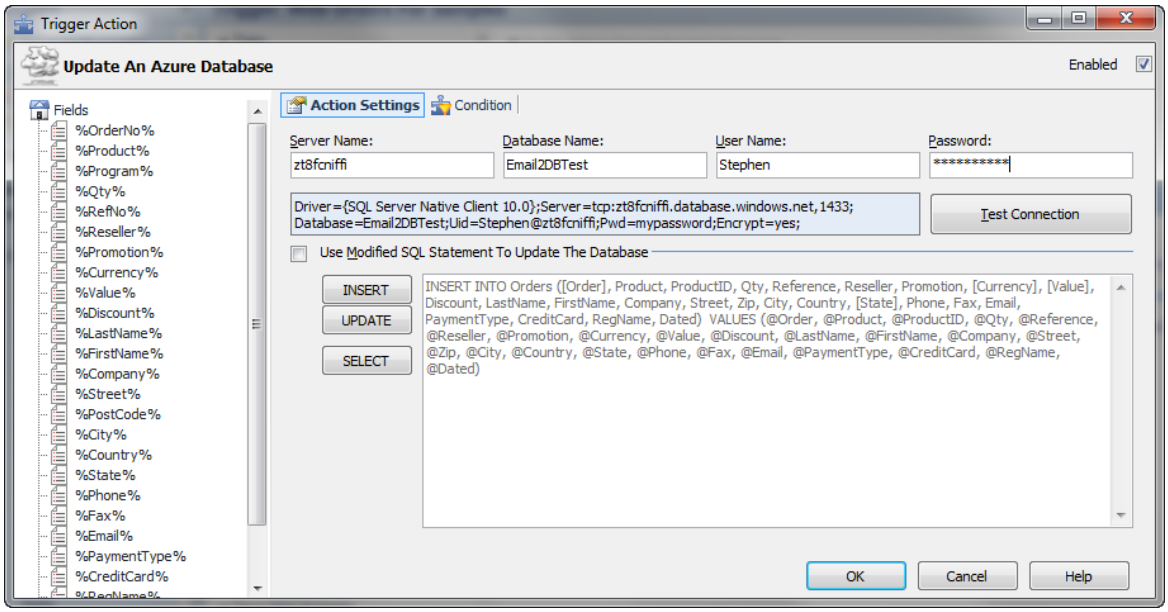
You can assign the saved path and filename to an Email2DB field or variable. Select the Field or Variable to assign using the **Assign Saved Path & Filename To Variable** drop down. Multiple saved attachments will be separated by commas.

Files uploaded to SharePoint can either be saved as **Individual Documents** or saved as attachments to a SharePoint List. Click **Save As List Attachment(s)** to save the uploaded files as List attachments. The select the **List** and the Email2DB Field or Variable which contains the List Item in the **Item With ID** drop down.

## 8.8.6 Windows Azure

### 8.8.6.1 Update An Azure Cloud Database

Updates a Microsoft Azure SQL Database with fields extracted from the incoming message.




Microsoft Azure is a cloud-based SQL Server database. See: https://www.windowsazure.com/en-us/ for more information.

You must create your Azure database and tables before you can update them with Email2DB.

The tables and field names used in the SQL statements are specified on the **'Insert Into Database Field'** tab on each individual field. Email2DB will then create the SQL INSERT statement automatically.

Enter the **Server Name**, **Database Name**, **User Name** & **Password**. You can find your server name in your Azure Management Portal.

The **Connection String** will be created automatically. Click the **Test Connection** button to verify the connection.

The UPDATE and SELECT statements will only be created if you have defined one or more of your fields as 'Key Fields'. Email2DB will then first check if a record exists with the key field values by issuing a SELECT * From... statement. It will then issue the Update statement if a record is found or the INSERT statement otherwise.

### 8.8.6.2 Get/Update Azure Table Entity

Gets or Updates a Microsoft Azure Table Entity.

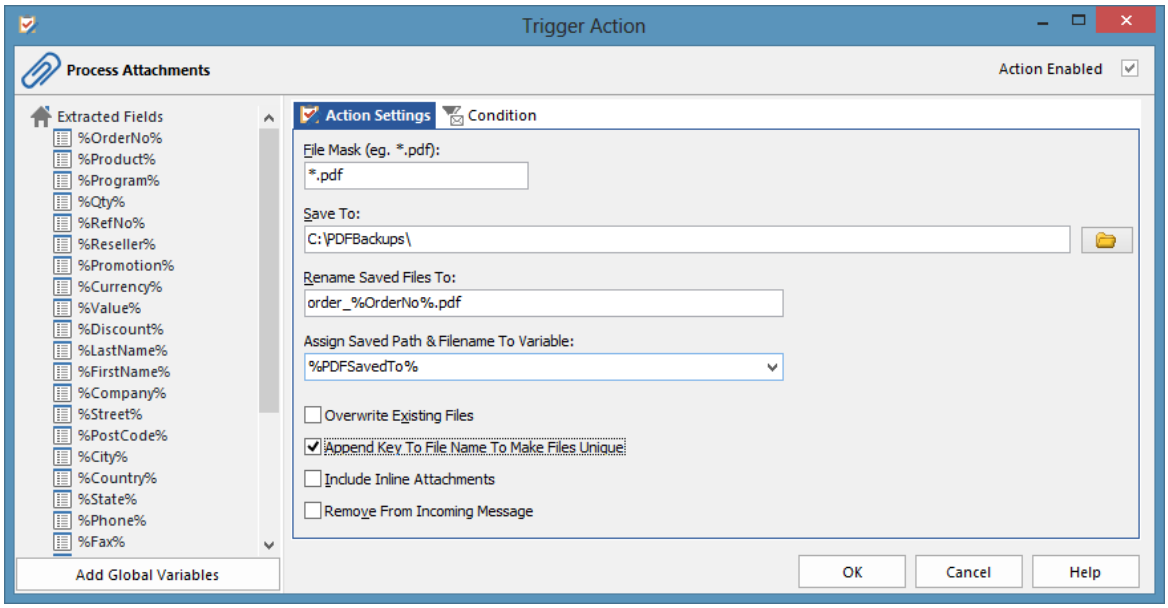
### 8.8.6.3 Get/Update Azure Blob

Gets or Updates a Microsoft Azure Blob.

## 8.8.7 Processing Attachments

### 8.8.7.1 Save Attachments

Saves attachments to specific folders on your file system.




Specify the **File Mask** for the extensions you want to save. For example: *.pdf to save all files with the extension .pdf. Use *.* to save all attachments. You can create multiple Save Attachment Actions in the same trigger if you need to process different file types differently.

In the **Save To** Folder select the folder on your file system to save the attachments to.

**Renaming Attachments**

In the **Rename To** entry you can optionally specify a new name for the file. You can use %fieldname % replacements in the Rename to - for example: order%orderno%.csv would rename the attachment order1234.csv if the OrderNo field contained 1234.

You can use the special field replacement '%filename%' to use the original file name as part of the renamed file. For example, suppose the incoming attachment was called "orderdata.csv" and the order number field was set to 1234 - renaming to: %filename%_No_%orderno%.csv would rename the file 'orderdata_No_1234.csv'.

If your rename string doesn't contain a file extension then the original extension will be used. For example: If the attachment is called 'attachment.csv' and you rename it to '%OrderNo%' - then the attachment will be renamed '1234.csv' (assuming the %OrderNo% field has a value of '1234').

You can assign the saved path and file name to an Email2DB field or variable. Select the Field or Variable to assign using the **Assign Saved Path & Filename To Variable** drop down. Multiple saved attachments will be separated by commas.

**Overwrite Existing Files**

Check this box if Email2DB should overwrite existing files.

**Append Key To File Name To Make Files Unique**

If this option is selected then Email2DB will append a date and time stamp to the file name (including

renamed files) to ensure that the file name is unique. The time stamp is in the format yyyymmddhhmmss_x

For example: order123420120307122033_1.csv

Would be saved for file order1234.csv on 7th March 2012 and 12:20:33. The _1 indicates that this is the first attempt at saving using this file name. If a file already existed with the same time stamp (for example, if Email2DB was processing multiple emails very quickly) the counter would be increased until a unique file was found.

When Email2DB saves attachments it updates the MSG_AttachmentSavedTo field which can be used in scripts to do further processing on the saved attachments. For example, you could forward specific attachments via email or launch them as applications etc.

**Include Inline Attachments**

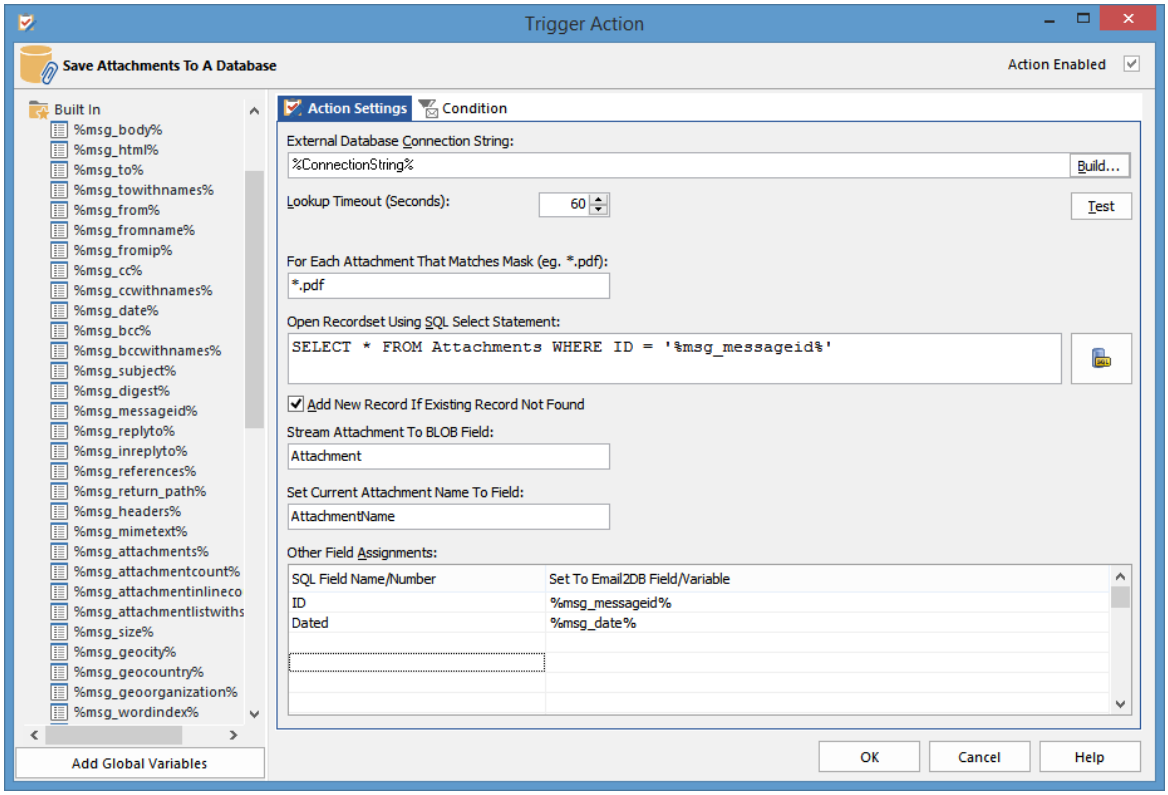
For HTML emails you can also save inline attachments - these would usually be images embedded in the HTML that don't appear as regular attachments.

See Also: Using Scripts To Process Attachments

*Note: If you are saving attachments to network folders you will need to change the user name that the Email2DB Service runs under. See: Running The Email2DB Service Under A Different User*

### 8.8.7.2 Save Attachments To A Database

Saves attachments to binary (BLOB) fields in an external SQL database.




You must specify a **Connection String** that Email2DB will use to open the data source. Click the **Build** button to build the connection string using any of the data source providers that are installed on your PC. Click the **Test** button to verify that Email2DB can login to the data source.

Enter the **Mask**. This is the file mask that attachments will be checked against. For each attachment that matches the mask, the SQL recordset will be opened and a new record added or an existing one updated.

Enter the **SQL Select Statement** to use to open a recordset in your database. This will normally be a SELECT statement, It must return all of the fields you want to update.

Select **Add New Record If Existing Record Not Found** if you want a new record inserting if the SELECT statement does not return a record.

In the **Stream Attachment To BLOB Field** entry, enter the field name in your database to which the contents of the current attachment will be assigned to. This field is required.

In the **Set Current Attachment Name To Field** enter a field name to assign the current attachment file name to. This is optional.

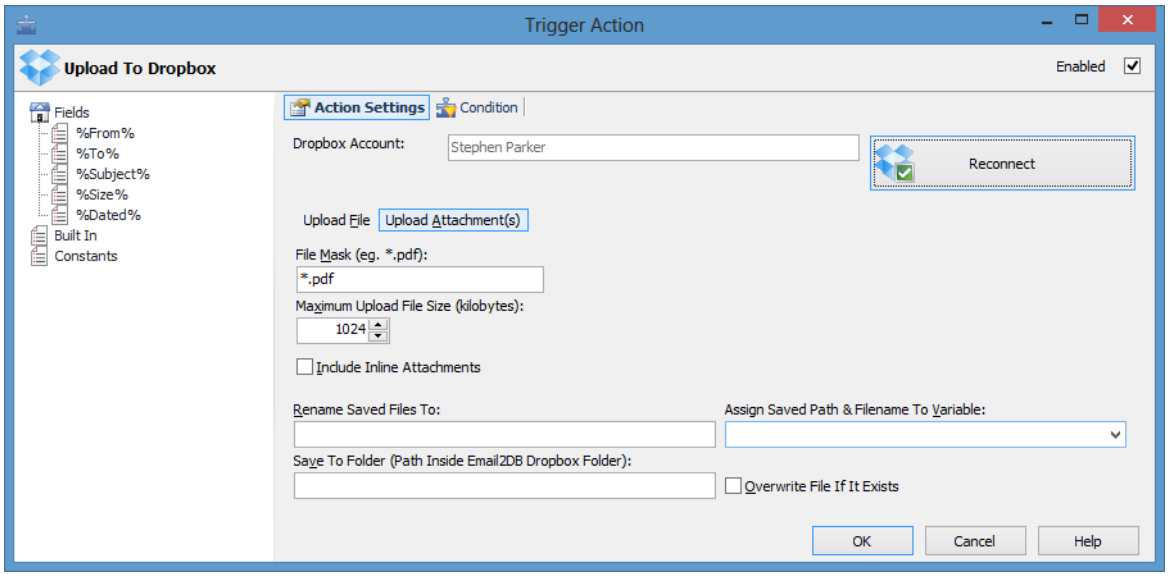
The **Other Field Assignments** grid enables you to assign Email2DB Fields, Variables and constants to other fields in your database.

## 8.8.8 Cloud Storage

### 8.8.8.1 Upload To Dropbox

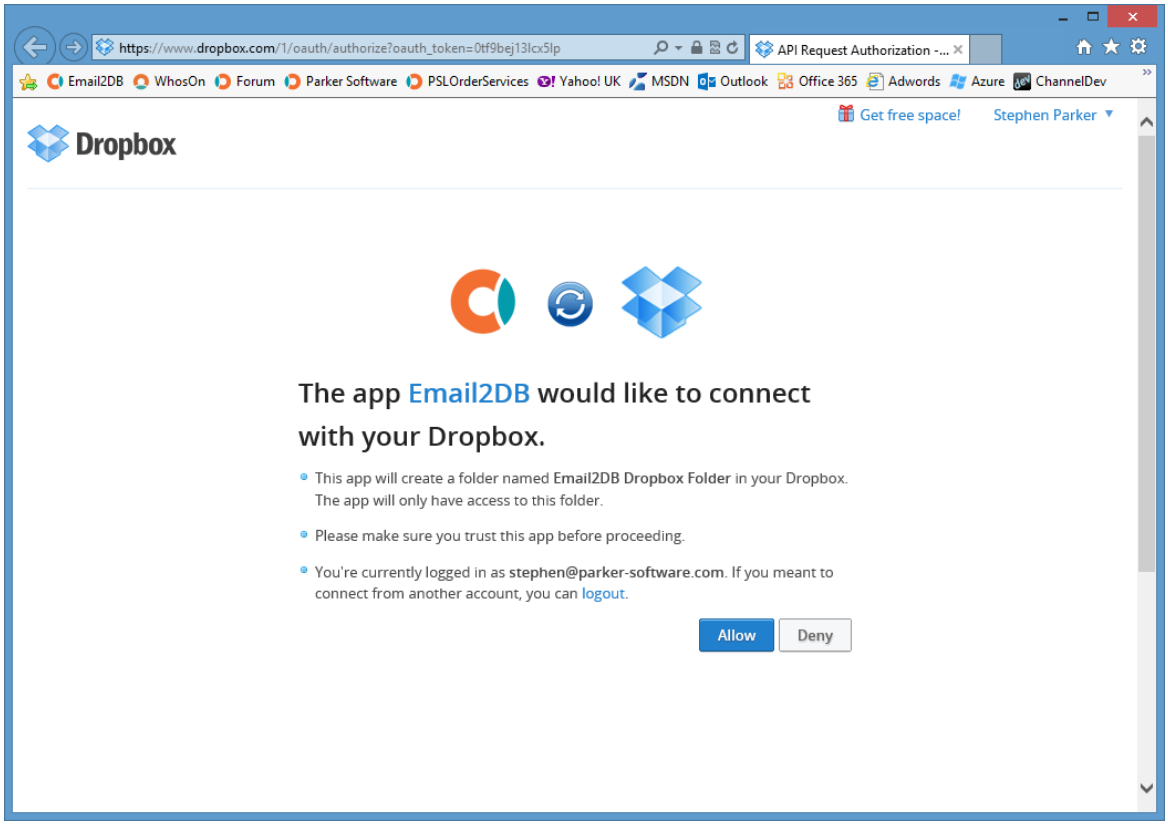
Saves files or attachments to your Dropbox. Dropbox is a web-based file hosting service operated by Dropbox Inc that uses cloud storage to enable users to store and share files and folders with others across the Internet using file synchronization. See: http://www.dropbox.com for more information.

Email2DB uses the Dropbox API to upload files. You do not need the Dropbox Client running on the Email2DB computer.




Before you can use the Dropbox Upload you need to connect to your Dropbox Account. Click the **Connect** button. This will open the Dropbox Website. Enter your Dropbox login details.

You will then be asked if you want to give Email2DB access to your Dropbox:




Click **Allow**.

*Note: Email2DB will ONLY have access to the Apps\Email2DB Dropbox Folder\ in your Dropbox. It has no access outside this folder.*

Close your browser window. Your Dropbox account name will then show in the **Dropbox Account** entry.

You then have the choice of uploading a single file or uploading attachments attached to the incoming message. Select **Upload File** or **Upload Attachments(s)**

For uploading a file enter the **File To Upload**. This can use field %replacements% if you want to upload a file created on a previous action (such as an export report PDF).

For uploading attachments enter the **File Mask** (eg *.PDF).

You can set a **Maximum Upload File Size** to stop large attachments being uploaded. Enter the value in Kilobytes.

Check **Include Inline Attachments** if you want to upload embedded images and other files that are included in HTML emails.

In the **Rename Saved Files To** entry you can optionally specify a new name for the file. You can use %fieldname% replacements in the Rename to - for example: order%orderno%.csv would rename the attachment order1234.csv if the OrderNo field contained 1234.

You can use the special field replacement '%filename%' to use the original filename as part of the renamed file. For example, suppose the incoming attachment was called "orderdata.csv" and the order number field was set to 1234 - renaming to: %filename%_No_%orderno%.csv would rename the file 'orderdata_No_1234.csv'.

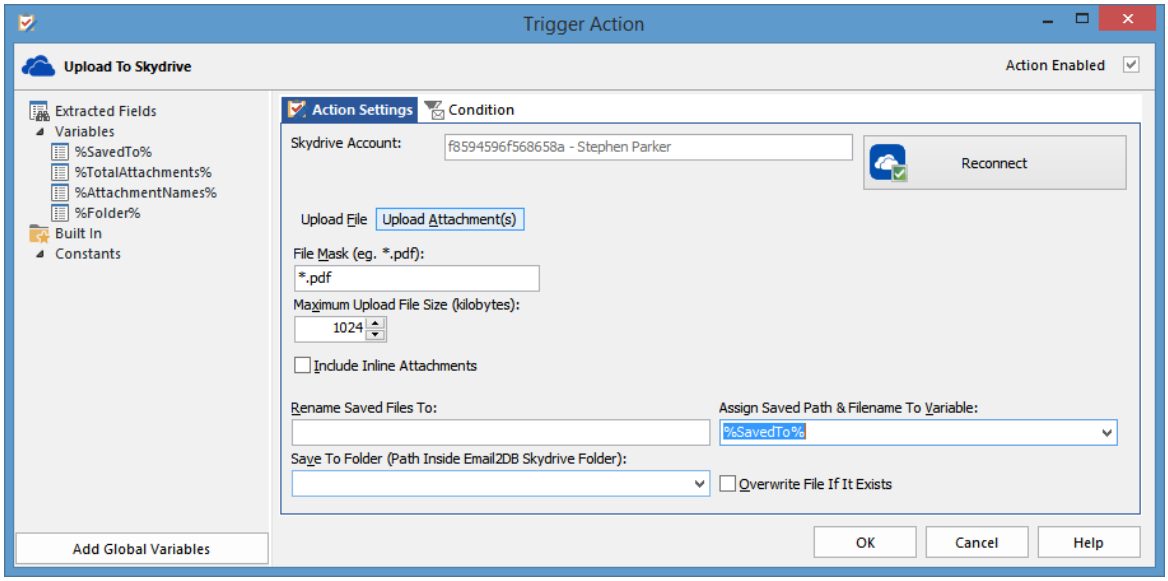
You can assign the saved path and filename to an Email2DB field or variable. Select the Field or Variable to assign using the **Assign Saved Path & Filename To Variable** drop down. Multiple saved attachments will be separated by commas.

Enter the **Save To Folder** if you want to save the files to a different folder within your Dropbox Apps \Email2DB Dropbox Folder\.

### 8.8.8.2 Upload To SkyDrive

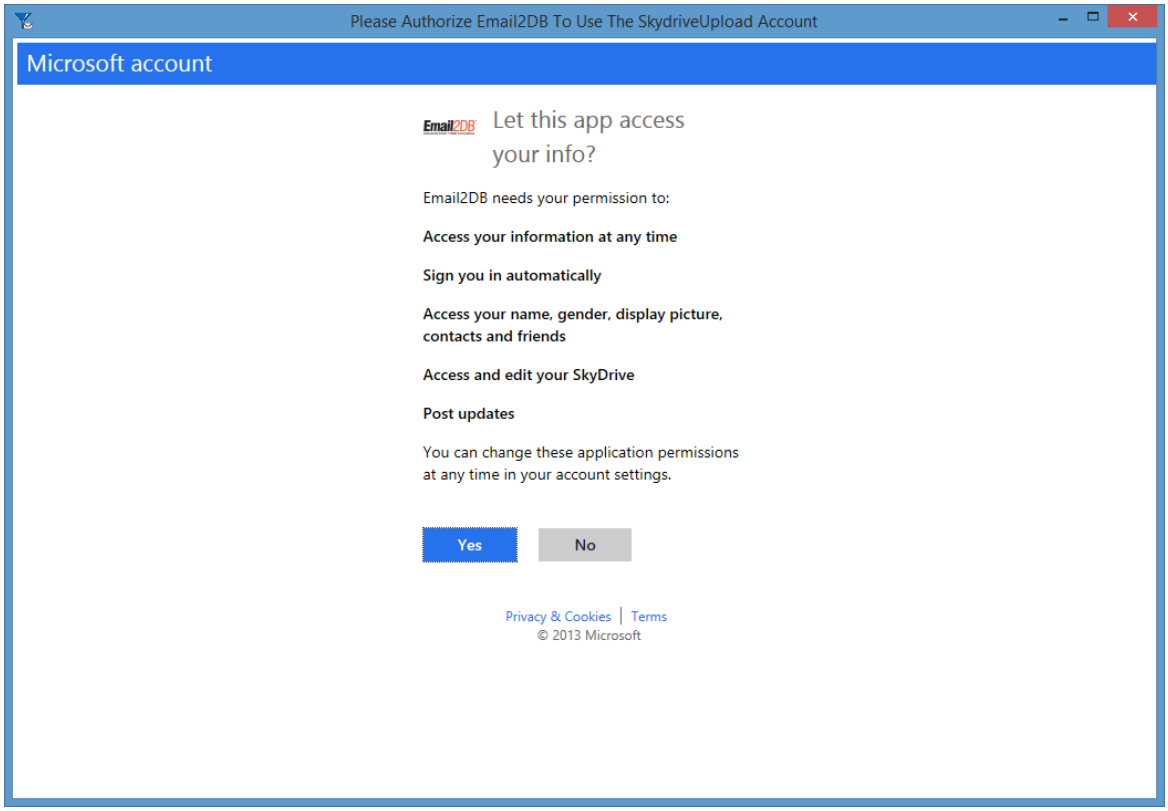
Saves files or attachments to your SkyDrive. SkyDrive is a web-based file hosting service operated by Microsoft that uses cloud storage to enable users to store and share files and folders with others across the Internet using file synchronization. See: http://skydrive.live.com for more information.

Email2DB uses the SkyDrive API to upload files. You do not need the SkyDrive Client running on the Email2DB computer.




Before you can use the Skydrive Upload you need to connect to your Skydrive Account. Click the **Connect** button. This will open the Skydrive Website. Enter your Microsoft Account login details.

You will then be asked if you want to give Email2DB access to your Skydrive:




Click **Yes**

After a few moments your Account Name will appear in the **Skydrive Account** entry.

You then have the choice of uploading a single file or uploading attachments attached to the incoming message. Select **Upload File** or **Upload Attachments(s)**

For uploading a file enter the **File To Upload**. This can use field %replacements% if you want to upload a file created on a previous action (such as an export report PDF).

For uploading attachments enter the **File Mask** (eg *.PDF).

You can set a **Maximum Upload File Size** to stop large attachments being uploaded. Enter the value in Kilobytes.

Check **Include Inline Attachments** if you want to upload embedded images and other files that are included in HTML emails.

In the **Rename Saved Files To** entry you can optionally specify a new name for the file. You can use %fieldname% replacements in the Rename to - for example: order%orderno%.csv would rename the attachment order1234.csv if the OrderNo field contained 1234.

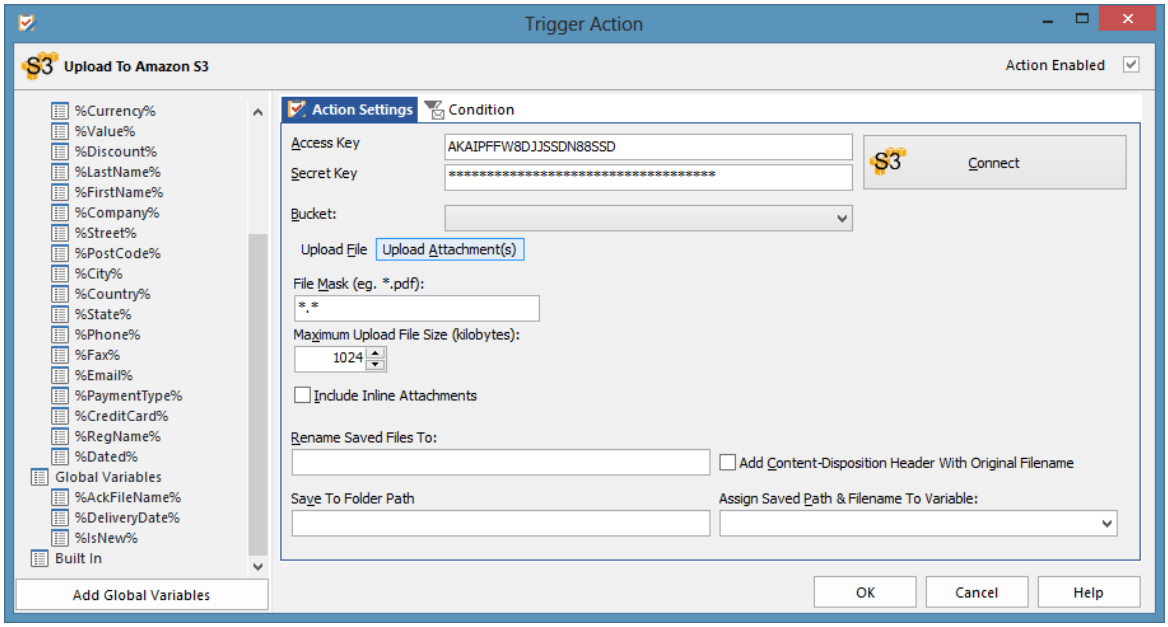
You can use the special field replacement '%filename%' to use the original filename as part of the renamed file. For example, suppose the incoming attachment was called "orderdata.csv" and the order number field was set to 1234 - renaming to: %filename%_No_%orderno%.csv would rename the file 'orderdata_No_1234.csv'.

You can assign the saved path and filename to an Email2DB field or variable. Select the Field or Variable to assign using the **Assign Saved Path & Filename To Variable** drop down. Multiple saved attachments will be separated by commas.

Enter the **Save To Folder** if you want to save the files to a different folder within your SkyDrive.

### 8.8.8.3 Upload To Amazon S3

Uploads files or Attachments to Amazon Simple Storage Service (S3). http://aws.amazon.com/s3/




Enter your Amazon S3 **Access Key** & **Secret Key** and click the **Connect** button to connect to your Amazon S3 Storage Account. Select the **Bucket** that you want to use.

You can upload any file available on your file system or you can upload Attachments that were attached to the incoming message.

Click **Upload File** to choose a file on your system or **Upload Attachment(s)** to upload attached files.

In the **File Mask** entry enter the type of attachments to upload. For example, entering *.pdf will upload all PDF files. Enter *.* or just * to upload all attachments.

You can set a **Maximum Upload File Size** to stop large attachments being uploaded. Enter the value in Kilobytes.

Check **Include Inline Attachments** if you want to upload embedded images and other files that are included in HTML emails.

In the **Rename Saved Files To** entry you can optionally specify a new name for the file. You can use %fieldname% replacements in the Rename to - for example: order%orderno%.csv would rename the attachment order1234.csv if the OrderNo field contained 1234.

You can use the special field replacement '%filename%' to use the original filename as part of the renamed file. For example, suppose the incoming attachment was called "orderdata.csv" and the order number field was set to 1234 - renaming to: %filename%_No_%orderno%.csv would rename the file 'orderdata_No_1234.csv'.

You can assign the saved path and filename to an Email2DB field or variable. Select the Field or Variable to assign using the **Assign Saved Path & Filename To Variable** drop down. Multiple

saved attachments will be separated by commas.

Enter the **Save To Folder Path** if you want to save the files to a different folder within your S3 Bucket.

#### 8.8.8.4 Upload To Google Drive

Saves files or attachments to your Google Drive. Google Drive is a web-based file hosting service operated by Google that uses cloud storage to enable users to store and share files and folders with others across the Internet using file synchronization.

Email2DB uses the Google Drive API to upload files. You do not need the Google Drive Client running on the Email2DB computer.

#### 8.8.8.5 Download From Dropbox

Downloads a file from your Dropbox and stores it to a folder on your local file system. Assigns the local file path to a field or variable.

#### 8.8.8.6 Download From SkyDrive

Downloads a file from your SkyDrive and stores it to a folder on your local file system. Assigns the local file path to a field or variable.

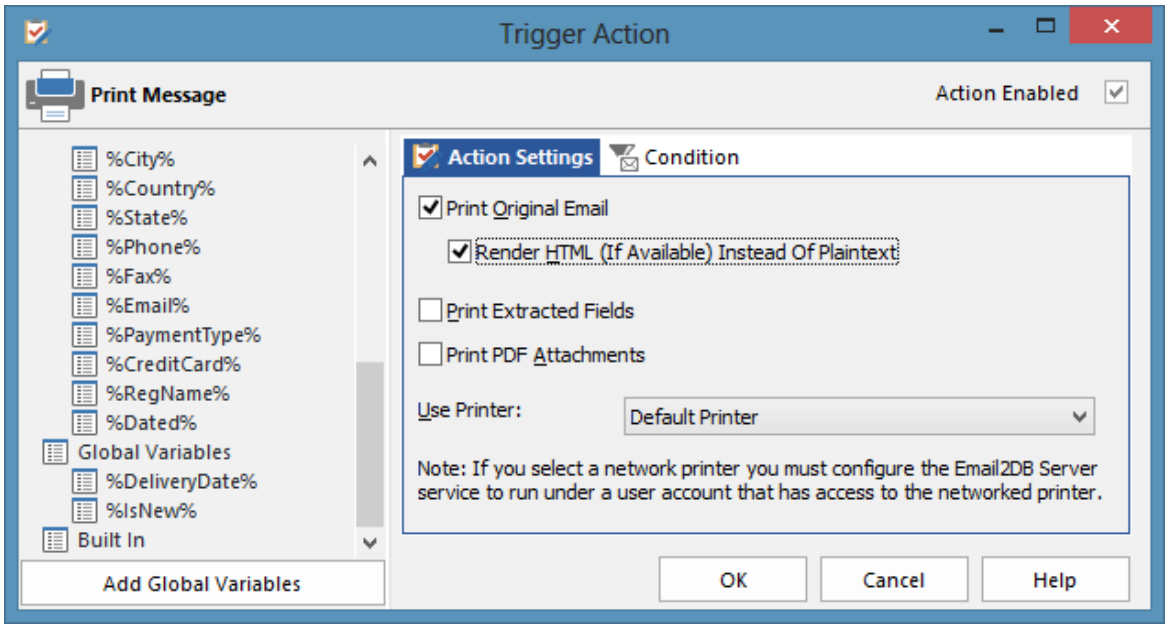
#### 8.8.8.7 Download From Google Drive

Downloads a file from your Google Drive and stores it to a folder on your local file system. Assigns the local file path to a field or variable.

## 8.8.9 Reporting

### 8.8.9.1 Print Message

Prints the original email and/or a report of extracted fields and PDF attachments.




Enable the **Print Original Email** option to print the incoming email.

Enable **Render HTML Instead Of Plaintext** if you want HTML formatted messages to be printed as they would appear in a browser. If this option is not selected the Email2DB will always print the plaintext version of the message.

Enable the **Print Extracted Fields** option to print a table showing extracted field names and values.

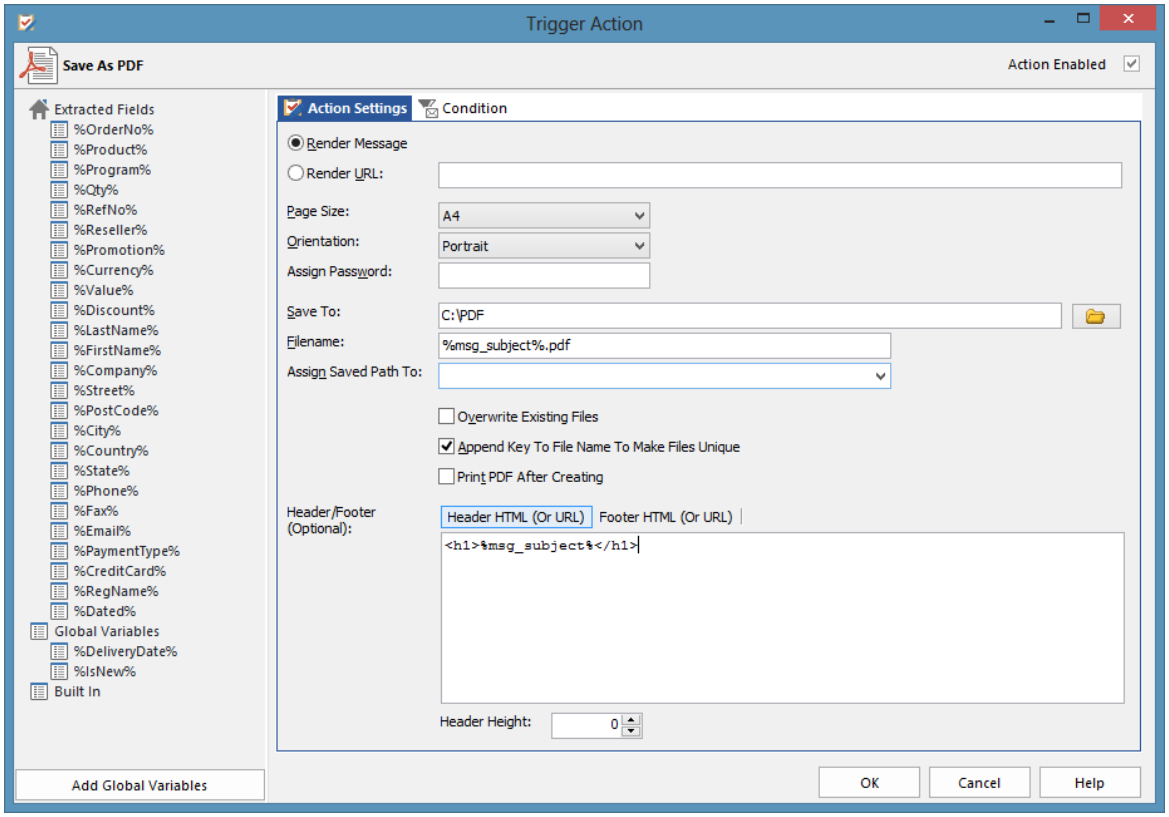
Enable the **Print PDF Attachments** option if you want Email2DB to print PDF file attachments.

You can pre-select the printer to use from the **Use Printer** drop down.

*Note: If you select a network printer the Email2DB service may not have permission to print to it. This is because the Email2DB service runs under the SYSTEM account by default and the SYSTEM account cannot access any network resources. See: Running The Email2DB Service Under A Different User for more information.*

### 8.8.9.2 Save As PDF

Renders the incoming message or a given URL as a PDF file that is saved as a file on your system.




Click **Render Message** to render in the incoming message. This will render the message as it would appear in a web browser - including all images.

Click **Render URL** to render any web page. Enter the full URL to the web page you want to save as a PDF.

Select the **Page Size** & **Orientation**.

You can also enter an open **Password**. Users opening the PDF must enter this password before they can view the content.

Select the **Save To** folder.

Enter the **File Name**.

You can assign the full save path & file name to an Email2DB Field or Variable to use on subsequent actions (for example, if you wanted to attach the file to an outgoing message). Select from the **Assign Saved Path To** drop down.

**Overwrite Existing Files**

Check this box if Email2DB should overwrite existing files.

**Append Key To File Name To Make Files Unique**

If this option is selected then Email2DB will append a date and time stamp to the file name (including renamed files) to ensure that the file name is unique. The time stamp is in the format yyyymmddhhmmss_x

For example: order123420120307122033_1.pdf

Would be saved for file order1234.pdf on 7th March 2012 and 12:20:33. The _1 indicates that this is the first attempt at saving using this file name. If a file already existed with the same time stamp (for example, if Email2DB was processing multiple emails very quickly) the counter would be increased until a unique file was found.

**Print PDF After Creating**

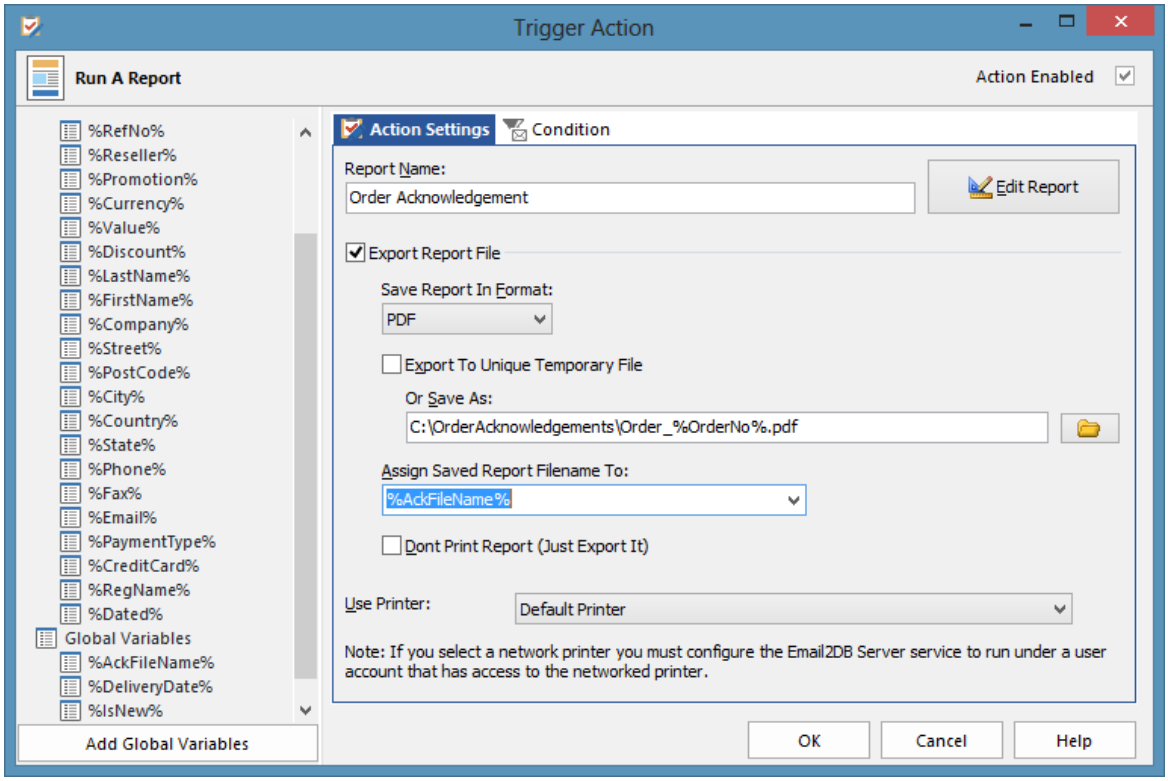
Check this box if you want Email2DB to print the PDF file after it is created. The PDF will be sent to the Default Printer.

**Header/Footer**

You can also add **Header** & **Footer HTML**. This will be rendered at the top & bottom of the page. Any HTML can be used in addition to %field replacements%. You can also enter the Header/Footer **Height** in pixels.

### 8.8.9.3 Run A Report

Creates a report using a pre-defined report template. Reports can be printed, and/or exported to various formats, including PDF, HTML, Rich text and Excel. Exported reports can be attached to outgoing emails.




Enter a **Report Name**.

Click the **Edit Report** to start the Report Designer.

Enable the **Export Report File** to export the report at run time. Select the file format from the Save Report In Format drop down. Available types are:

- Acrobat PDF
- Rich Text (which can be opened in Word)

- HTML
- Tagged Image Format Files (images that are ideal for Faxing or displaying on web pages)
- Text Files
- Excel Spreadsheet Files (version 8)

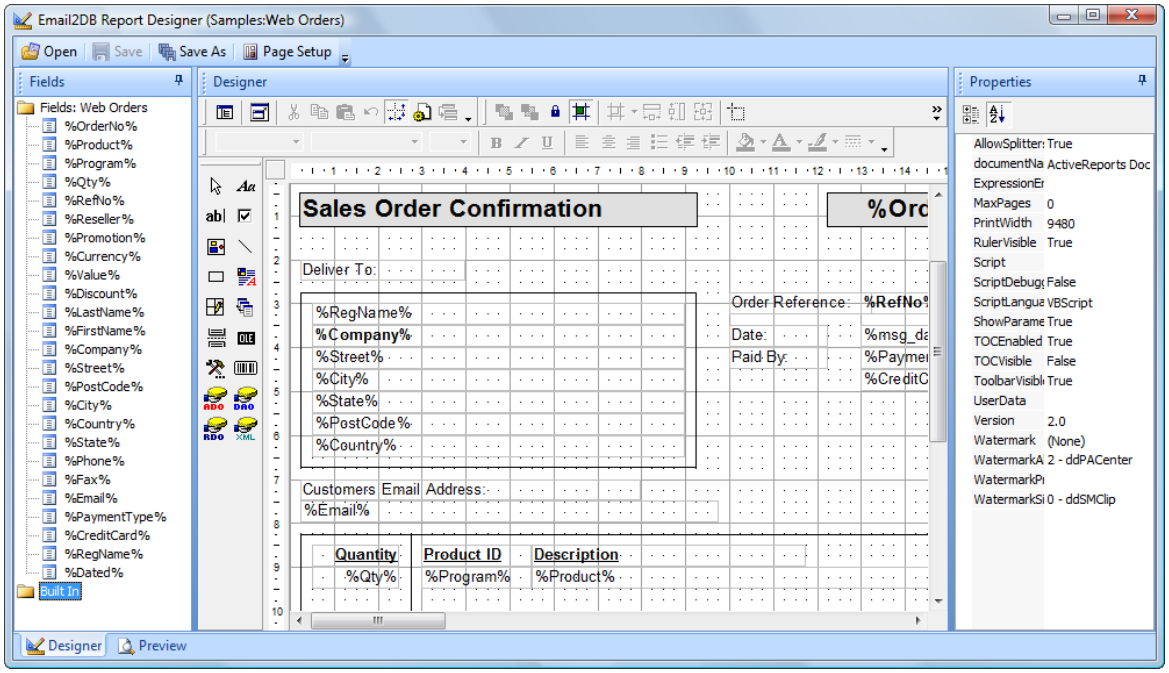
The **Export To Unique Temporary File Option** will create a unique file for each saved report. This file will be deleted when the Trigger has finished processing all actions for the current message. Alternatively you can specify a file name in the **Or Save As** entry - this can use %field% replacements.

Use the **Assign Saved Report Filename To** drop down to select an Email2DB Field or Variable to assign the exported file name to. You can then use this %fieldname% on other Actions - such as the Send Email action to add the report as an attachment.

Select the **Don't Print Report (Just Export It)** option if you do not want the report to be printed at run time.

## Using The Report Designer

Email2DB includes a report designer that allows you to create custom report templates. Reports can use extracted field data and link to external data sources.

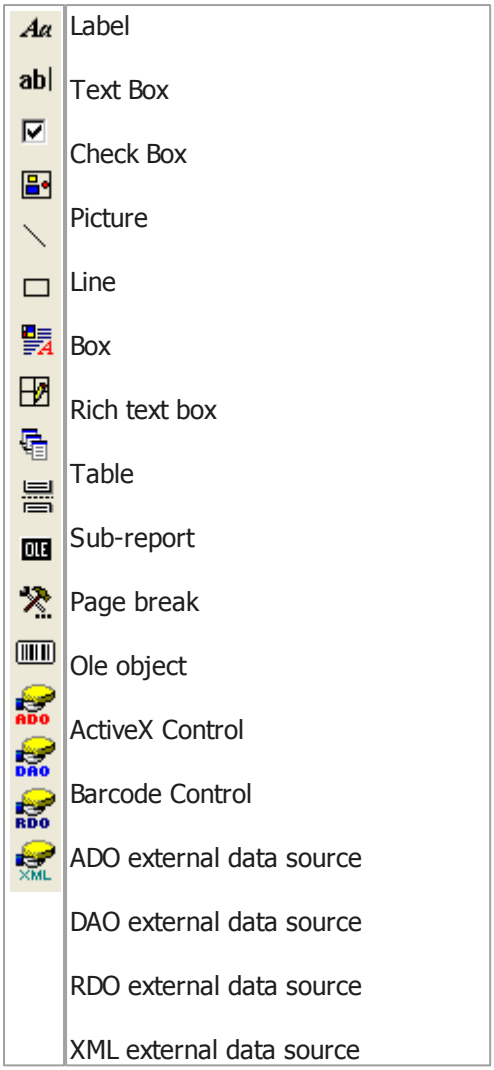



The Extracted Fields and built-in fields will be listed in the Trigger Fields tree. You can drag any of these onto the report designer surface.

After you have dragged a field to the designer, click it to edit its properties in the Property Toolbox. You can change colors, alignment, borders, fonts etc. You should also re-size the field so that it will fit the data contents. (Note: If the 'CanGrow' property is set to True then the field height will grow automatically based on the field content).

You can also change field properties via the Toolbar and by right-clicking a field in the designer.

You can also draw additional items onto the designer from the left hand Toolbox. These include:



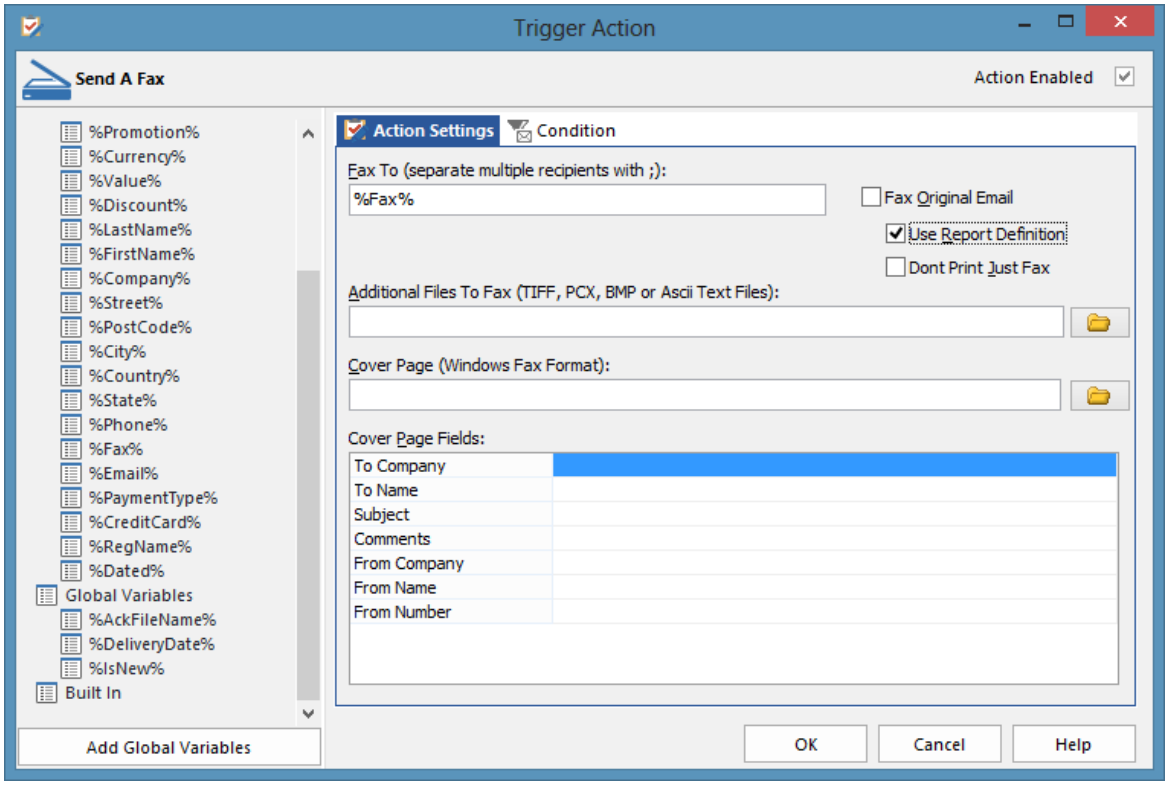

After you have drawn a control on to the designer, right-click it to edit it's properties.

The ADO, DAO, RDO controls allows you to include data from other sources in your report. See: Using External Data Sources

Click the **Preview** tab to view how the report will look when its printed.

#### 8.8.9.4 Send A Fax

Send a fax to multiple recipients and optionally include a cover page. Faxes are sent direct to the Fax Modem installed on the Email2DB computer. No other software is required.




Enter the **Fax To** number. You can send to multiple numbers by separating them with semi-colons or commas. As with all entries on this form you can use %fieldname% replacements.

Fax numbers do not need to be in any specific format. Email2DB will just dial the number as it's entered. If you need to dial 9 to access an outside line you must enter this (Email2DB doesn't make use of 'dialing rules').

Enable the **Fax Original Email** option if you want Email2DB to just fax the incoming email unformatted. The output is the same as if you printed the email from your mail client and then faxed the pages.

### Use Report Definition

This option enables you to create a report definition (via the Run Report Action) which can then be automatically faxed. You can use the report designer to create a professional looking document spanning multiple pages. You do not need to enable any of the report export options when doing this, since Email2DB will export the report to a fax document automatically if this option is enabled. By default Email2DB will also print reports created with the report designer. Enable the **Don't Print Just Fax** option if you just want to fax the report.

### Additional Files To Fax

In addition to reports, Email2DB can fax other documents. Supported formats are: TIFF (Black and white only), PCX, BMP and ASCII text files. Any additional files will be appended to pages sent via the report definition (if any).

### Cover Page

Email2DB faxes can also include a cover page. These are in Windows Fax format. A number of cover pages are included in the FaxCover folder in your C:\Program Files\Email2DB folder.

The Cover Page Fields grid allows you to enter fields which can be used on the cover page. Again, all

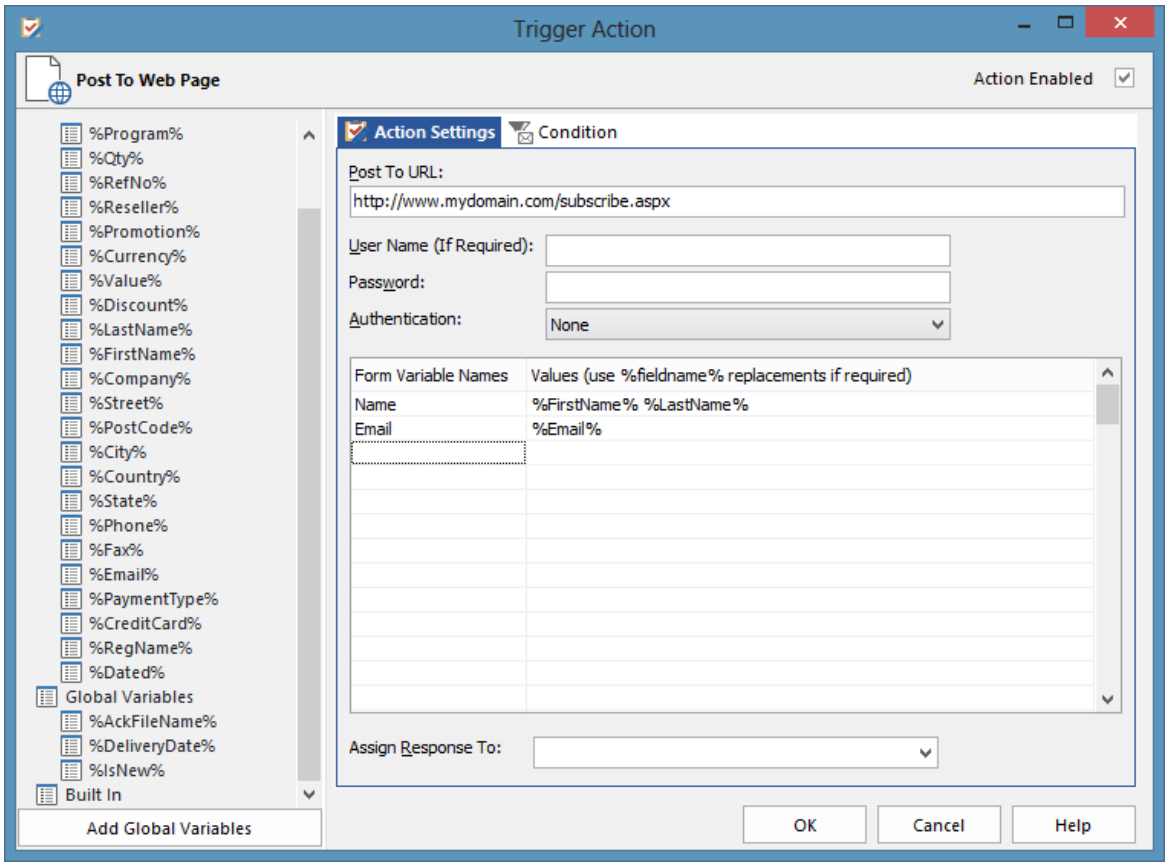
entries can make use of %fieldname% replacements.

*Note: Before sending faxes, you need to specific the COMM port that Email2DB should use. This is done in the main program options - Fax tab*

## 8.8.10 Web

### 8.8.10.1 Post To A Web Page

Post data to any web page using a HTTP POST. This is the same as if someone had filled in a web form and then submitted it to a page.




Specify the **Post To URL** of the web page to post to.

If your web page requires you to login prior to posting, select the **Authentication** method and enter the **User Name** & **Password**.

You can specify any number of **Form Variable Names** and **Values**. The Values can be fixed or % fieldname% replacements or combinations of both.

The format of the body for this action is application/x-www-form-urlencoded. The action resembles the traditional post action from an HTML <form> element. The body of the post will look as follows when using the values present in the above screen shot:

name=%FirstName%&email=%Email%

*Notes: The URL specified should be the page that RECEIVES the post. This will be different than your web form page. For instance, if you have a feedback form on a page called 'feedback.htm' then POST to value for this form will be something like 'receivefeedback.asp' or 'receivefeedback.php' - it is THIS page that you post to, not 'feedback.htm '.*

### Posting To .NET Web Services

You can use this option to call a .NET web service. You need to set the URL to the ASMX file along with the method to call, eg:

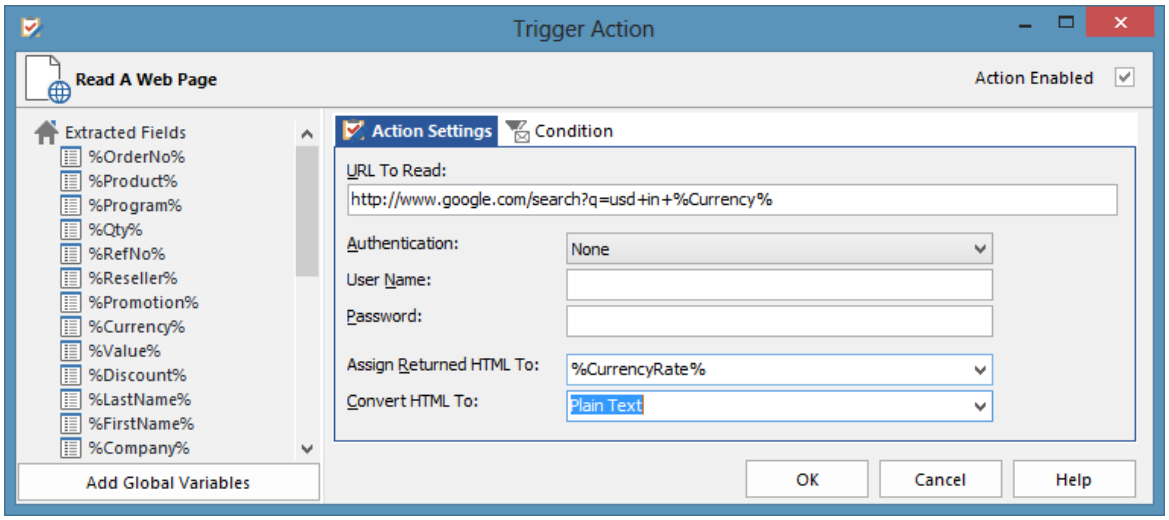
```
http://www.mysite.com/mywebsevices/webservice.asmx/Subscribe
```

By default .NET Web services will not allow POST's to them. You can enable the HTTP POST option in the protocols section of the web.config file of the web service.

See Also: Calling A Web Service

### 8.8.10.2 Read A Web Page

Reads any web page and assigns the returned HTML or plain text to an Email2DB Field or Variable.




Use this Action to read any web page. Specify the page to read in the **URL To Read** Entry (including any QueryString).

If the web page being read requires you to login first then specify the **Authentication** method, **User Name** and **Password**.

The returned HTML must then be assigned to an Email2DB Field or Variable. Select from the **Assign Returned HTML To** drop down.

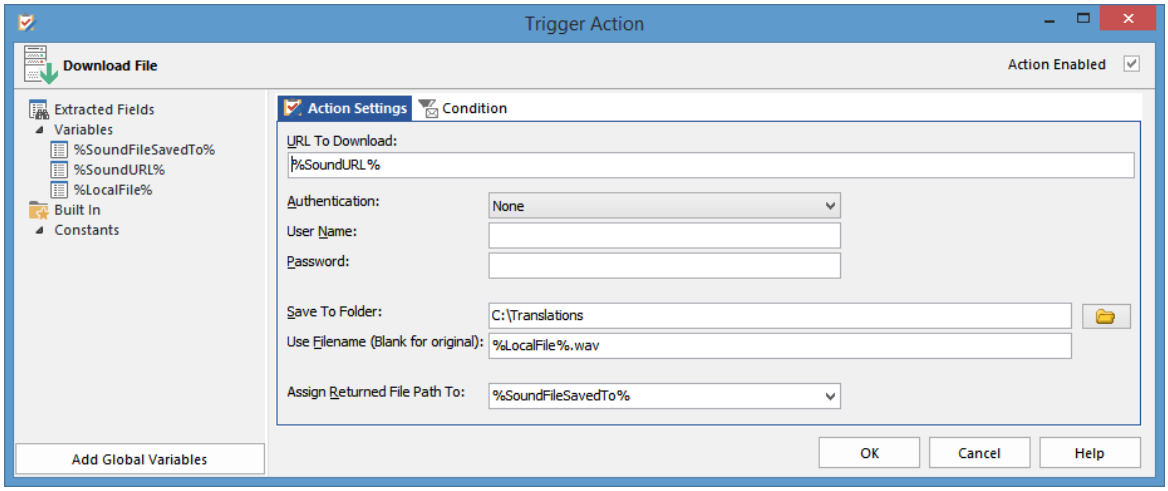
The **Convert HTML To** option enables you to convert the returned HTML to Plain Text or XML before it is assigned to the variable.

You can then make use of the HTML (or converted Plain Text or XML) in subsequent Actions.

See Also: HTML To XML Conversion

### 8.8.10.3 Download File

Download a file via HTTP or HTTPS from any given URL.




Enter the **URL To Download**. This can be any HTTP or HTTPS URL that points to a downloadable file.

If the web page being read requires you to login first then specify the **Authentication** method, **User Name** and **Password**.

Enter or select the Save To Folder. This is the folder on your file system that you want the downloaded file to be saved in.

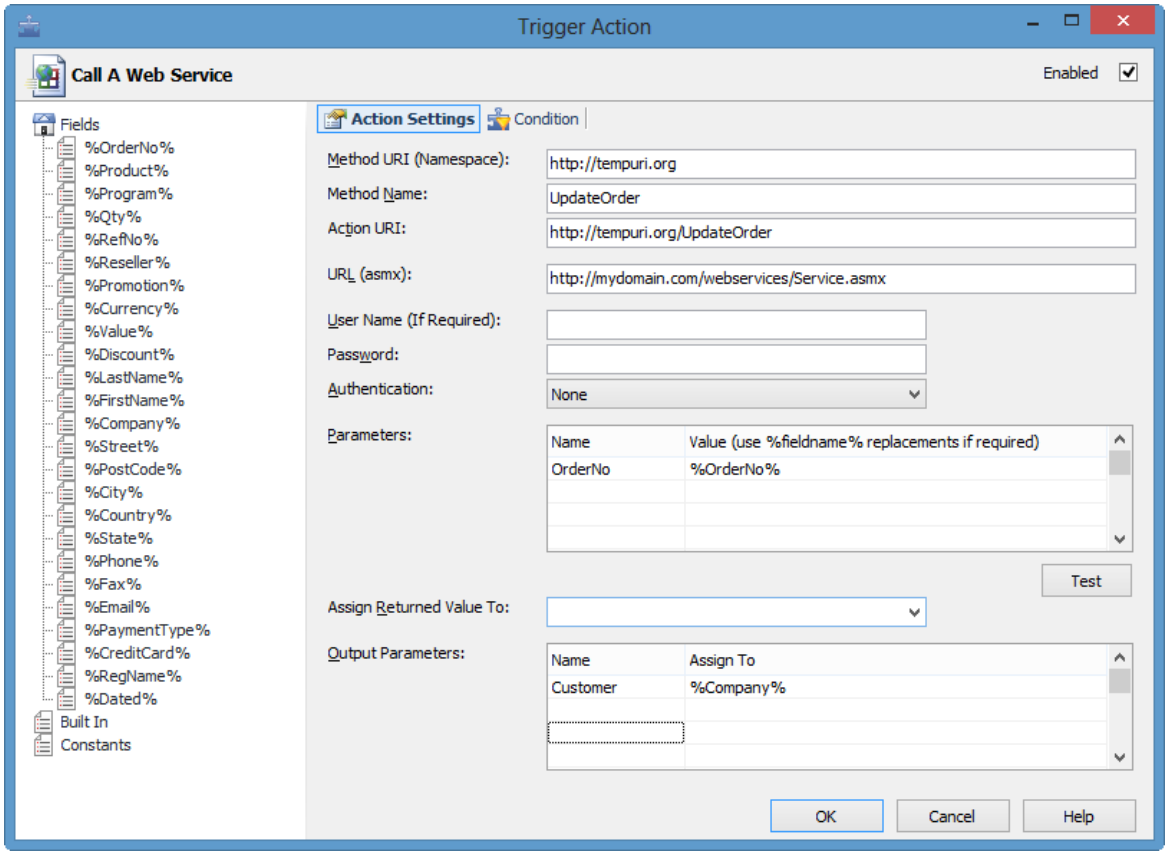
Optionally enter a **Use Filename**. If no filename is specified then the filename will be extracted from the URL.

Once the file is downloaded the resulting full path/filename will be assigned to the variable selected from the **Assign Returned File Path To** dropdown.

You can then use this variable to perform other actions such as adding the file as an attachment to outgoing emails.

#### 8.8.10.4 Call A Web Service

Executes a SOAP or .NET Web Service and returns the results to Email2DB fields or variables.




**Method URI (Namespace)**

This is the namespace of the web service. You can find the namespace by viewing the Web Service Definition (WSDL). For .NET Web Services view the .asmx page and click the Service Description link. The namespace will be shown in the targetNamespace element. .NET Web Services have a default namespace of http://tempuri.org. You should change this before making your web services public.

**Method Name**

This is the name of the method you want to call. You can view a list of available methods provided by the web service by viewing the asmx file.

**Action URI**

This defaults to the Method URI/Method name.

**URL (asmx)**

Enter the full public URL to the asmx page. You can use a secure address (https://) if required.

If your web services require a login before being accessed enter the **User Name** & **Password** and the **Authentication** type.

You must specify a value for all of the **Parameters** that the web service method expects. Enter the parameter **Name** and **Value**. For the value you can use %field replacements%.

**Assign Returned Value To**

You can assign the returned value to an Email2DB field or variable. Select the field or variable from the drop down list. If the web service returns a single value then this will be assigned to the field/

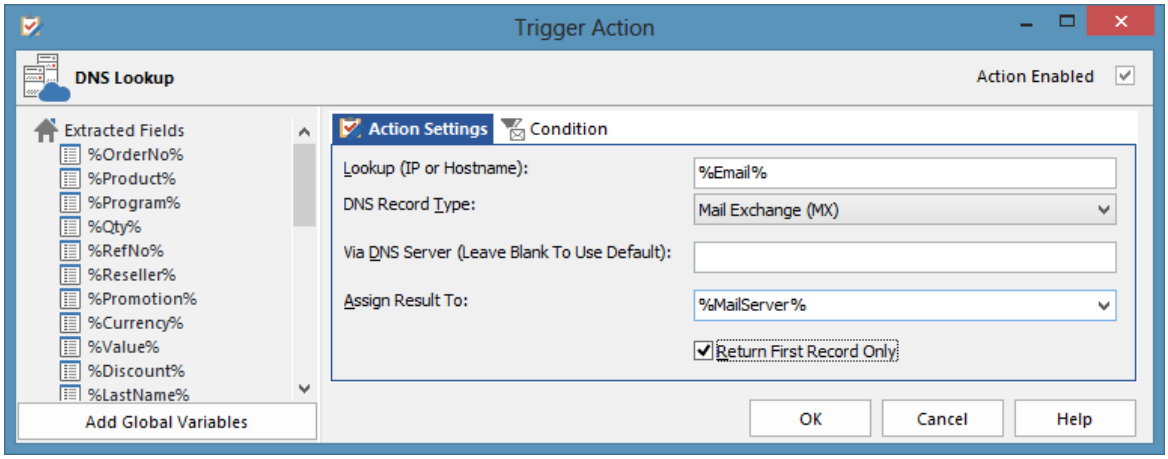
variable. If the web service returns a complex data type - such as a DataSet - then then entire XML response will be assigned.

**Output Parameters**

If the web service returns output parameters then you can assign individual parameter values to Email2DB fields or variables. If a parameter name specified is not returned as an output parameter then Email2DB will scan the returned XML and extract the specified parameter name as a tag and assign to the tag value to the Email2DB field/variable.

### 8.8.10.5 DNS Lookup

Performs a DNS Lookup and assigns the returned data to an Email2DB Field or Variable.




In the **Lookup** entry enter an IP Address or Hostname. If an email address is used Email2DB will automatically use just the domain name (the section after the @ sign).

Select the **DNS Record Type** to lookup. Here you can select any valid DNS type. Currently these are:

- Address (A Record)
- Name Server (NS)
- Mail Destination (MD)
- Mail Forwarder (MF)
- Canonical Name (CName)
- Start Of Authority (SOA)
- Mailbox (MB)
- Mail Group(MG)
- Mailbox Rename (MR)
- NULL (NULL)
- Well Known Service (WKS)
- Pointer (PTR)
- Host Information (HINFO)
- Mail Information (MINFO)
- Mail Exchange (MX)
- Text (TXT)
- Responsible Person (RP)
- Service (AFSDB)
- X25 (X25)
- ISDN (ISDN)
- Route Through (RT)
- AAAA (AAAA)

- Naming Authority (NAPTR)
- Reverse DNS

The **Via DNS Server** entry allows you to specify a specific DNS server IP address to use for the lookup. Leave blank to use the default.

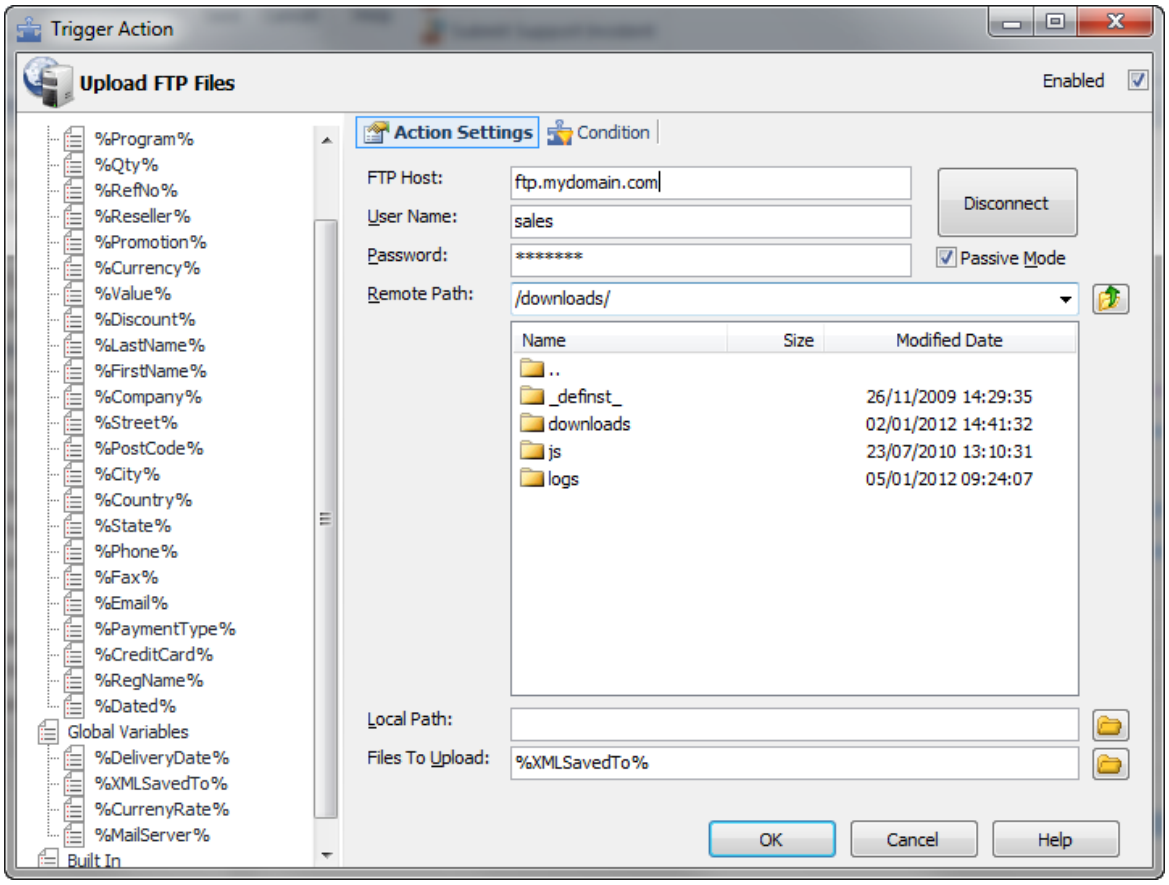
You must then select the Email2DB Field or Variable to assign the result to by selecting from the **Assign Result To** drop down.

### 8.8.10.6 Update WebDav Resource

Enter topic text here.

### 8.8.10.7 Upload To An FTP Server

Uploads saved attachments or other files to an FTP Server.




Enter your **FTP Host**, **User Name** & **Password**.

Click the **Connect** button to connect to your FTP Server. You may need to uncheck the **Passive Mode** option if your FTP server doesn't support passive mode.

You can then select a **Remote Path** to upload your files to.

The **Local Path** can be used to select a local directory to upload from. This would only be used if

you are not uploading Attachments - for example, you may want to upload the CSV or Excel file you have previously updated.

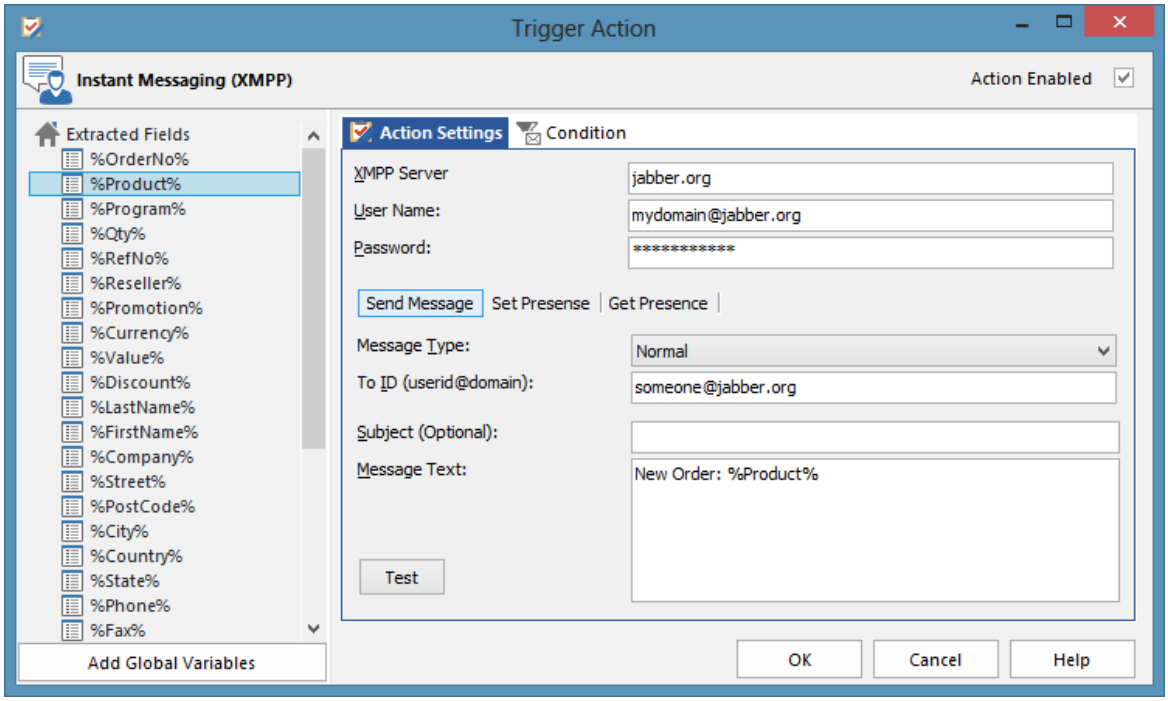
In the **Files To Upload** entry enter a list of file masks to upload separated by commas. You can use wildcards. If no Local Path is specified then Email2DB assumes you want to upload attachments. So if you entered *.pdf in the Files To Upload entry then all PDF attachments would be uploaded.

You can also use %field% replacements allowing you to upload specific attachments returned to a Variable via the **Save Attachments** action.

*Note: The Email2DB Message Processor will wait for the files to upload before continuing with the next message or Action.*

### 8.8.10.8 Instant Messaging (XMPP)

Posts a message and gets presence information against any XMPP (jabber) compatible instant messaging service. Most instant messaging services provide support for XMPP including Microsoft Windows Live, Google Talk, Jabber.org, Apple iChat, ICQ and many more.




Specify a **XMPP Server**. This should be the hostname or IP address of the instant messaging server you want to post to.

Enter your **User Name** & **Password** for your account on the instant messaging server that you want to post from.

Select the **Send Message** tab to send a message:

The **Message Type** can be:
- Normal - this would be the default in most instances. The message is simply sent to the recipient.
- Chat - a message sent in the context of a 2-way chat between 2 entities.
- Group Chat - a message is sent to multiple entities.
- Headline - a message that represents one of a list of items.

You must specify the **To Jabber ID**. This will be the full user id of the user you want to send the message to. On some instant messaging services this user must already be a contact of yours.

Enter the **Subject** and **Message Text**.

When the Action is executed the message is sent instantly to the recipient.

Select the **Set Presence** tab to change the online status for the user specified in the User Name/ Password settings.

### Reading XMPP Presence Information

Select the **Get Presence** tab to read the presence information for all or some of the users on the specified users buddy list.

Enable the **Get Presence** option. Enter the user **ID(s)** or leave blank to get the presence information for all users that appear on the given users buddy list. Select the variable to **Assign Presence To**.

Presence information is returned as follows:

Each user appears in a separate block:

```
-----------------------
Connected To Server:    myserver.cloudapp.net
Connected Using User:   Stephen
Date:                   2013-10-23 12:12:12
Client ID:              122
Domain:                 pslxmpp
Resource:               SPDEV
Name:                   Stephen
Status:                 1
Status Text:            Online
-----------------------
```

You can pass each user block to another Trigger for further processing. Select the Trigger from the **Pass Each Presence Message To Trigger** dropdown. This Trigger could then parse each user status to perform additional actions (such as recording presence information in a database).

#### 8.8.10.9 Send TCP Data

Sends data to any TCP Socket and optionally assigns the response to a variable.

## 8.8.11 Social Networks

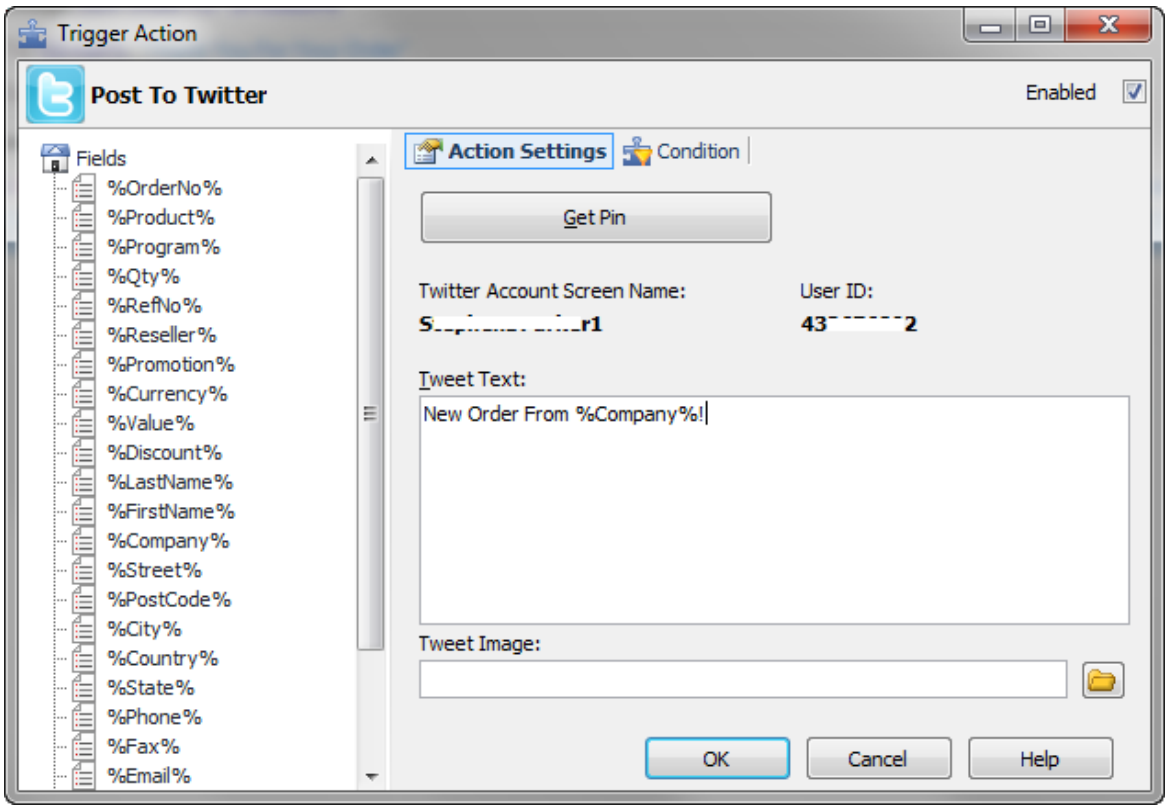
### 8.8.11.1 Post To Facebook

Creates a new status update on Facebook for the specified user.

Not Yet Implemented.

### 8.8.11.2 Post To Twitter

Posts a new item to Twitter for the specified Twitter user.



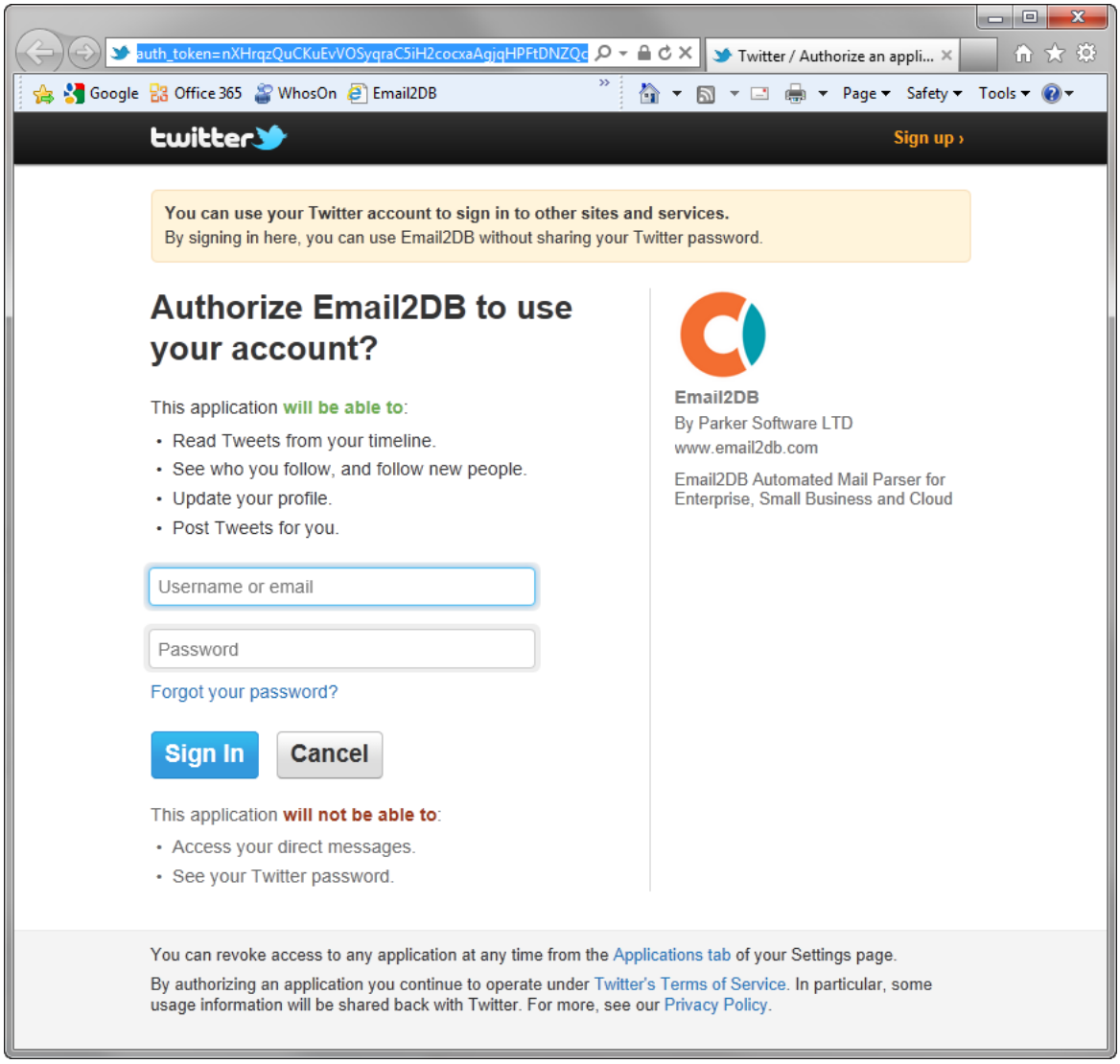

Enter the **Tweet Text** and an optional **Tweet Image**.

when this Action is run, Email2DB will post to the Twitter feed of the specified Twitter Account Screen Name. You must authenticate to Twitter to specify the screen name to use. Click the **Get Pin** button to start the authentication process. Email2DB uses the secure OAuth mechanism to authenticate with Twitter. This ensures your Twitter password is never stored in the Email2DB Metadata.

#### Authenticating With Twitter

Click the **Get Pin** button to start the authentication process.

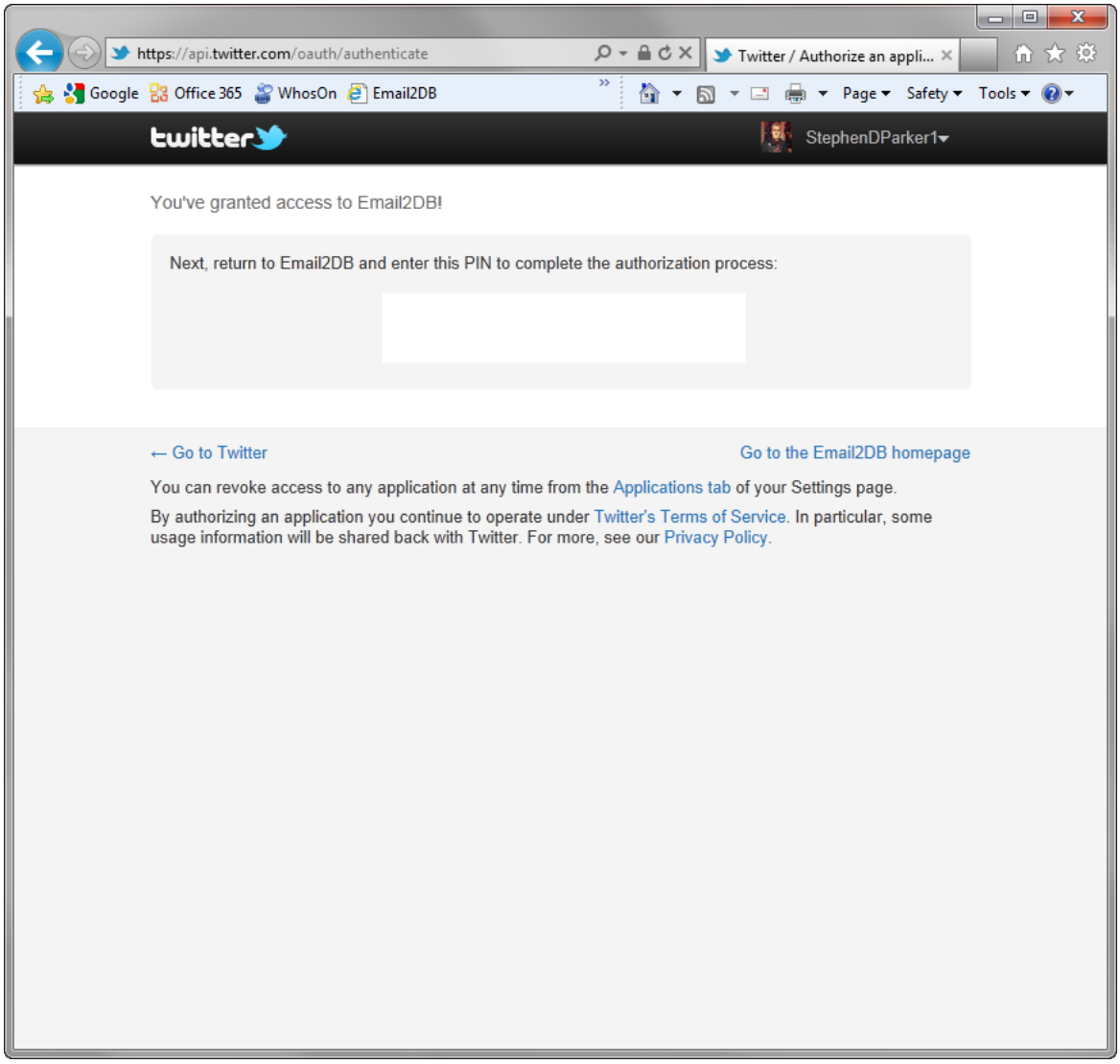
A Twitter login page will then be launched in your web browser:



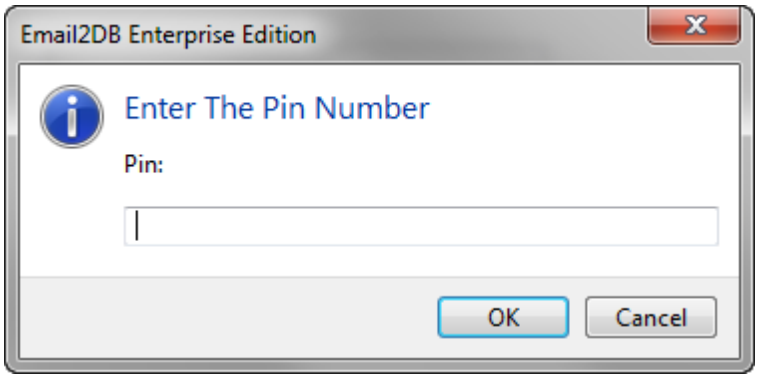

Login with your Twitter **username** & **password**. *Note: This information is NOT shared with Email2DB.*

By logging in you are authorizing the Email2DB Application to post tweets to your feed.

Once you have signed in a Pin number will be displayed:




Enter the Pin number in the Email2DB prompt:



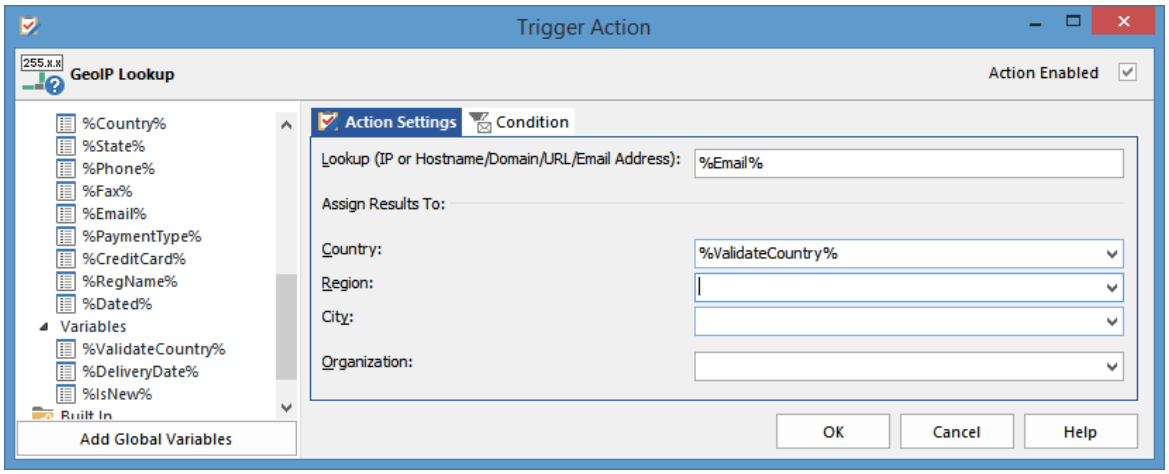

### 8.8.11.3 Post To LinkedIn

Post a Share update to LinkedIn.

## 8.8.12 Language/Location

### 8.8.12.1 GeoIP Lookup

Performs a Geo IP lookup for any IP address, domain name, email address or URL. The Geo IP lookup returns the City, Region & Country for any IP address. Also returns the Organization name.



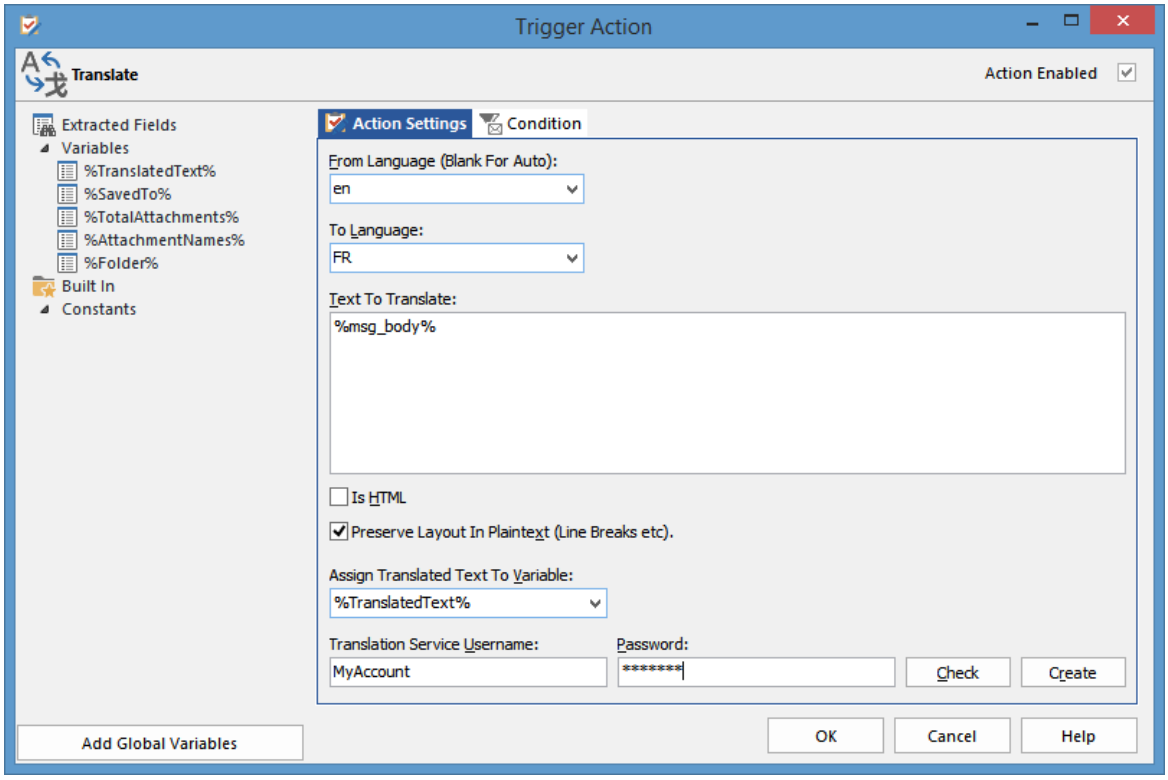

In the **Lookup** field enter any field, variable or constant. If the field contains an IP address then this will be used. If no IP address is found then the first email address will be used. If no email address then the first URL. If an email address/domain name is used then Email2DB will first perform a reverse DNS lookup to find the IP address and then use that to do the Geo IP lookup.

You can then assign the **Country**, **Region**, **City** & **Organization** to fields or variables.

This action is useful for validating customer details - for instance you can check that the GeoIP Country of the customers email address matches the country they specify on an order form.

### 8.8.12.2 Translate Text

Translates text from one language to another and returns the result to a field or variable.




Enter or select the **From Language**. If blank the language will be automatically detected (if possible).

Enter or select the **To Language**. This is required.

Enter the **Text To Translate**. This is limited to 10000 characters.

Enable **Is HTML** if the text will contain HTML tags. Passing the built-in field %msg_html% would pass the entire HTML of the incoming message.

Enable **Preserve Layout In Plaintext** if you are passing none HTML (for example, the %msg_body % field contains the plain-text version of the incoming message). If this option is enabled then any carriage return/line feed characters and any white space will be preserved. Otherwise all control characters and white space is removed from the text before translation.

Select the variable you want the translated text assigned to from the **Assign Translated Text To Variable** dropdown.

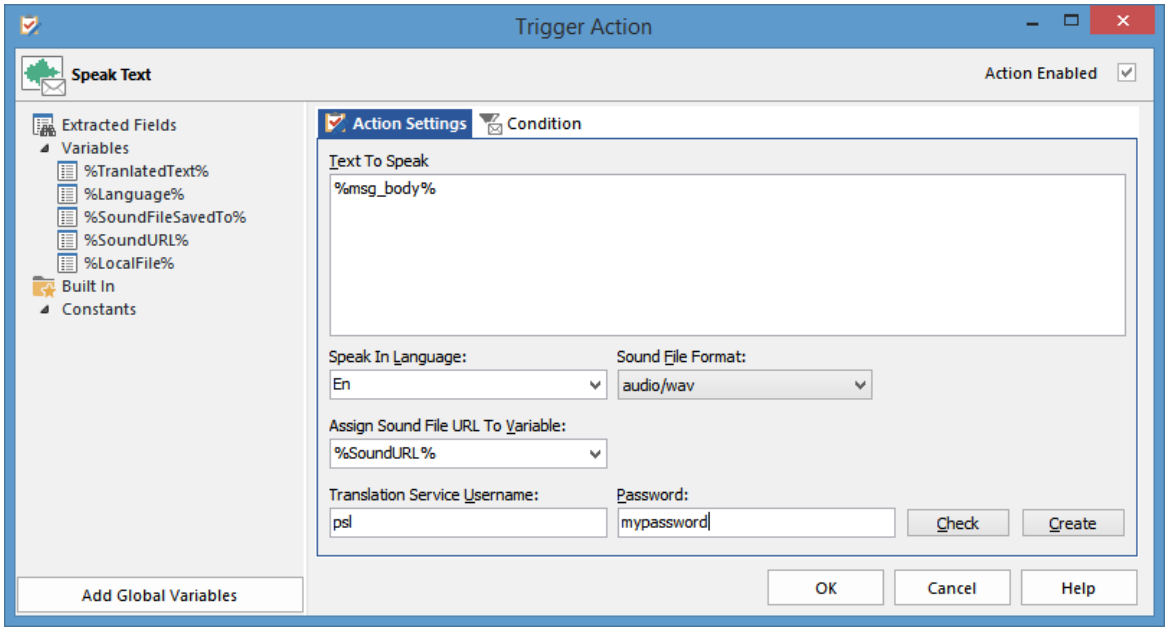
Enter your Parker Software **Translation Service Username** & **Password**. Click **Check** to verify your account. Click **Create** to create a new account.

See: Supported Languages

*Note: You need to setup a Translation Services account before you can use the Translate, Speak & Detect Actions.*

### 8.8.12.3 Speak Text

Returns a URL of a WAV or MP3 file containing spoken text in the desired language.




Enter the **Text To Speak**. This is limited to 2000 characters.

Enter or select the **Speak In Language**. This can be the same as the input language if you just want to speak the text as is.

Select the **Sound File Format** (WAV or MP3).

Select the variable you want the URL to be assigned to from the **Assign Sound File URL To Variable** dropdown.

You can use the Download Action if you want to download the file to your system.

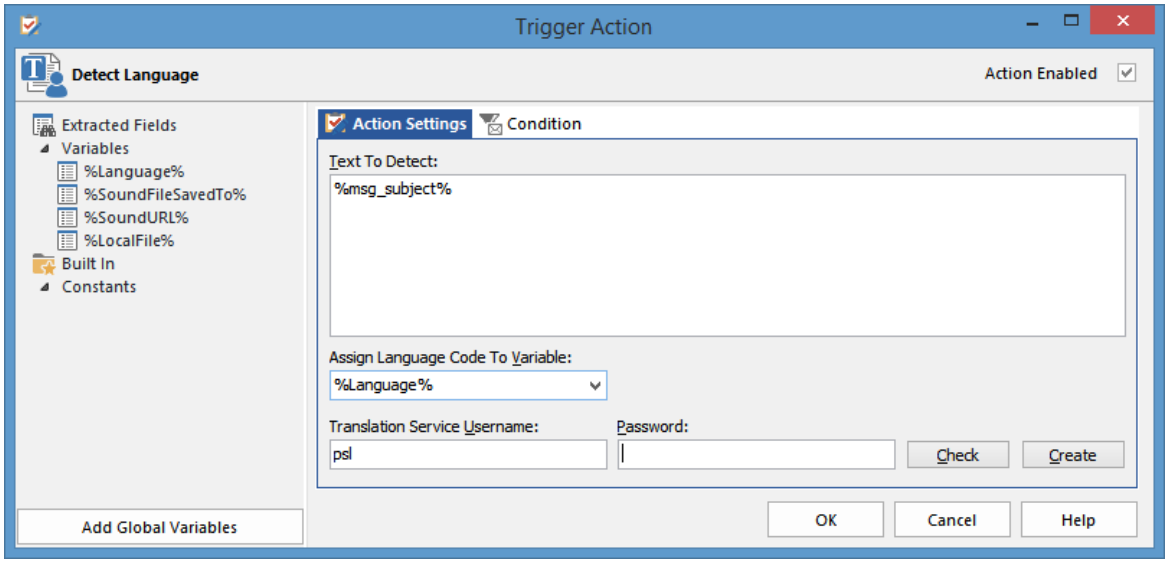
Enter your Parker Software **Translation Service Username** & **Password**. Click **Check** to verify your account. Click **Create** to create a new account.

See: Supported Languages

*Note: You need to setup a Translation Services account before you can use the Translate, Speak & Detect Actions.*

### 8.8.12.4 Detect Language

Detects the language for any text and returns the language code to an Email2DB Field or Variable.




Enter the **Text To Detect**. This field is limited to 10000 characters.

Select the variable you want the language code assigned to from the **Assign Language Code To Variable** dropdown.

Enter your Parker Software **Translation Service Username** & **Password**. Click **Check** to verify your account. Click **Create** to create a new account.

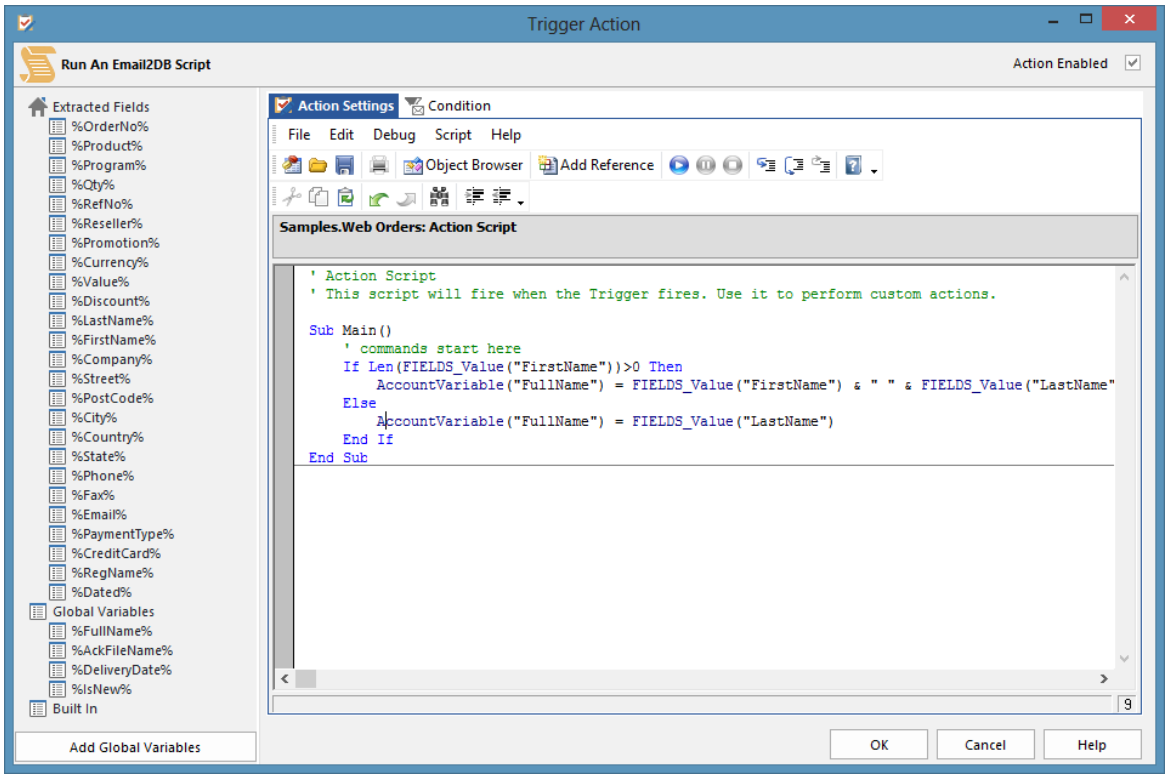
See: Supported Languages

*Note: You need to setup a Translation Services account before you can use the Translate, Speak & Detect Actions.*

## 8.8.13 Scripting/System

### 8.8.13.1 Run An Email2DB Script

Runs an Email2DB Script. You can create Visual Basic compatible scripts to perform custom actions. Scripts can be as big as you need and contain multiple Sub-Procedures and Functions.




Write your basic script directly in the code window. Each script must have a **Sub Main** procedure. You can use the **Add Reference** button to use external COM objects (like ADO).

Within the script you can access properties about the current message. Click the **Object Browser** button to view a list of available properties. For example, the property **MSG_Body** returns the complete message text. **MSG_Header()** and **MSG_HeaderValue()** are arrays containing all headers and header values. **MSG_HeaderCount** returns the number of headers.

Extracted fields can be referenced using **FIELDS_Value('fieldname')**. You can read and write to any Extracted Field.

Within your triggers you can set account specific variables and these are accessible only against the account they occupy, these are known as Account Variables and can be referenced using **AccountVariable('variablename')** with scripted commands. Advantages to using Account Variables are that they only apply to the account in question and its triggers. This is also the way that you access variables that have been declared using **Set Variable** action.

Variables can be referenced using **GlobalVariable('variablename')**. You can read and write to any Variable.

You can drag and drop Email2DB Extracted Fields or Variables onto the code editor. When dropping fields they will be automatically changed to FIELDS_Value('fieldname') which is the format required for referencing fields in scripts, or GlobalVariable('variablename') for Variables.

Within the code editor you can access on-line help for all basic functions. Help is context sensitive.

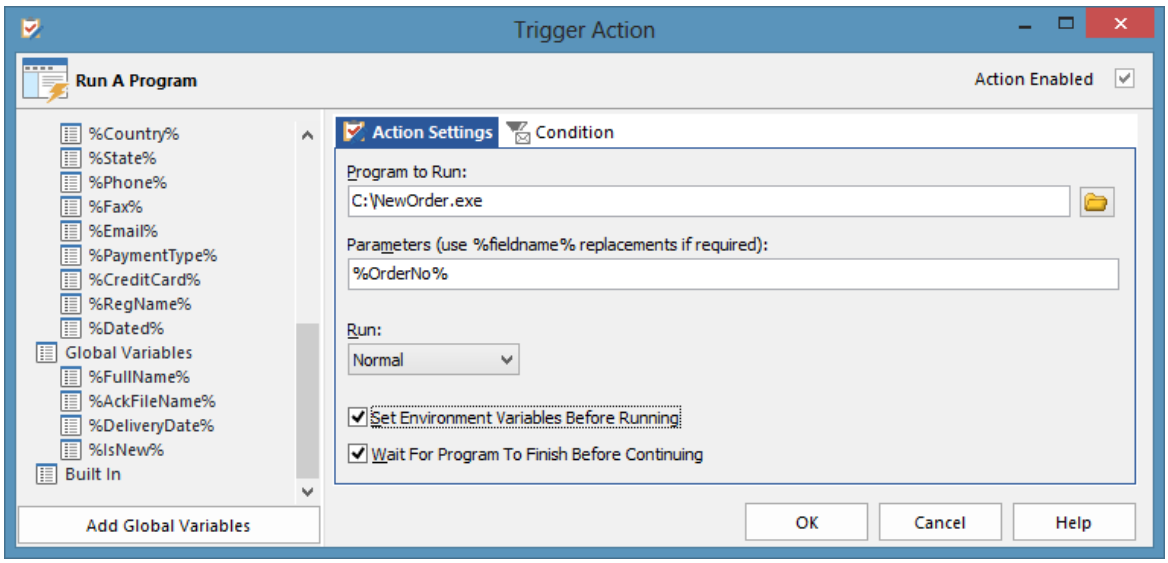
Move your cursor to the beginning of the keyword and press **F1**.

See Also: Scripting Introduction
See Also: Properties & Method Available In Email2DB Scripts
See Also: Processing Attachments Using Scripts

** The Email2DB Cloud Edition can run scripts but they must be enabled by the server administrator. If you are using the Email2DB Cloud Edition you will need to contact the support desk to enable the script each time you make a change. The support desk will verify the script to ensure it will not have a negative effect on server performance before enabling it.*

### 8.8.13.2 Run A Program

Execute any Windows executable file and optionally pass it parameters.




Enter the **Program To Run** and any **Parameters**. You can use %fieldname% replacements in both entries.

Check the **Set Environment Variables Before Running** option if you want to Email2DB to set the environment variables to Extracted Field values before the process is run. This allows you to access field values in your process.

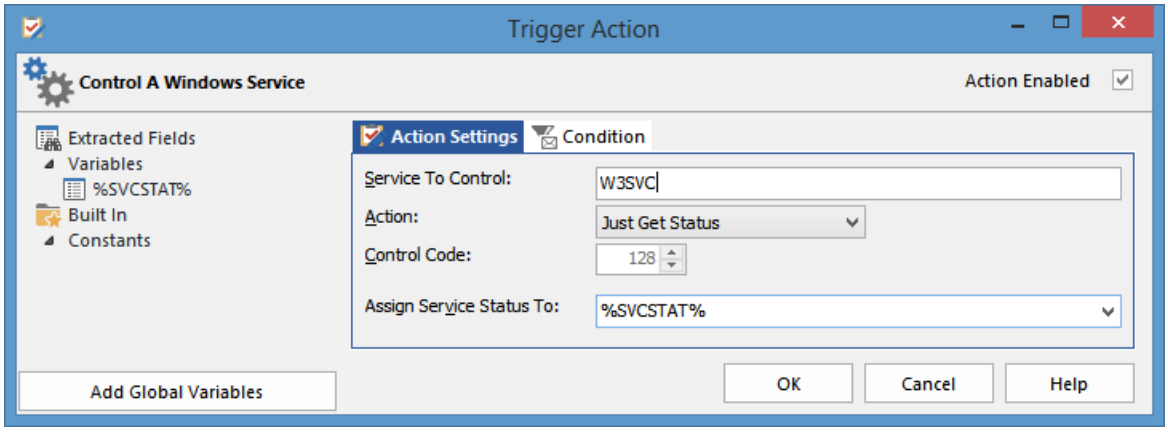
Check the **Wait For Program To Finish Before Continuing** option if you want Email2DB to stop executing further actions until the process completes. Ensure your process closes itself otherwise this could cause Email2DB to hang.

The executable being run must not have a visible user interface. This is because it will be executed under the context of the Email2DB Message Processor Service.

*Note: The Email2DB Message Processor Service must have sufficient rights to launch the executable file. You may need to run the service under a different user than the default SYSTEM account to use this Action.*

### 8.8.13.3 Control A Service

Control any Windows Service.




Enter the **Service To Control**. You can use multiple services separated by semi-colons. You can also enter %fieldname% replacements. Allowing you to pass the service name via email.

You can then select the Service control **Action**: Start, Stop, Pause, Continue, Restart the service, Send Control Code & Just Get Status.

The service status can be returned to a variable in the **Assign Service Status To** list, after the selected action. This will be: Running, Stopped, Paused or Not Installed.

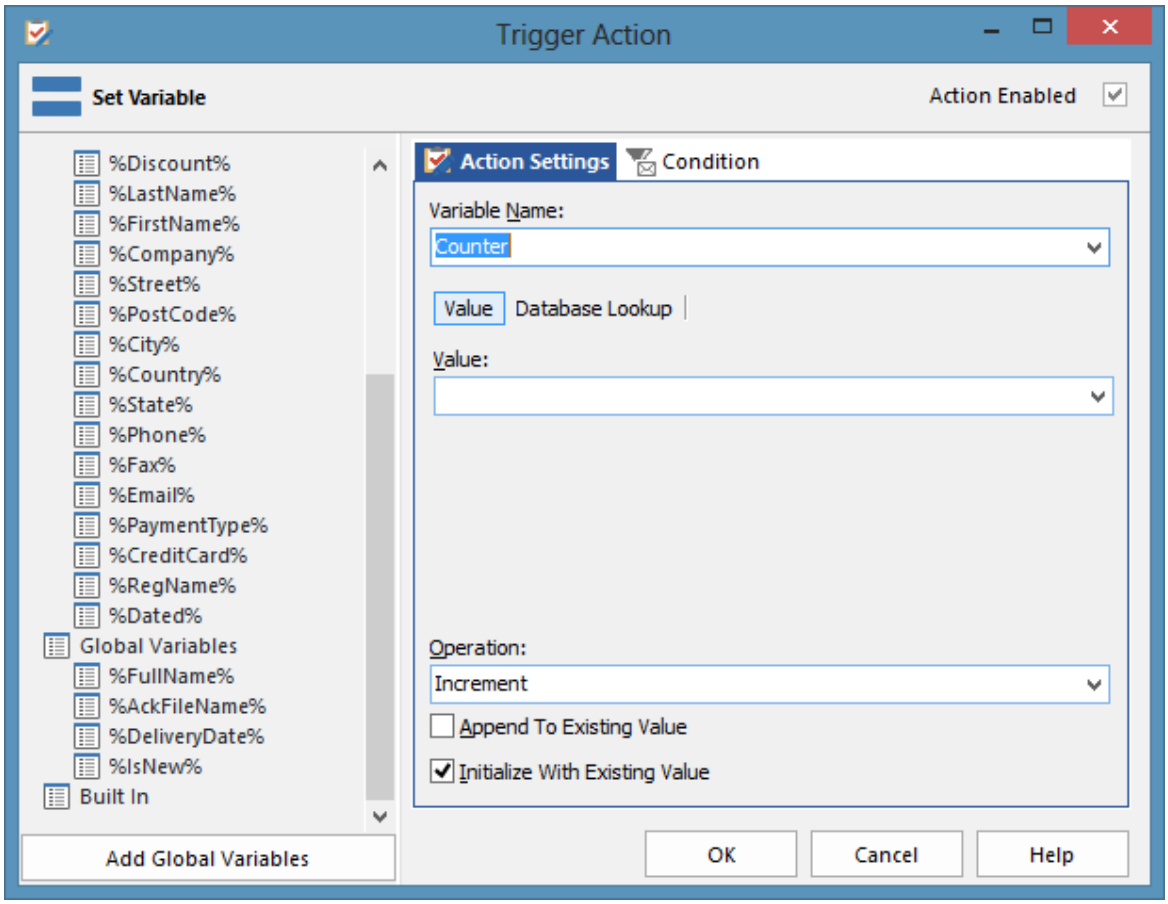
You could then create conditional actions based on this value - for example: Send an email if a service is not running.

*Note: The Email2DB Message Processor Service must have sufficient rights to control the service. You may need to run the service under a different user than the default SYSTEM account to use this Action.*

## 8.8.14 File Operations, Set Variable, Misc

### 8.8.14.1 Set Variable

Creates and/or sets an Email2DB Global Variable.




In addition to Extracted Fields you can also make use of 'Variables' in your Email2DB Actions.

A Variable is simply a place holder for a specific value. Variables maintain their values between each message processed - whereas Extracted Fields are reset on each new message.

Some other Email2DB Actions (such as Save Attachments, Read A Web Page etc) can return values which can be assigned to Variables. In these cases you must create the Variable first by simply dragging into the Actions list and giving it a name BEFORE you want use it in other Actions.

Each Variable must be given a **Variable Name**.

You can optionally assign it a **Value**. The Value can be a fixed value or the value of another Extracted Field or Variable (using %field% replacements). See: Using Field Replacements

The **Operation** option allows you to perform an optional additional operation on the Value before it's assigned to the Variable.

The available Operations are:

| | |
|---|---|
| Increment | Adds 1 to the current value of the Variable. |
| Decrement | Subtracts 1 from the current value of the Variable. |
| Add To | Adds the value specified in the Value property to the current value of the Variable. |

| | |
|---|---|
| Subtract From | Subtracts the value specified in the Value property from the current value of the Variable. |
| Add Days To | If the existing value is a date then adds the days specified in the value to the existing date value. |
| Subtract Days From | If the existing value is a date then subtracts the days specified in the value to the existing date value. |
| Convert To Upper Case | Converts the Value to UPPERCASE. |
| Convert To Lower Case | Converts the Value to lowercase. |
| Trim Blanks | Removes leading and trailing blanks from the Value. |
| Remove All Whitespace | Trims and removes all whitespace (multiple spaces, tabs, line feeds, carriage returns) are converted to single spaces. |
| Remove First & Last Characters | Trims the first and last characters (useful for removing quotes or brackets from a string). |
| Add Space Character To End | Adds a space character to the end of the Value if a space does not already exist (useful when concatenating names). |
| Word Capitalize | Converts to lowercase then Capitalizes each word in the Value. |
| Convert HTML To Plaintext | If the Value is HTML then all tags are removed. |
| Convert HTML To XML | If the Value is HTML then the text is converted to well-formed XML. |
| Extract First Email Address | The Variable is set to the first email address found in the Value. |
| Extract All Email Addresses | Extracts all email addresses found in Value. Returns comma separated list. |
| Extract First URL | Extracts the first URL (web address) found in the Value. |
| Extract All URL's | Extracts all URL's found in Value. Returns comma separated list. |
| Encrypt | Encrypts the Value using 128 bit AES encryption. The resulting value is URL encoded. |
| Decrypt | Decrypts the Value using 128 bit AES encryption using the same key as the Encrypt operation. |
| URL Encode | URL Encodes the Value. |
| URL Decode | URL Decodes the Value. |
| Base 64 Encode | Base 64 Encodes the Value. |
| Base 64 Decode | Base 64 Decodes the Value. |
| Quoted Printable Encode | Quoted Printable Encodes the Value. |
| Quoted Printable Decode | Quoted Printable Decodes the Value. |
| Create MD5 Hash | Creates an MD5 Hash of the Value which is then URL encoded. |
| Create GUID | Generates a Globally Unique Identifier. |

If the **Append To Existing Value** option is enabled then the Value will be appended to the existing value for the Variable. Care should be taken using this if the Variable is persisted and is not cleared at some point - to avoid ending up with very big strings.

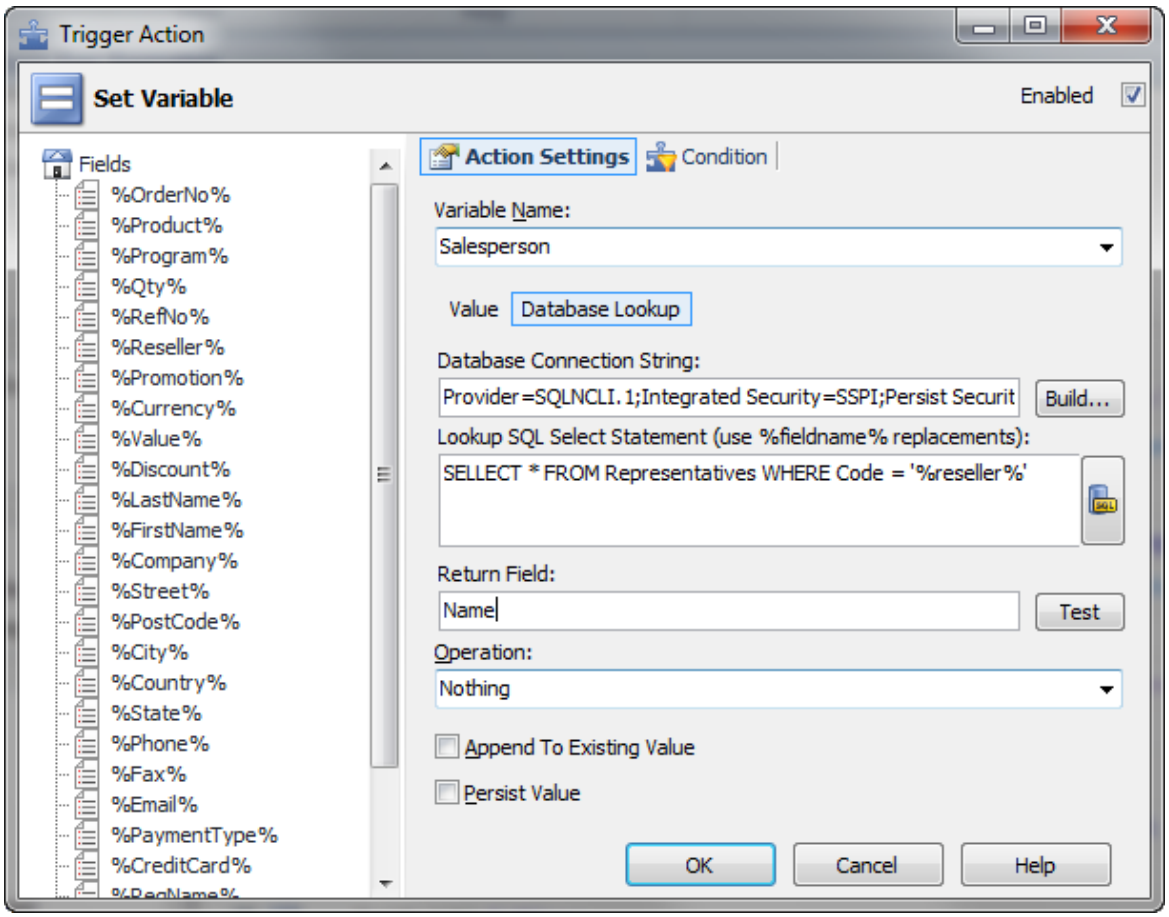
If the **Persist Value** option is enabled the the Variable value will be saved in the Registry each time it is set. The value will then be persisted even when the Email2DB Service is restarted. Email2DB places a limit on the size of persisted values to 2MB.

*Note: Email2DB sets a size limit 10MB for each global variable instance.*

## Database Lookup Variables

You can also assign it using a **Database Lookup**.

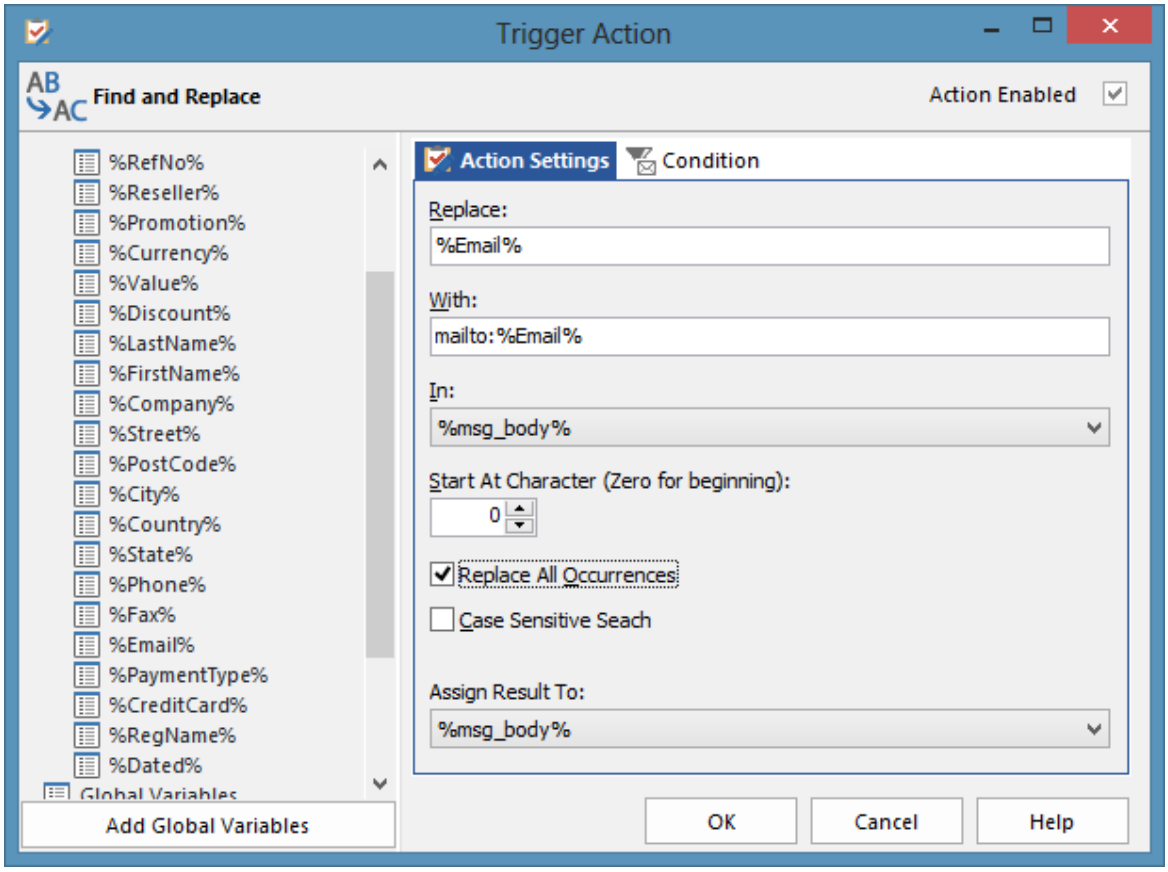
When using a Database Lookup you create a **Database Connection String** and **SQL Select** statement to lookup a value from an external database.




The **Return Field** is set to the field in the SELECT statement that you want to return the data for and assign to the Variable.

### 8.8.14.2 Find And Replace

Finds and replaces text in any Email2DB field or variable and returns the result to the same or a different field/variable.




Enter the text to find in the **Replace** entry. You can make use of %field% replacements here.

Enter the text to replace with in the **With** entry. Again you can use %field% replacements.

In the **In** dropdown - select the field to use for the replacement. This can be any of your extracted fields or any of the built-in fields.

By default the text search will start at the beginning of the text. Enter a **Start At Character** value if you want to start the search from a specific character position.

Enable **Replace All Occurrences** if you want all occurrences of the Replace text to be replaced.

Enable **Case Sensitive** Search if the find should be case sensitive.

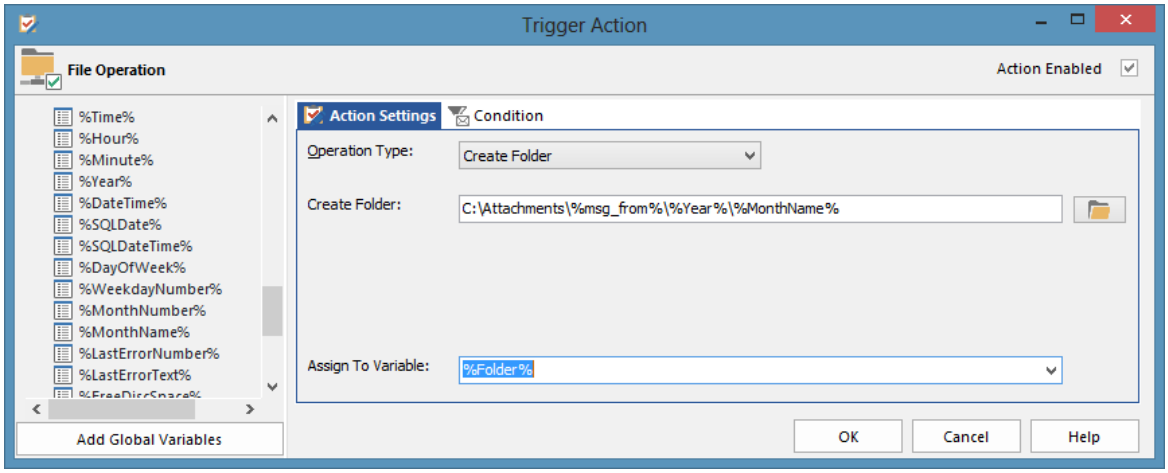
You must then select a field or variable to assign the replaced text to from the **Assign Result To** dropdown. By default the same field/variable that you select in the In selection will be used.

Subsequent actions that make use the the updated field/variable will now see the replaced value.

*Note: For the built-in fields: %msg_body%,%msg_html%,%msg_mimetext%,%msg_to%,%msg_from%,%msg_fromname%,%msg_cc%,%msg_bcc%,%msg_subject% the underlying mime text of the message will also be changed - resulting in the changed message being stored in the Email2DB Message Store.*

### 8.8.14.3 File Operations

Create new folders, copy, move, rename or delete files.




Use this Action to create new folders or to copy, move, rename & delete files.

Select the **Operation Type**.

Depending on the Operation Type enter the **Folder**, **File Name**, **To Folder** & To **File Name**.

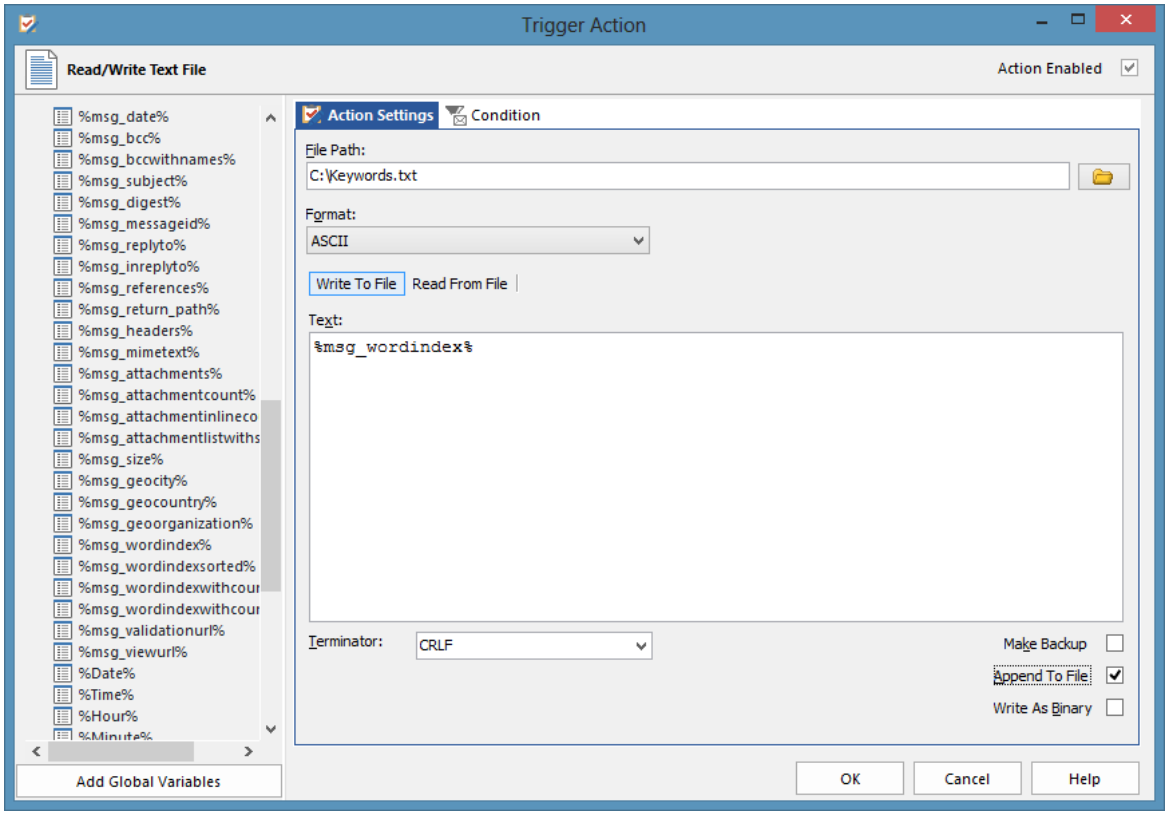
You can assign the result to a variable. Select from the **Assign To Variable** list. The variable will be set to the new folder and file name depending on the Operation Type. For example, for the Create Folder operation the variable will be assigned to the new folder name. For the Copy operation the variable will be assigned the full path & file name of the new file. If the operation fails the variable will be assigned a blank string and the error will be shown in the log.

For the Create Folder operation the complete folder structure will be created if any levels do not exist. If the full folder already exists the Assign to variable will be assigned the existing folder name and no error will be reported. An error will only be reported for the Create Folder operation if the creation of a new folder fails.

*Note: If you are access folder/files on your network you will need to change the user name that the Email2DB Service runs under. See: Running The Email2DB Service Under A Different User*

### 8.8.14.4 Read/Write Text File

Writes data to any text file and optionally reads the contents of the file into an Email2DB Field or Variable.




The **File Path** must contain a path & filename of the text file to write to/read from.

On the **Write To File** tab you can define the text that you want to write to the file. The file will be created if it does not already exist.

Select the file **Format**. This can be ASCII, Unicode or UTF8. Use Unicode if your data will contain any non-ASCII characters.

Select the **Terminator** that will be appended to the text written to the file.

Enter the **Text** that you want to write to the file.

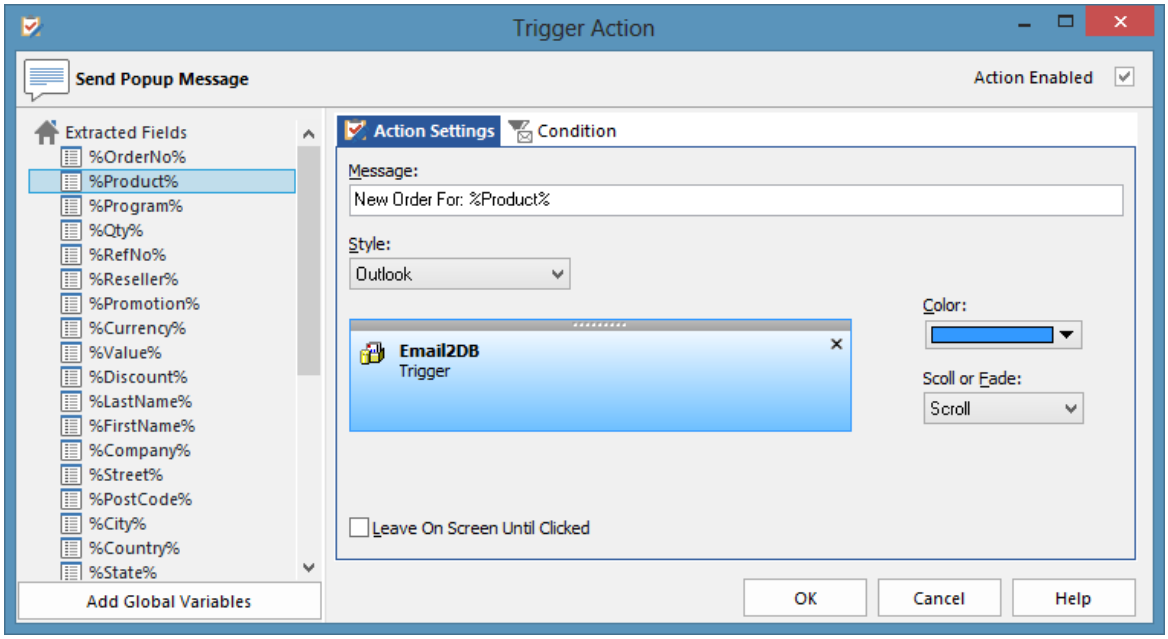
If the **Append To File** option is enabled then the new text will be added to the end of the existing text file (if it exists).

If the **Make Backup** option is enabled the existing file will be copied to the same file name with a .BAK file first.

On the **Read From File** tab you can select an Email2DB Field or Variable that you want to assign the text file contents to. The file will be read again after any text is written to it.

### 8.8.14.5 Send Popup Message

Sends a popup message to users running the Email2DB Administrator or Email2DB Client.




You can choose between MSN Messenger and Outlook style popups. You can also customize the **Message** text, **Color** & display modes of the popup.

Popups will only be sent to users who have access to the Account/Trigger.

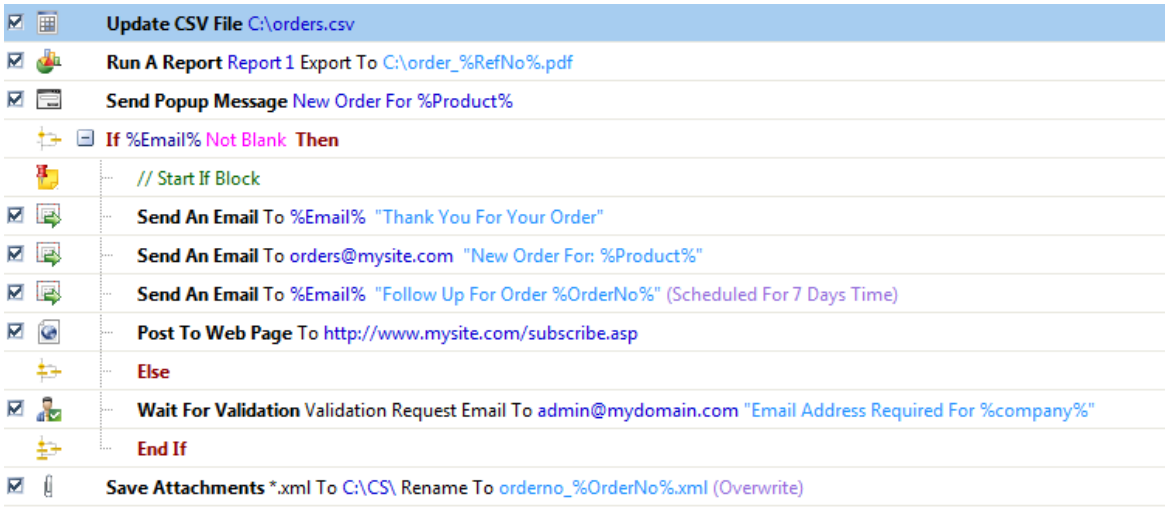
You can include %field% replacements in the Message. If you include the %msg_validationurl% for Validation Actions (or any other URL) then the user will be able to click the popup to launch their web browser to view the web page.

*Note: If many messages are processed within a short interval Email2DB will suspend the showing of popups to prevent the screen from completely filling and slowing down the users computer.*

## 8.8.15 Logical Operators

Logical Operations allows you to conditionally execute blocks of Actions.

**Example 1**:




The above Actions list has an **If** ... **Else** ... **End If** block.

The Actions between the **If** and **Else** statements will be executed if the Extracted Field %Email% is not blank, otherwise the Actions between the **Else** and **End If** statements will execute.

If blocks can just contain **If** .. **End If** - the Else is optional.

If blocks can be nested inside other If blocks.

**Example 2**:


The above example Saves any PDF attachments to C:\PDF. The resulting saved files are assigned to the Variable 'PDF'.

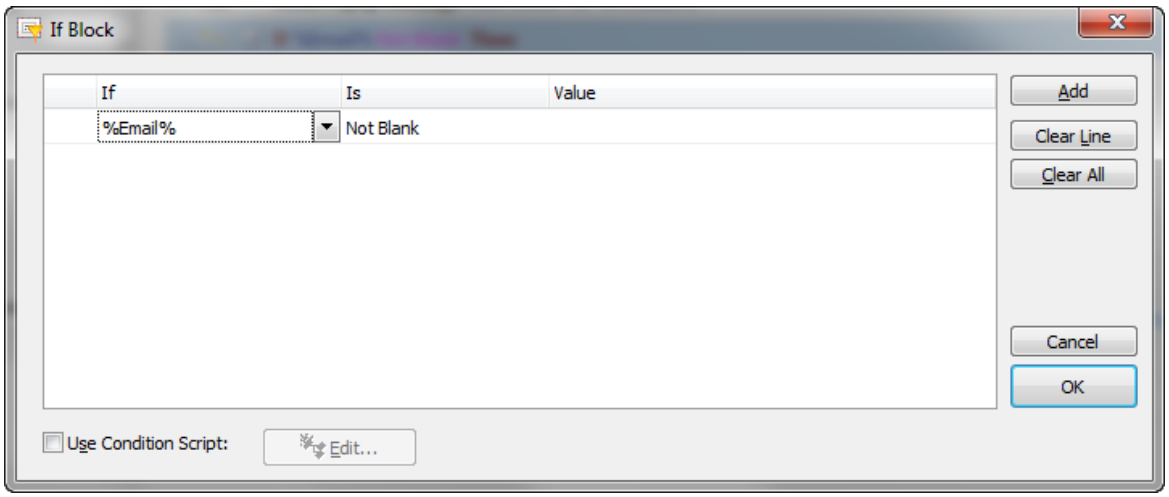
The If block Condition is set to If %PDF% Is Not Blank.

If a PDF attachment is saved then a pop-up message is sent, otherwise processing ends for the current message.

### 8.8.15.1 If.. Then.. Else .. End If

Conditionally executes a group of Actions based on a Condition.

When you drag an If Action onto the Actions list the **Condition Builder** will be displayed.




Here you create a Condition that will be tested. The Condition must be true to start the If block otherwise processing moves to the next Else or End If statements.

In the **If** column you select a message property or one of your Extracted Fields or Variables.

In the **Is** column you select one of the following:

- Equal To
- Not Equal To
- Less Than
- Greater Than
- Less Than Or Equal To
- Greater Than Or Equal To
- Is Blank
- Is Not Blank
- Contains
- Does Not Contain
- Starts With
- Length Equal
- Length Less Than
- Length Greater Than

In the **Value** column you can type a value to compare against or select one of your Extracted Fields or Variables.

Click the **Add** button to add another line. The new line can be assigned as an AND or OR clause.

Each **If** statement MUST have an **End If**.

If blocks can be nested inside other If Blocks.

**Example**:


The above example Saves any PDF attachments to C:\PDF. The resulting saved files are assigned to the Variable 'PDF'.

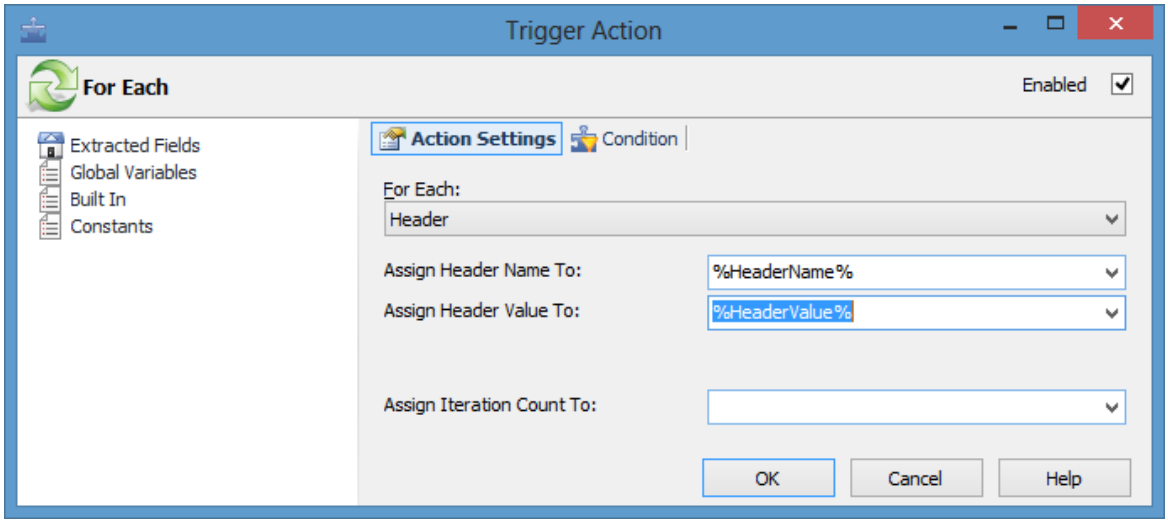
The If block Condition is set to If %PDF% Is Not Blank.

If a PDF attachment is saved then a pop-up message is sent, otherwise processing ends for the current message.

An **Else** statement can be dropped inside an If block:


### 8.8.15.2 For Each Loop .. Next Loop

Allows you to create a loop based on the following: Recipients, Attachments, Keywords, Extracted Fields, Message Headers.




The For Each action enables you to repeatedly execute a block of actions for each value selected in

the **For Each** drop down.

Select:

| | |
|---|---|
| Recipient(all) - | loop on all recipients for the current message (To, CC & BCC). |
| Recipient(To) - | loop on just recipients in the To address. |
| Recipient(CC) - | loop on just the recipients in the CC address. |
| Recipient(BCC) - | loop on just the recipients in the BCC address. |
| Attachment - | loop on each attachment. |
| Attachment(inline) - | loop on each inline attachment. |
| Keyword - | loop on each keyword extracted from the message body. |
| Header - | loop on each message header. |
| Extracted Field - | loop on each extracted field. |

Inside the loop you can assign the values of the current item to variables. You can then use these values in actions within the loop.

For example, when looping on Headers you can assign the Header Name & Header Value of the current header in the loop to variables by selecting from the Assign to drop downs.

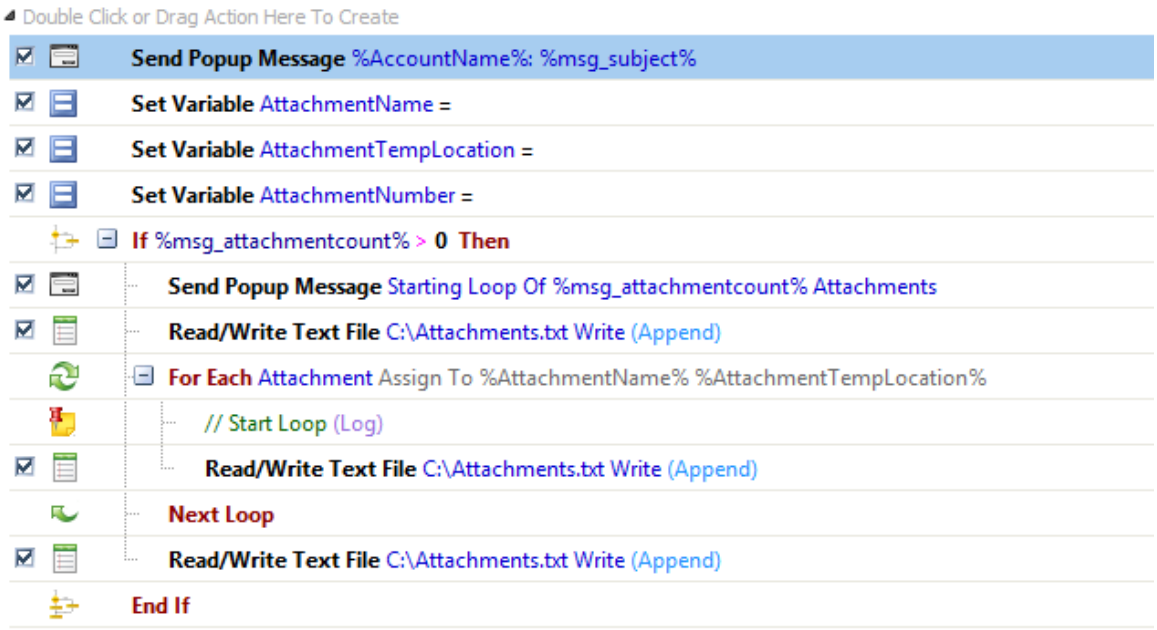
You then place Actions inside the For Each - Next Loop block.

## An Example

Suppose we want to update a text file when each email is received. The text file will be in the following format:

```
------------------------------------------------------------------------
From: %msg_from%
To: %msg_to%
%msg_subject%

Attachments:
1: %attachment%
2: %attachment%
{repeat}
- End Of Message
------------------------------------------------------------------------
-
```

We would create the Actions thus:




We first create 3 variables: AttachmentName, AttachmentTempLocation, AttachmentNumber

We then create an If block to execute if the message contains more than zero attachments.

We then use the Read/Write Text file to create the header section. Enable the Append option on the Read/Write Text File action.

We then use a For Each block to create a loop for each attachment.

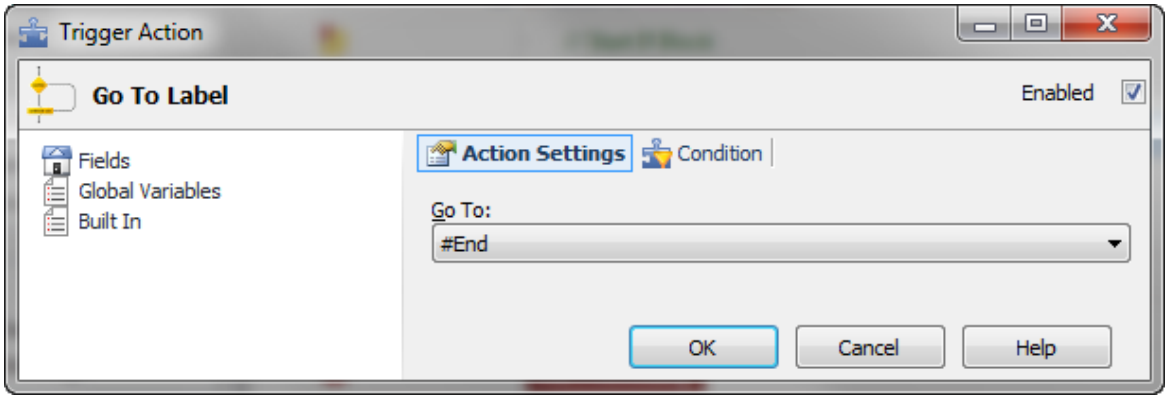
Inside the loop the current attachment name will be assigned to the AttachmentName variable.

We use another Read/Write Text File action to write to the same text file using the % AttachmentName% variable.

After the Next Loop action we use another Read/Write Text File action to add the -End Of Message text to the existing file.

### 8.8.15.3 Go To

Move Action execution to the specified Label.




You can assign a **Condition** to the Go To Action to create a Conditional Go To, eg: If %Email% Is Blank Then Go To #End

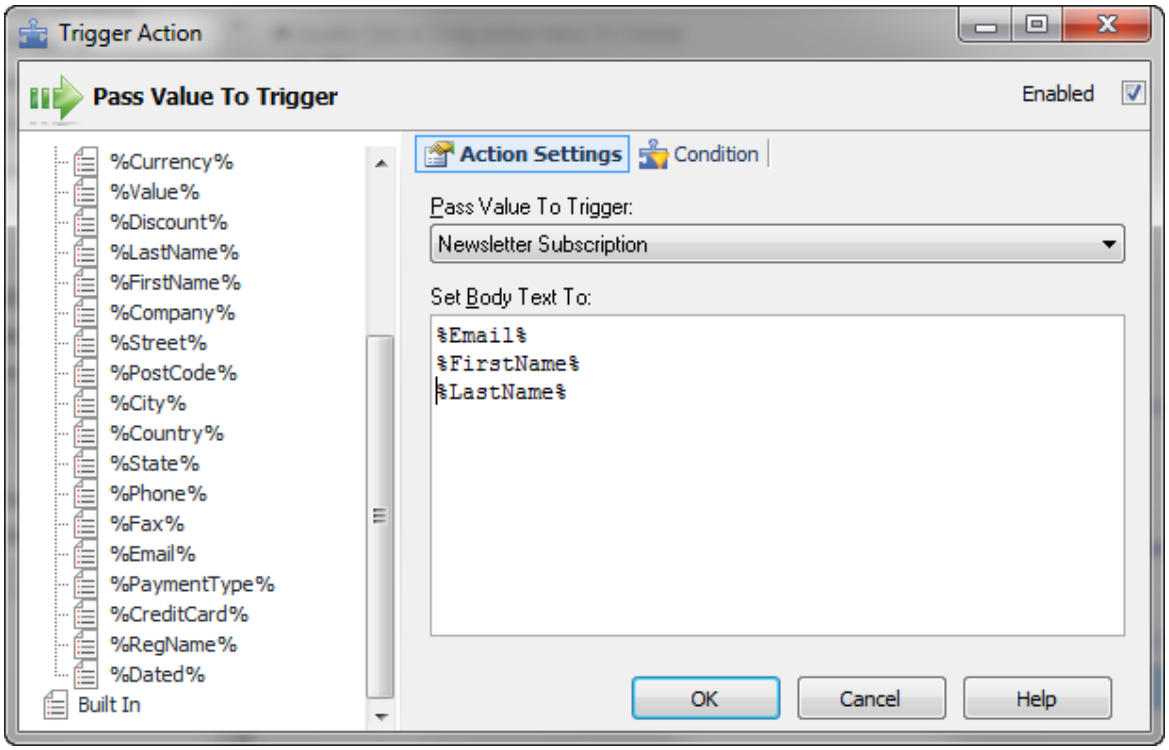
You need to avoid looping. For example, if you create a 'Start' label at the beginning of the Action List and then have a Go To #Start without any means of exiting then you could create a loop. Email2DB monitors for looping and exits automatically - however it would still cause your Trigger to execute slowly.

### 8.8.15.4 Label

The Label Action is used with the **Go To** Action to move processing to another section of the Action List. You can create any number of labels. They do not effect processing other then when used with the Go To Action.

### 8.8.15.5 Pass Value To Trigger

Passes text to another Trigger for processing.



The Pass To Action can be used to split up processing of sub-sections of a message into multiple Triggers.

Set the **Pass Value To Trigger** drop down to the Trigger you want to pass the text to. The Trigger must be in the same Account.

Set the **Body Text** To a value to pass to the Trigger. This can be fixed text, %field% replacements or both.

Email2DB will then create a new message and pass it to the specified Trigger. This Trigger will process the message independently.

#### 8.8.15.6 End Processing

This Action will end processing for the current message. Email2DB will move on to the next message.

You can add multiple End Processing Actions. When used inside If Blocks it allows you to just stop processing depending on a Condition.

#### 8.8.15.7 Comment

Any number of Comment Actions can be added.

Comments can also add to the Email2DB Log when Triggers are executing unless you enable the **Do Not Show In Log** option.

Comments can contain %field% replacements - which will be replaced before the comment is added to the log during Trigger execution.

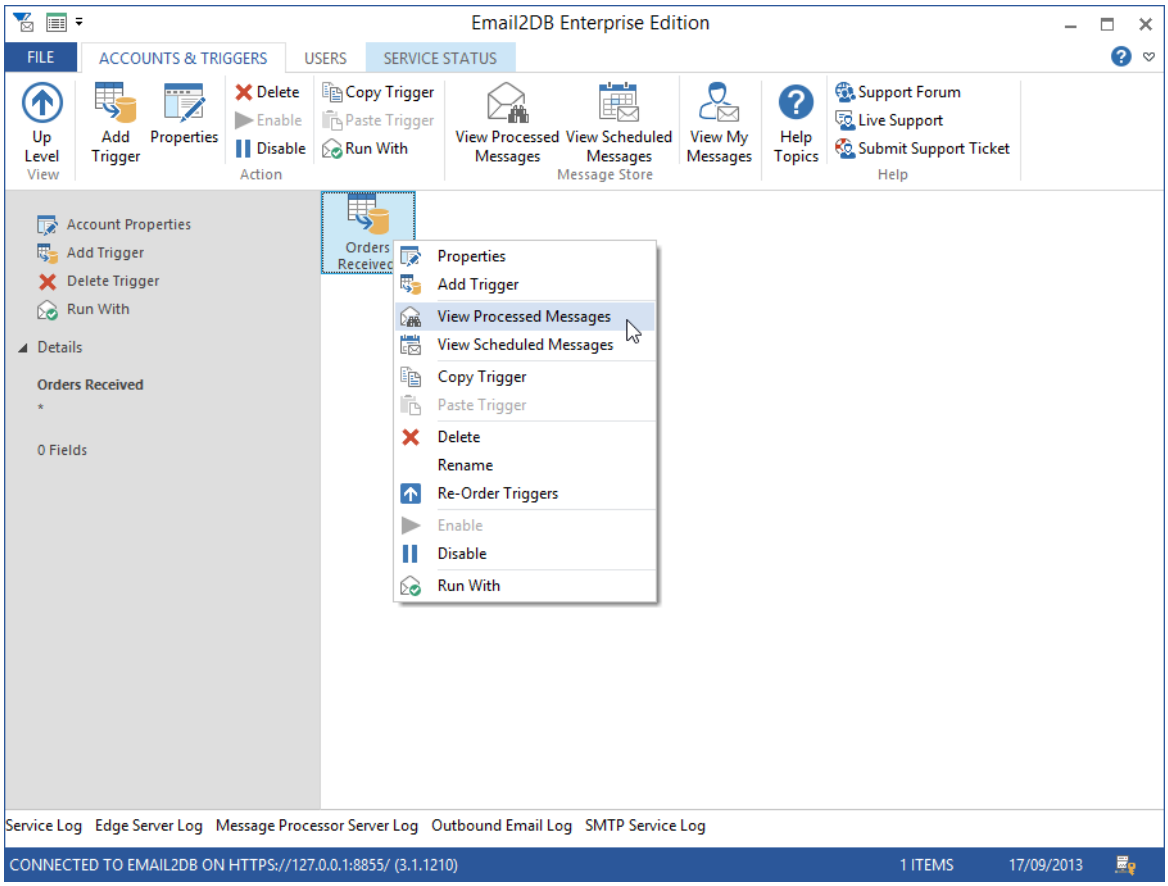
# Part

# 9 The Message Store Database

When Email2DB reads a messages from the source it stores a copy of the message in its own 'Message Store' database. This prevents the same message from being checked and processed twice for the same Email2DB Account and enables you to easily view all messages processed by Email2DB.

Against each Account you also have the option of storing a copy of the full message in the database or just the message ID.

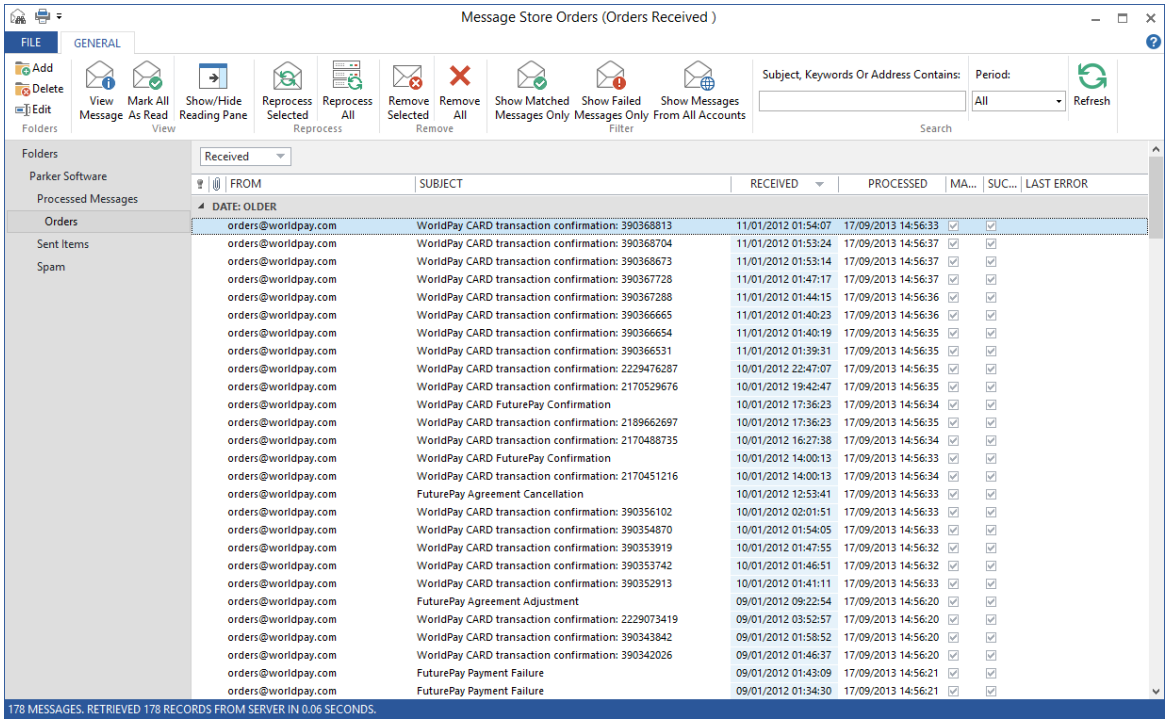
You can view the Message Store for any Account (and for specific Triggers within an Account) by right-clicking an Account or Trigger and selecting **View Processed Messages**.




See Also: Viewing Messages
See Also: Reprocessing Messages
See Also: Organizing Folders

# 9.1 Viewing Messages

In the Email2DB Administrator, right-click any Account or Trigger and select **View Processed Messages**. A list of all processed messages will be displayed for the selected Account/Trigger.



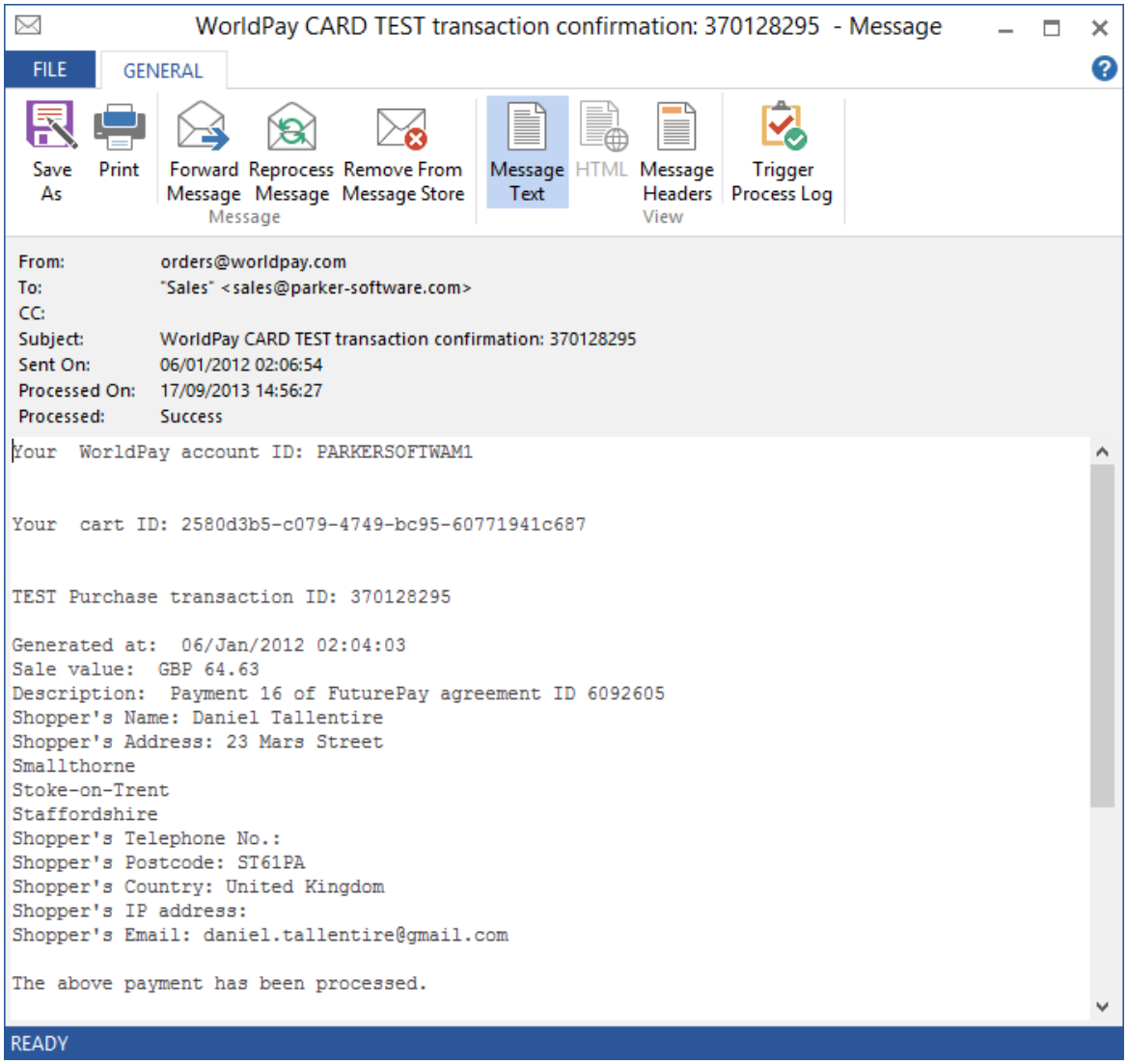

## Searching

You can select a date range in the **Period** drop-down, and/or specific text in the **Subject, Keywords or Address Contains** entry. Click the **Refresh** button to update the list depending on the criteria. See Also: Add Message Store Keywords

Click the **Failed Messages Only** button to only view messages where the Trigger actions caused an error (for example, if your database failed to update). You can view the last error message generated in the 'Last Error' column.

The **Show Messages From All Accounts** button enables you to view messages in the selected folder regardless of the Account/Trigger that processed them. By default when you view the message store only messages for the currently selected Account or Trigger are shown. Select this button to view all messages in the selected folder.

Click the **View Message** button or double-click a message line to view the full message.




The **Message Text** option shows the plain text message. The **HTML** option shows the HTML formatted message. The **Message Headers** option shows the complete MIME headers for the message. If the message contains any attachments, these will be shown. You can right-click an attachment and save it to a folder on your PC.

Click **Trigger Process Log** to view the Email2DB Log File for the Actions performed when the message was processed.

Click the **Forward Message** to forward the message as a new email to someone. Enter the email address to forward to (you can use multiple addresses separated by commas).

## Deleting Messages

You can delete a specific message, or all messages currently shown in the list. If you delete a message Email2DB will re-read and process the message again - if it still exists at the message source (POP3, IMAP). Click **Remove Selected** to remove all selected messages. Click **Remove All** to remove all messages shown in the current view.

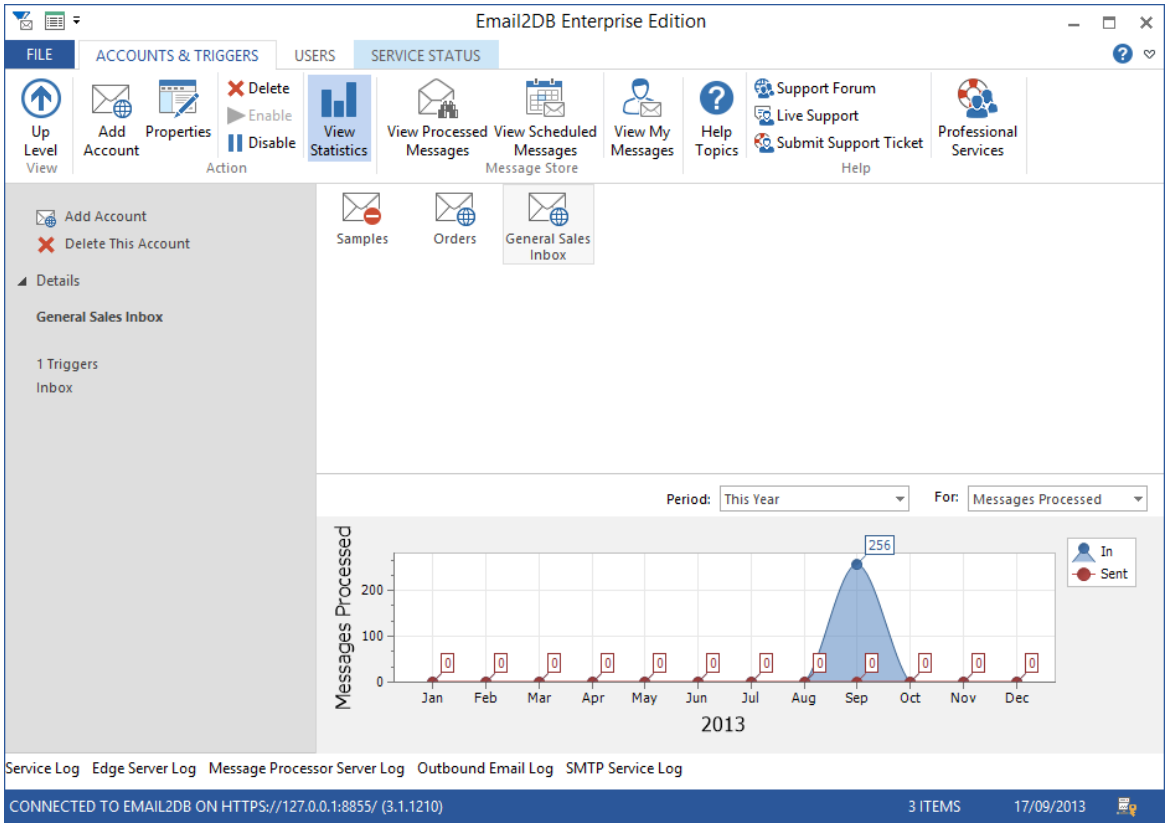
See Also: Organizing Folders
See Also: Reconfiguring The Message Store Database
See Also: Add Message Store Keywords

### Viewing Messages Via The Web

You can also view your message store via a web browser. See: Message Store Web Viewer

## 9.2 Viewing Statistics

You can view a chart showing the number of messages processed for any Account. Select the Account in the Email2DB Administrator and click the **View Statistics** button:




The Chart will show the number of messages processed by the selected Account for each day in **This Month**. You can select **Last Month**, **This Year** & **Last Year** from the **Period** drop down.

The **For** drop down allows you to select additional chart types:

- Messages Received - total messages received for each day, with messages sent also showing (in red).
- Bytes Received - the combined size of all messages received.
- Sender Domain - a pie chart showing the top 10 domain names of messages received.
- Sender Country - a pie chart showing the top 10 sender countries.
- Attachment File Type - a pie chart showing the top 10 attachment types received.

Any **Message Store Token** counts created with the Update Message Store Token action will also be shown here if you have added any of these Action types to any Triggers for the selected account.

## 9.3 Reprocessing Messages

You can reprocess any message or range of messages in the Message Store. If you select to reprocess a message the message will be copied back into the Edge Queue. The Email2DB Message Processor Server will pick up the message and process the Actions defined on the Trigger again.

Select an Account & Trigger, then choose **View Processed Messages** from the Ribbon Bar - or right-click a Trigger and select **View Processed Messages** from the popup menu.

The messages held in the Message Store for the selected Trigger will be displayed.

Highlight the message you want to reprocess and click the **Reprocess Selected** button, or right-click the message and select **Reprocess This Message** from the popup menu.

You can select a range of messages by holding shift and the cursor up or down. The selected messages will be highlighted. The Reprocess Selected option will then apply to the selection.

Click **Reprocess All** to reprocess all the messages in the current view. If you have limited the message view by selecting a **Period** or **Subject Or Address Contains** text, then only those messages will be reprocessed when the Reprocess All option is used.

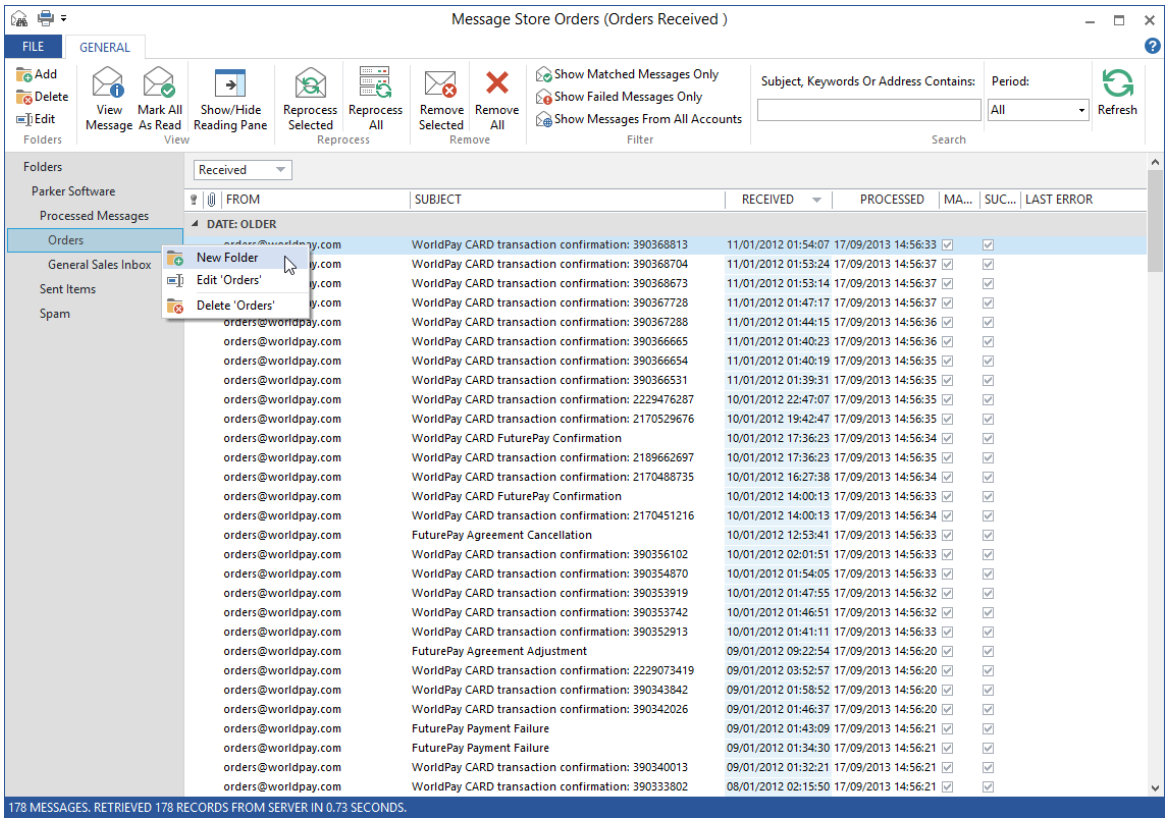
## 9.4 Organizing Folders

The Message Store database can be organized into folders. Email2DB creates 3 folders by default:

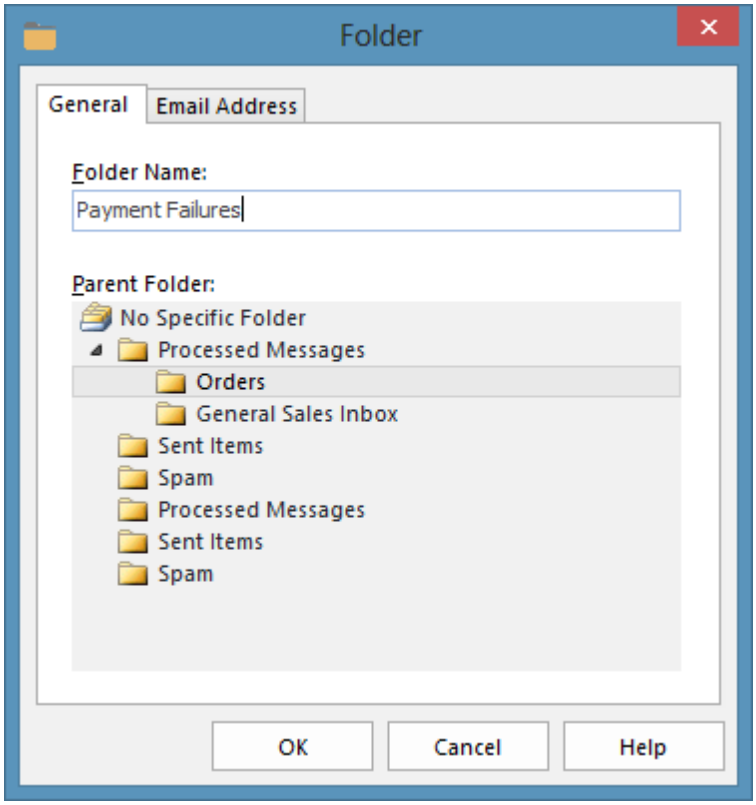
Processed Messages - all processed messages are stored here.
Sent Items - if you create an 'Outbox' Account, then messages processed by the Outbox Account will be stored here.
Spam - Emails received by the built-in mail server that have been marked as spam are stored here.

You can create any number of folders and sub-folders. You can then assign a folder to a Trigger, so that messages that are processed by the Trigger are saved to a specific folder.

Whilst viewing the message store, select a folder and click the **Add** folder button, or right-click a folder and select **New Folder** from the popup:




The **Folder** form will be displayed:




Enter the **Folder Name** and select the **Parent Folder**. Click **OK** to save the folder.

Your Triggers can now save processed messages to this folder. Select the **Save To** option on the Trigger Properties and select any of your folders to save processed messages to.

You can also drag and drop messages to move them into different folders.

## 9.5 Viewing Scheduled Outgoing Messages

When you send a Scheduled Message using the Send An Email or Send An SMS Message Action the outgoing message is added to the Outbox table in the Email2DB Message Store. The ScheduledDate field contains the date & time when the message should be sent.

You can view pending Scheduled Messages for any Account & Trigger.

Select an Account or Trigger and click **View Scheduled Outgoing Messages** on the Ribbon Bar, or right-click an Account or Trigger and select **View Scheduled Messages** from the popup menu.

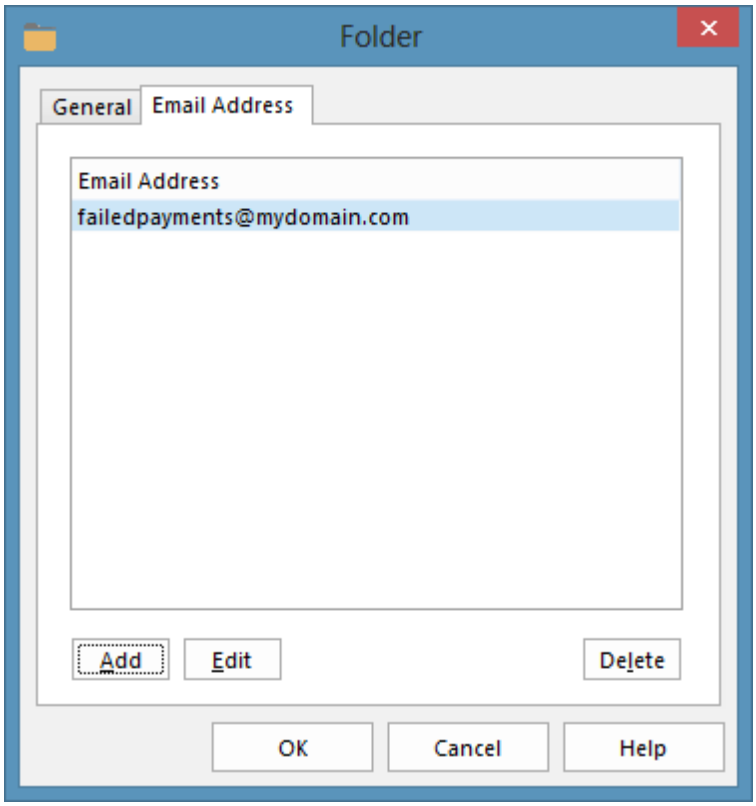
All pending outgoing messages for the selected Account or Trigger will be shown.

Click Delete to delete the currently selected message. Click Delete All to delete all messages currently in the view. You can search for messages for a specific email address or with specific text in the subject by entering text in the **Subject or Address Contains** entry and clicking **Refresh**.

## 9.6 Mail Enabling Message Store Folders

You can assign one or more email addresses to Message Store folders. When the Email2DB mail server receives an incoming email it will automatically place the email in the correct folder without the need to create a Trigger to do so.

Select the **Email Address** tab of the Message Store **Folder** properties:




Click **Add** to add an **Email Address**. Each folder can accept emails for any number of addresses.

If you specify just the alias for an email address (IE: the name without the @ sign and domain) then the mail server will assume the default hostname specified in the Mail Server options for the domain name. For example, if the hostname is 'email2db.com' and you specify 'sales' for the email address, then Email2DB will accept messages for 'sales@email2db.com' for the user.

Wild cards can be used in email address. For example, sales@* would accept any incoming email addressed for the alias 'sales' - regardless of the domain. stephen*@email2db.com would accept emails for stephen@email2db.com, stephen.parker@email2db.com, stephenp@email2db.com etc.

## 9.7 Message Store Web Viewer

You can also view your message store via the included web application which is part of the Email2DB Web Services.

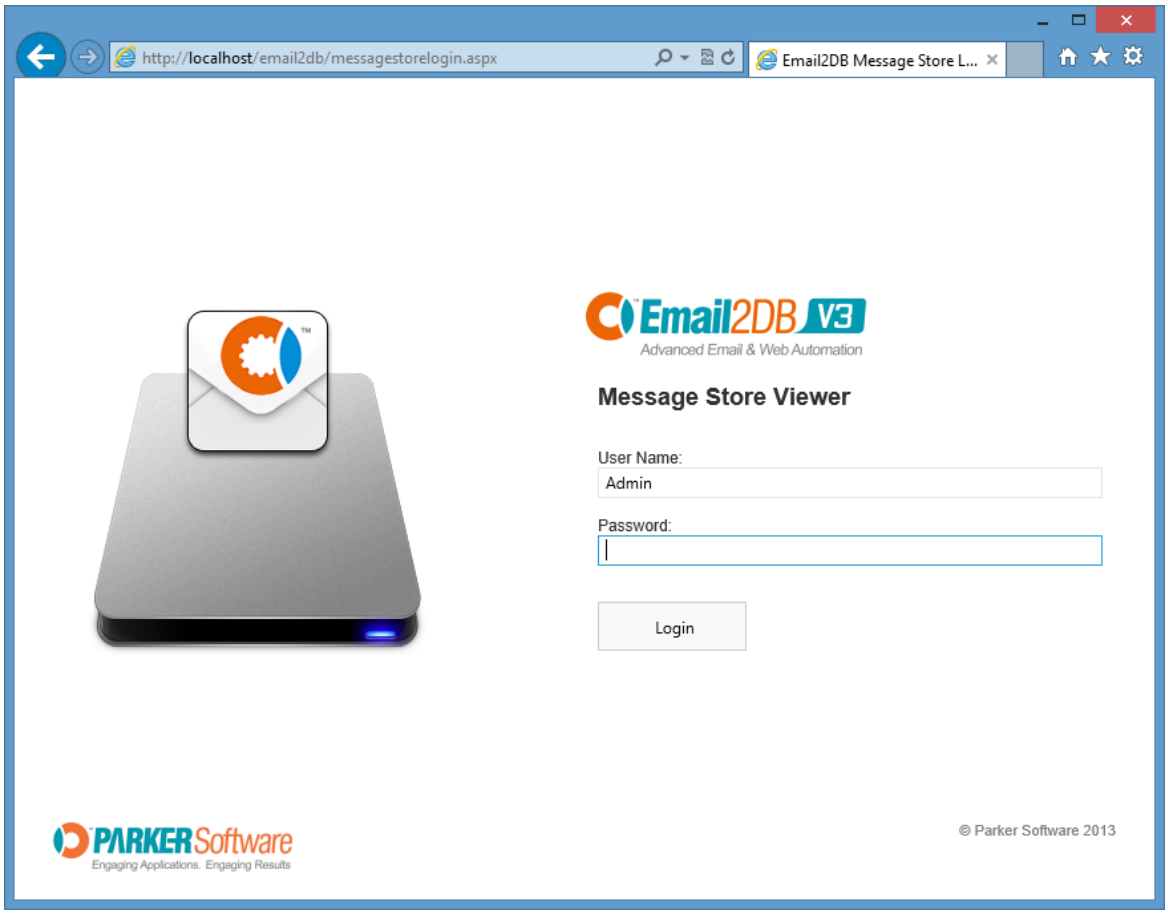
Ensure the Email2DB Web Services are configured and working first.

Then use your browser to open:

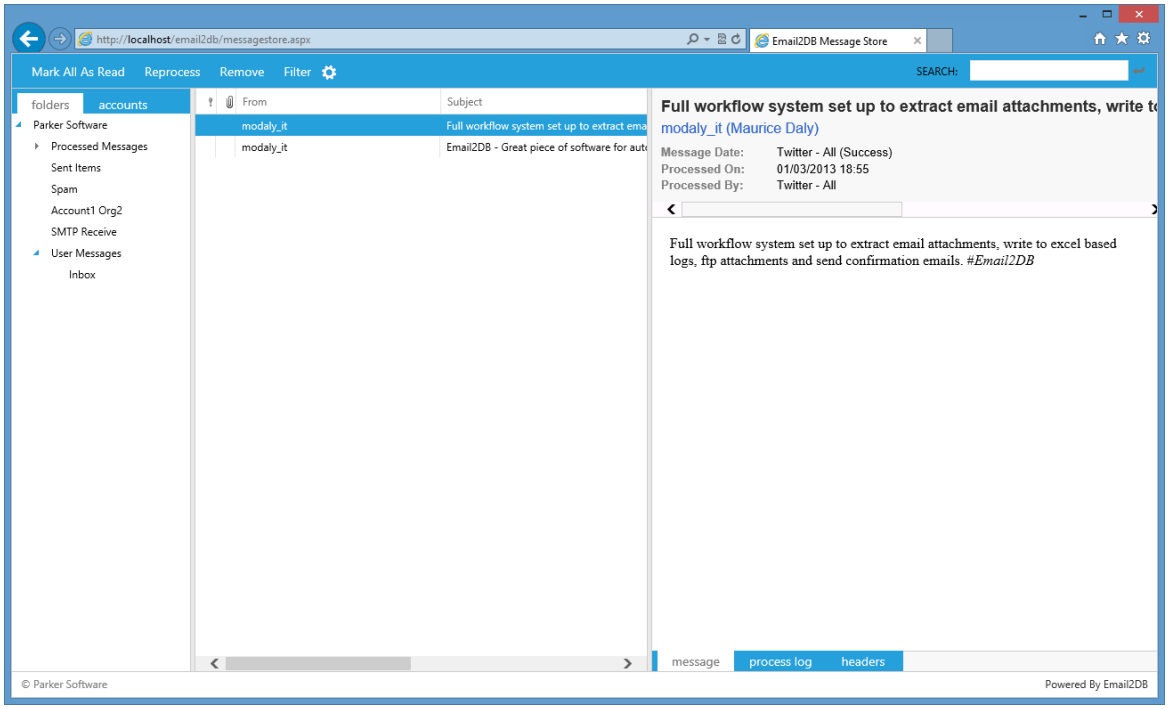
http://{email2dbserver}/email2db/messagestorelogin.aspx

Where {email2dbserver} is the DNS name or IP address of your Email2DB computer.

The login form should then be shown:




Enter your **User Name** & **Password**.




Your message store will then be displayed.

You can choose to view messages by Folder or Account. Select the **folders** or **accounts** tabs. Once selected you can select the specific message store folder or Email2DB account from the tree.

You can **Remove** and **Reprocess** messages from here.

The Message Store Web Viewer is compatible with the following browsers:

- Internet Explorer 9 or higher.
- Google Chrome.
- Safari.
- Firefox.

# Part

# 10 The Email2DB Mail Server

Email2DB Enterprise Edition is able to receive messages for processing directly via SMTP. Email2DB acts as a mail server in its own right and can receive messages for processing - or relay messages onto the intended recipients.

When a new email is received by the Mail Server, the 'To' Address is checked and assigned in the following order:

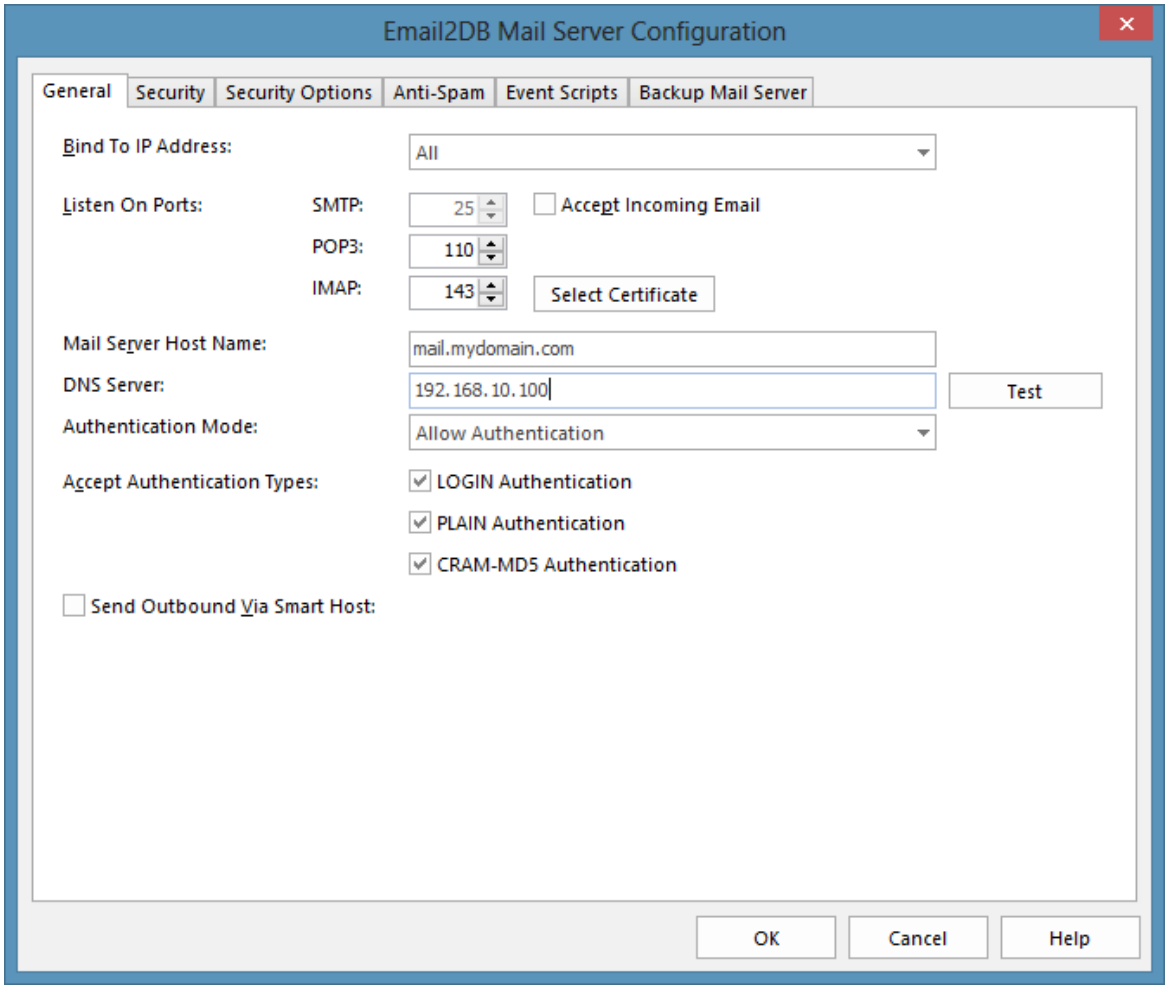
1. Email2DB checks the Accounts - SMTP Receive settings. If a match is found the email is passed to that Account and the Triggers are executed immediately. The email will then be stored in the Message Store against that Account.
2. Email2DB checks the Users. If the email is for a User then the email is added to the Message Store against the matching User.
3. Email2DB checks the Folders. If the email is for a Folder then the email is added to the Message Store against the matching Folder.

If the incoming email does not match an **Account**, **User** or **Folder** then the Mail Server will relay the email onto the intended recipient provided that the sender is allowed to relay. By default only senders on your local network will be allowed to relay (by default, the Email2DB Mail Server is a 'closed relay' and thus cannot be used by spammers). You can add more Trusted Hosts - these are remote IP addresses that will be allowed to relay.

## Mail Server Options

Choose **File** - **Mail Server Options** to configure the Email2DB built-in mail server.

### General Options




Select the **Accept Incoming Email** option to enable the mail server. (This option is not available in the Small Business Edition).

The mail server will listen for incoming SMTP connections on all IP addresses on your system. You can restrict it to listen on just one IP address by selecting the IP in the **Bind To IP Address** drop down.

The **Listen On Ports** entries allow you to change the default listening ports for the SMTP, POP3 and IMAP services. Setting any of these ports to zero will disable listening.

**Mail Server Host Name**

This specifies the name of the mail server. It must not be empty. It is used by the mail server when it communicates with clients or introduces itself to other mail servers. Typically it would be mail.yourdomain.com. If the mail server will only be used internally within your own network (and not be accessible from the Internet) then any text can be used.

**DNS Server**

If the Email2DB mail server will be used to send outgoing emails you must specify the DNS server to use. Email2DB will lookup the MX (mail exchange) records for external domains using the DNS server specified. This entry defaults to the DNS server specified on your default network connection settings.

**Authentication Mode**

This option enables you to control which SMTP authentication methods the mail server will allow for incoming SMTP Connections. By default the mail server will allow authentication using any authentication method.

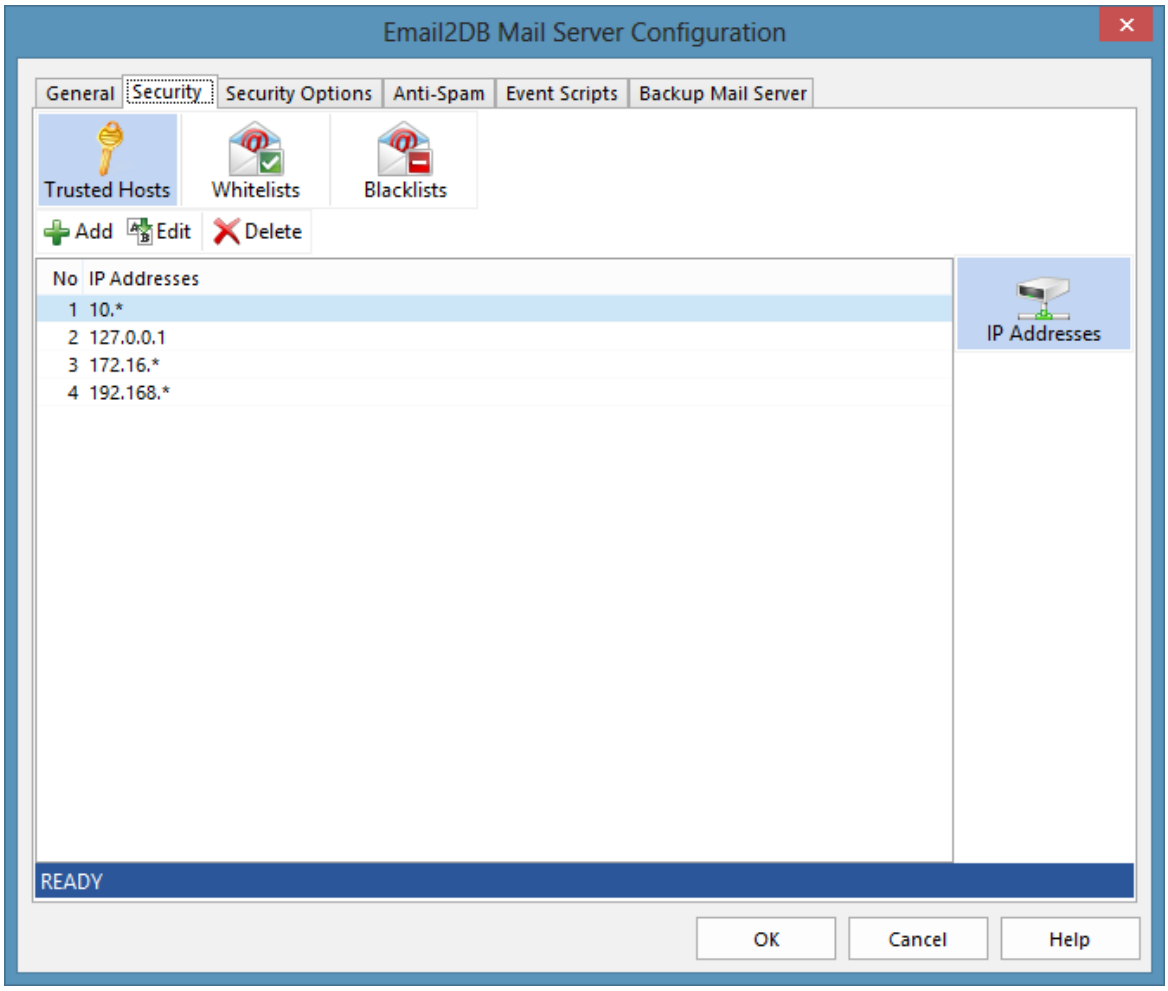
**Send Outbound Through Smart Host**

By default Email2DB sends outgoing messages directly to the recipients mail server. You can instead relay outgoing messages via another mail server (a smart host). Select this option and then provide the Smart Host properties.

## 10.2 Mail Server Security Options

### Security

The security tab is used to configure the mail server security options.




In this section you define your 'Trusted Hosts', 'Whitelists' and 'Blacklists'.

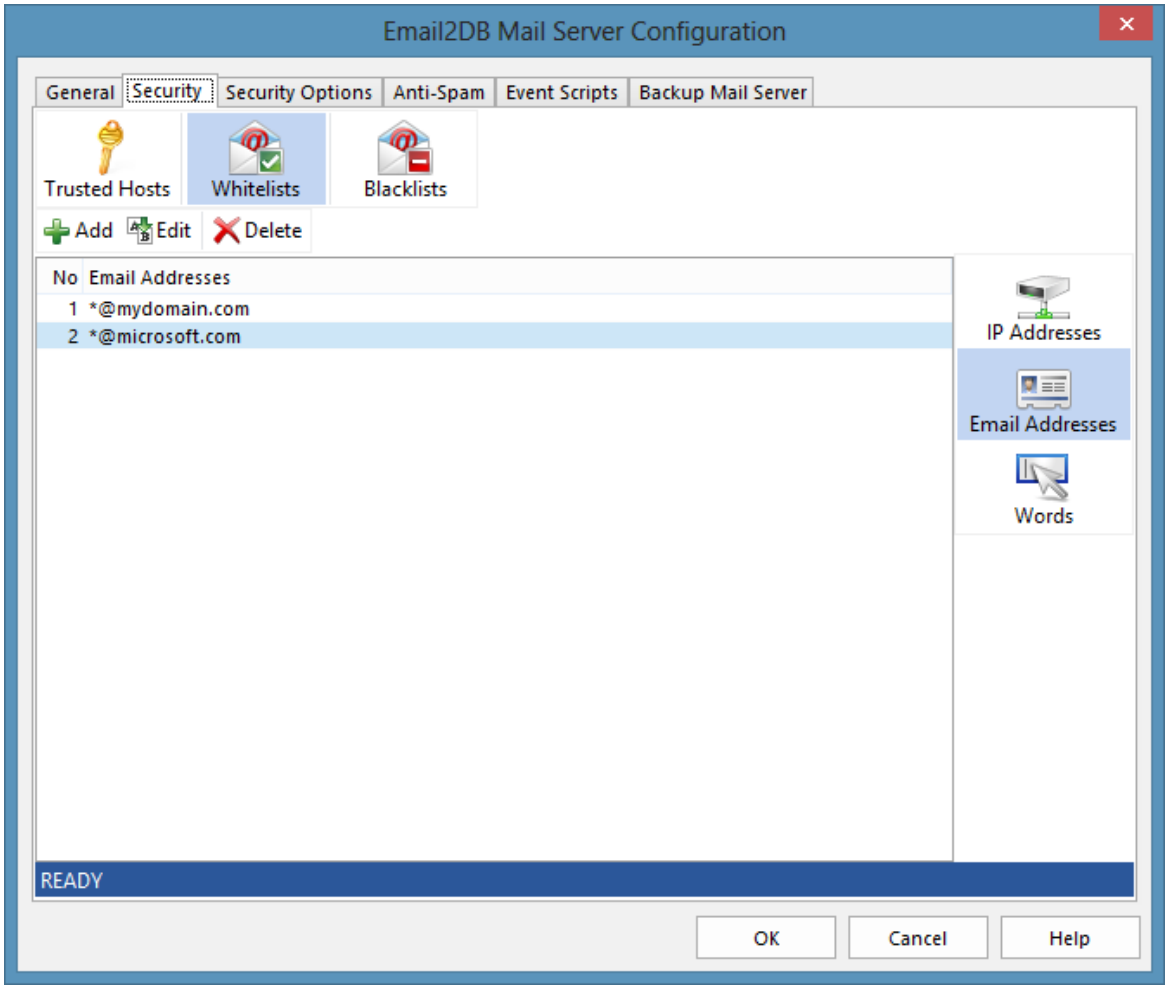
### Trusted Hosts

Trusted Hosts are IP addresses of PC's that you trust. Connections from these IP addresses will be allowed to 'relay' through the mail server. That is, they will be allowed to send email to anyone via the mail server.

By default Email2DB adds all Internal Addresses to the trusted hosts list. An Internal Address is an IP address that is inside your network. By default all External IP addresses will NOT be allowed to relay. This is called a closed relay. It prevents spammers from outside your network sending emails via your mail server.

You can add your own IP external IP addresses to the trusted hosts list. Select the Trusted Hosts button, then click the **Add** button.

### Whitelists



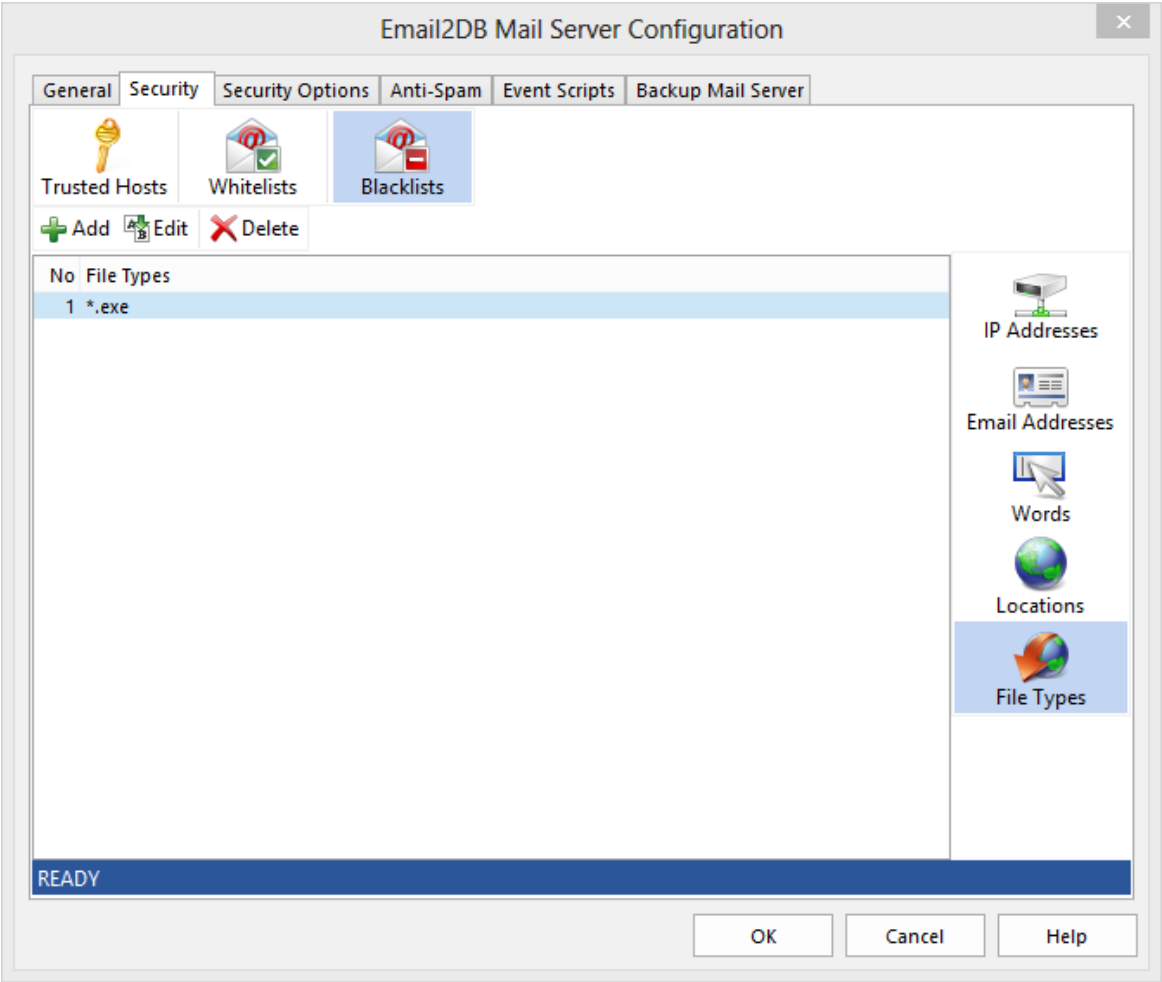

The Whitelists are lists of IP Addresses, Email Addresses and Words that the mail server considers good.

When the mail server receives an incoming email it first checks if the senders IP address or from address is in the IP or Email whitelist. If so, the email is accepted without further checks. It then checks the body text. If the body contains one or more of the words contained in the Words whitelist, then it is accepted without further checks.

Select the **IP Addresses**, **Email Addresses** or **Words** list from the buttons on the left hand side. Then click **Add** to add a new item.

All items can contain wildcards, so for example: '*@parker-software.com' would accept any email ending with '@parker-software.com'.

## Blacklists



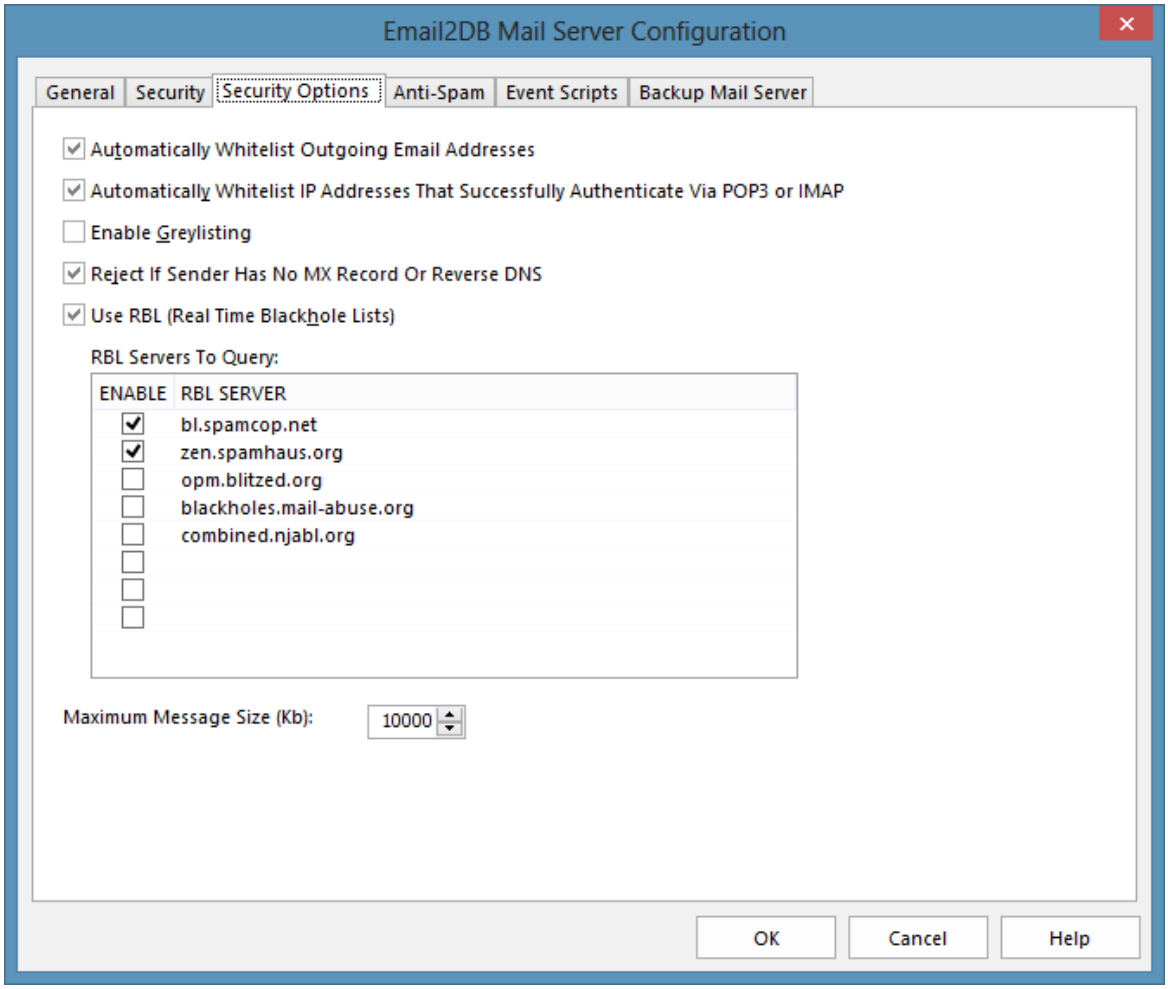

These are the opposite of Whitelists. Any incoming email containing an item in any of the blacklists will be blocked.

In addition to **IP Addresses**, **Email Addresses** and **Words**, you can blacklist **File Types** (by specifying the file extension) and email from Geographical **Locations**. For locations you can specify the Country and/or City. When an email arrives, Email2DB looks up the physical location of the senders IP address using 'geo-ip'.

The Whitelist overrides the Blacklist. So if an email arrives containing an item on the Whitelist then it will be allowed, regardless of whether it contains items on the Blacklist.

## Security Options

The Security Options tab is used to define settings that control the security checks performed on incoming email messages that the mail server receives. These checks are not performed on emails from Trusted Hosts or that are Whitelisted.




### Automatically Whitelist Outgoing Email Addresses

If you enable this option then the 'to' addresses of all outgoing emails will be added to the incoming email Whitelist. This means that if you send an email to someone via the Email2DB mail server, then their address will be whitelisted and they will be allowed to send you emails from then on that will never be blocked.

### Automatically Whitelist IP Addresses That Successfully Authenticate Via POP3 or IMAP

If this option is enabled then the IP address of any email client that successfully authenticates with the mail server will be whitelisted.

### Enable Greylisting

If this option is enabled then the Email2DB mail server will reject the first SMTP connection from an IP address that is not whitelisted. The sender must re-try after 2 minutes. This is an anti-spam mechanism. Most spammers won't retry, so this is an effective way of reducing spam. Once the sender re-tries, their IP address will be stored for 30 days and they won't be asked to re-try again within that period. For more information about Greylisting see: http://en.wikipedia.org/wiki/Greylisting

### Reject If Sender Has No MX Record Or Reverse DNS

If this option is enabled then Email2DB will lookup the MX and reverse DNS records for the senders IP address. If no records are found the email will be rejected.

**Use RBL (Real Time Blackhole Lists)**

Real time blackhole lists are publicly accessible lists of IP addresses of known spammers. If this option is enabled, Email2DB will lookup the senders IP address and check if it exists on any of the lists specified in the RBL Servers To Query. If any of these RBL Servers return a match then the incoming email will be blocked.
For more information about RBL see: http://en.wikipedia.org/wiki/DNSBL

**Maximum Message Size**

You can define the maximum size of messages you will allow the mail server to receive. Specify the value in Kilobytes (10,000 = 10 Mb)

## 10.3 Mail Server Geo-IP

Email2DB is able to assign incoming email messages with a 'Geo-IP' tag. Geo-IP refers to the physical location of the senders IP address. The Geo-IP data includes, Country, City and Region. It can also include the senders Organization name if you install the full Email2DB Geo-IP Database.

Email2DB includes a special database of IP addresses and their physical location. When an email arrives, Email2DB will use the senders IP address (the actual IP address of the senders TCP/IP connection - not the IP address in the Received header) to lookup their Geo-IP data from this database. Email2DB then adds the following headers to the email (for example):

```
Email2DB-GeoIP: Stoke On Trent (Staffs) United Kingdom
Email2DB-Organization: Parker Software
```

The Organization name that is shown is the name that is registered to the senders IP address. For most larger businesses this will be the actual business name. For home users and businesses using dial-up connections, it will be their ISP name.

### Benefits

The Geo-IP and Organization name will be used as part of the Spam filtering tokens. So, if you get a lot of Spam messages from (for example),China - then emails received with a Geo-IP tag containing 'China' will increase the spam probability for the email.

You can Blacklist certain locations or Organizations. For example, if you add 'China' to the locations-blacklist then all emails coming from China will be blocked.

You can also access the Geo-IP headers in your Triggers and thus create location or organization based Trigger conditions.

### Installing The Organization Database

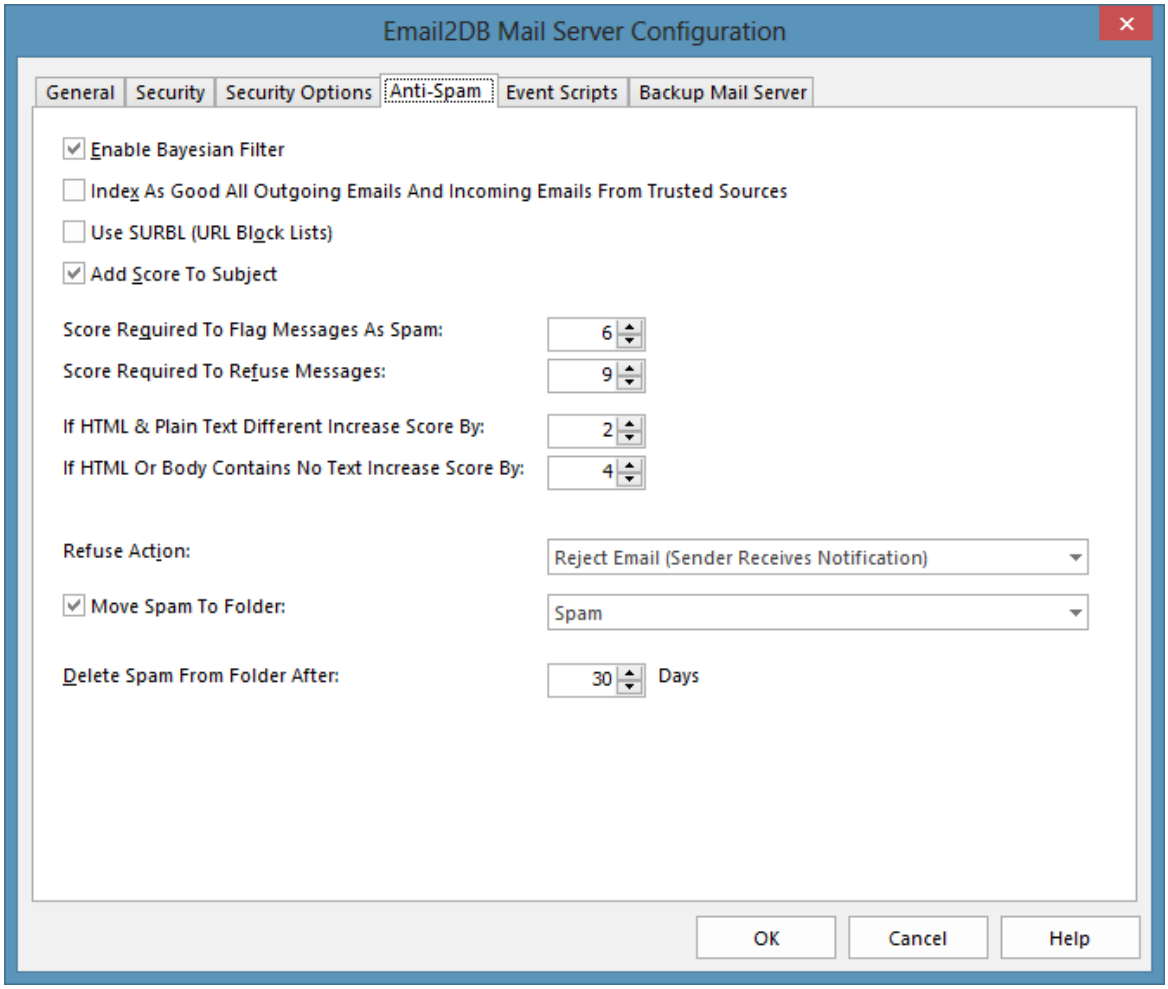
By default Email2DB assigns only the Country, City and Region to incoming emails. The Organization database is not installed as part of the standard setup due to it's size (it is 90mb compressed). If you want to assign the Organization name to incoming emails you will need to install the full Geo-IP database. You can download this by clicking the Check For Updates button on the File menu.

### Updating The Geo-IP Database

IP addresses are being re-assigned all the time, so the Geo-IP database is always changing. We update the Geo-IP database about once per month. You can install the latest version using the Check For Updates option.

## 10.4 Anti Spam Features

The Email2DB Mail Server includes a number of features designed to reduce the number of spam emails that you receive.




When an email message is received by the mail server it will be checked against Trusted Hosts and Whitelists. Any emails that are not from trusted hosts or are not whitelisted will be checked against the security options and spam filter.

**Enable Bayesian Filter**

If this option is enabled then email messages will be scanned through the 'bayesian filter'. Email2DB will split the incoming email into a list of tokens (including headers). These tokens are then looked up from a database. The database contains a count of good messages and spam messages for each token. A subset of the tokens are used to calculate a bayes score. The higher the score, the more likely the message is spam.

The filter will improve in accuracy over time as more tokens are added to its database or good and bad token counts.

For more information about Bayesian spam filtering, see: http://en.wikipedia.org/wiki/Baysian_filter

**Index As Good All Outgoing Emails And Incoming Emails From Trusted Sources**

This option should normally be enabled. It ensures that the bayes database is updated with good

messages. All outgoing emails will be scanned and tokens added to the database as good. All incoming emails from Trusted Hosts or that are Whitelisted will also be indexed as good emails.

**Use SURBL (URL Block List)**

If this option is enabled then any incoming emails that contain HTML links will be checked against the SURBL list. This is a public list of links that commonly occur in spam messages. For more information about URL Block Lists, see: http://en.wikipedia.org/wiki/SURBL

**Add Score To Subject**

Email2DB will assign a spam score of between 1 and 10 to each message. The higher the score, the more likely the email is spam. Email2DB can add this score to the message subject allowing you to easily see the score when viewing messages.

**Score Required To Flag Messages As Spam**

This defaults to 7. You can lower the value to mark more messages as spam. If a message has a score equal or higher than this value then it will be flagged as spam and optionally moved to the spam folder (see below).

**Score Required To Refuse Messages**

If any message spam score is equal to or higher than this number then it will be rejected by the mail server.

**Refuse Action**

Messages refused due to their spam score can either be deleted, or rejected. If the message is deleted, then the sender thinks you have received the email - you just won't see it. If the message is refused then the sender will receive a NDR (non delivery report - or a 'bounce' message).

**Move Spam To Folder**

For spam that is not rejected, you can move the messages to a folder in the Message Store, instead of the Inbox of the recipient. The Spam folder is shared for all users. You should then select the folder in the message store to use. A 'spam' folder is created automatically. The email alias 'spam' is assigned to this folder. Any email sent to the mail server to 'spam@' will be indexed as spam.

**Delete Spam From Folder After**

Email2DB can delete old messages from the Spam folder after a certain number of days. Specify zero days to never delete.

## 10.5 Training The Spam Filter

The Email2DB Spam filter will only be effective when it has built-up a database of good and bad tokens.

You need to 'train' the spam filter before using it in a production environment.

### Training For Good Messages

Any email that is received by the mail server from a Trusted Source or any email outgoing email that is sent through the mail server by a Trusted Source will be indexed as a good message. You should send yourself lots of good messages. At least 100 messages - containing regular words and phases that are applicable to your everyday work.

### Training For Spam Messages

The Checked Message folder that is assigned to receive spam messages (by Default the 'Spam' folder), can be assigned one or more email addresses (eg spam@yourdomain.com). Any emails that are sent directly to any address assigned to this folder will be indexed as spam automatically. Training for spam messages then involves sending lots of spam messages to your spam@ address. The more the better - but at least 100 before you should consider the spam filter 'ready'.

You can also drag and drop messages into the Spam folder whilst viewing the Message Store. Messages moved to the Spam folder will be automatically indexed as spam by the Email2DB Mail Server.

## 10.6 POP3 & IMAP Client Access

The Email2DB Server includes POP3 and IMAP server. This enables any email client to be used to retrieve email messages stored in the Email2DB Message Store.

POP3 and IMAP users connect using the username/password specified on their User record.

For IMAP access, users will be able to see all of the Message Store folders that they are allowed access to in addition to their inbox. For POP3 users, only the inbox will be visible.

## 10.7 Using As A Backup Mail Server

The Email2DB Mail Server can be used as a backup mail server. What this means is that if your main mail server is down, messages will be delivered to the Email2DB mail server. Email2DB will then forward the messages to your main mail server when it is back up.

Using the **Mail Server Options**, select the **Backup Mail Server** tab.

Enter the **Domain Names** that you want Email2DB to act as a backup mail server for.

In the **Send To Server** column you can specify the DNS name or IP address of the mail server to forward the messages to. Leave this entry blank to use the first mail server specified in the MX record for the domain.

You will need to edit your MX record and add the IP address of the Email2DB PC. Make sure this has a lower priority than your main mail server.

### Should You Use A Backup Mail Server?

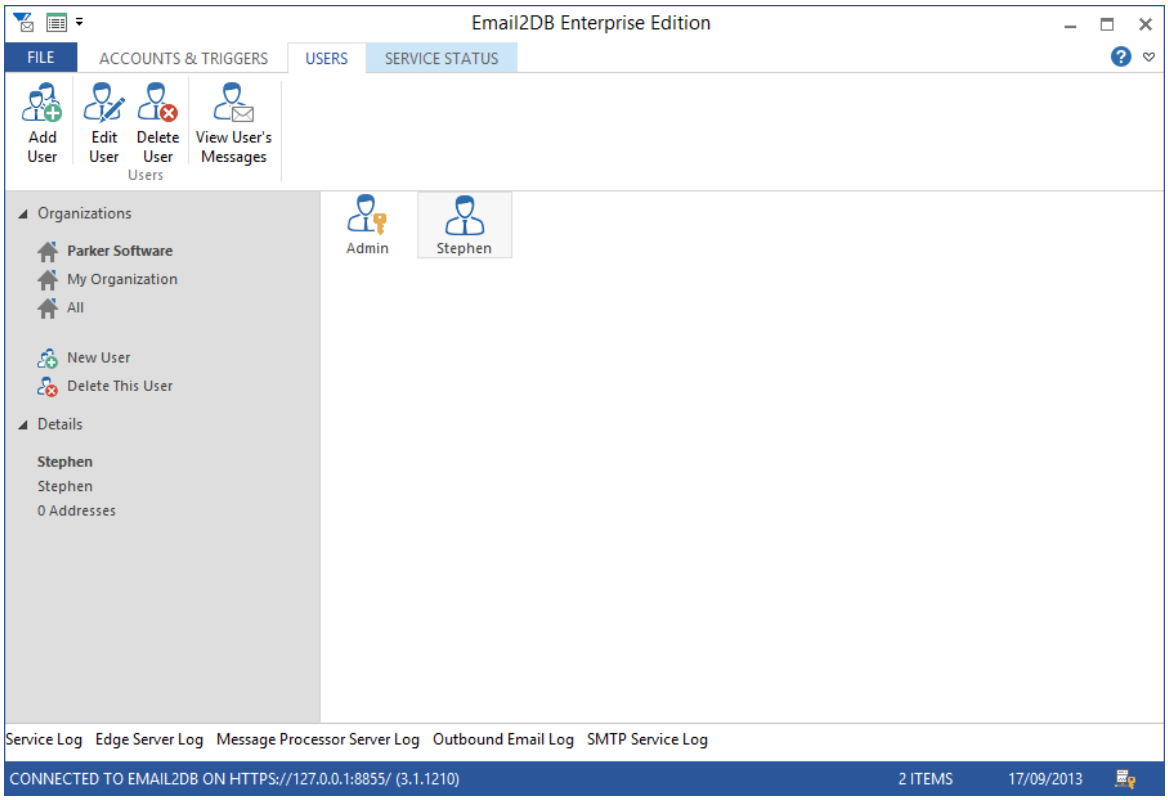
Many experts now recommend against specifying backup mail servers in your MX records. The reason is spam. Unfortunately spammers now send most of their emails to the backup mail server found in recipients MX records. The backup mail server then sends the emails on to the primary mail server. The primary mail server is likely to trust the backup mail server, meaning the spam could get through. Most senders will re-try sending you an email if your primary mail server is down anyway - so the usefulness of backup mail servers compared against the extra spam threat is now negligible.

# Part XI

# 11 Users

Users using the Email2DB Administrator and Client Applications must login with a user name & password.

When you first install Email2DB a default user called 'Admin' is created with no password. You can edit this user and create more users. Click the **Users** tab on the Ribbon Bar.




Click **Edit User** to edit the selected User. Click **Add User** to add a new user.

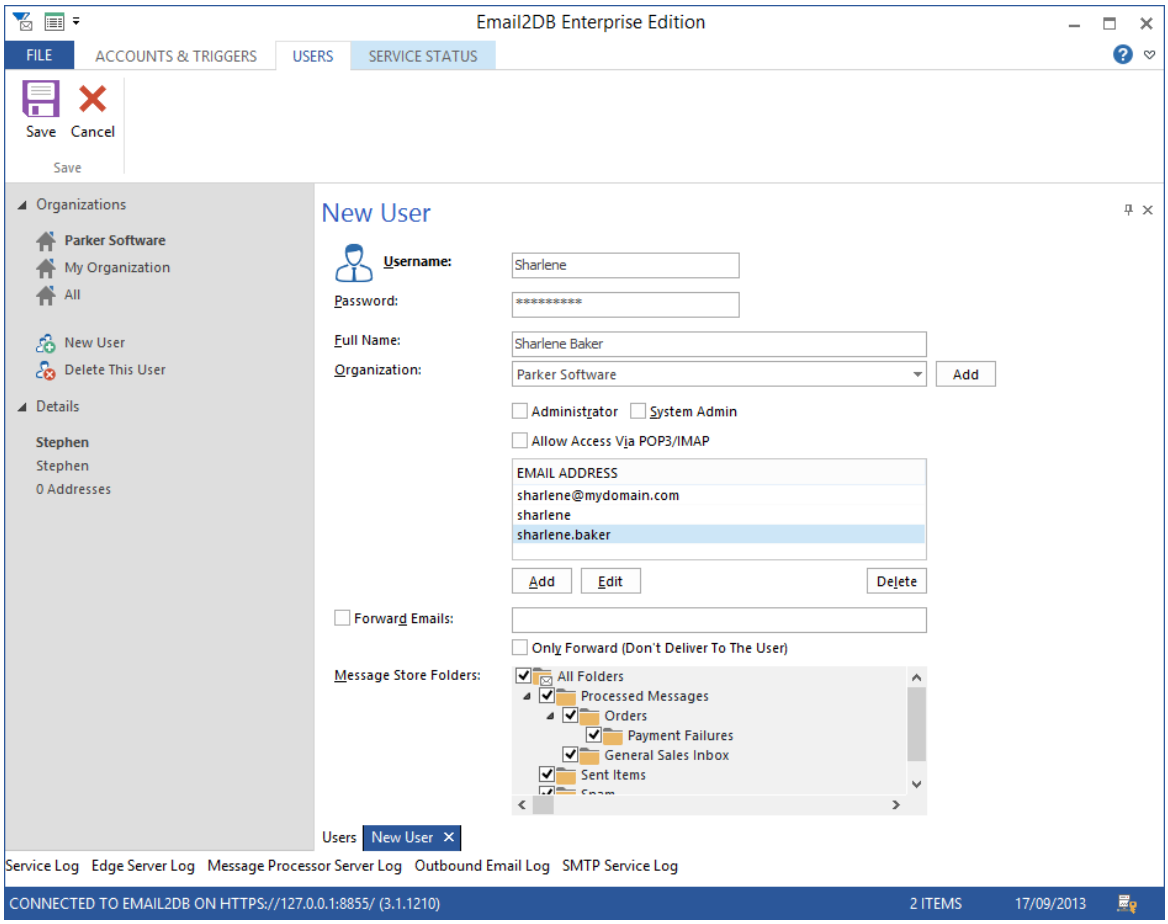
You can only access the Users tab if you are logged on with a User who has Admin rights.

See Also: Adding Users

## 11.1 Adding Users

You can create any number of User accounts. When an incoming email arrives via the built-in mail server, Email2DB will check if the message is for a user, and place the message into the users Inbox in the Message Store database without the need to create a Trigger to do so. Users can view their messages using the Email2DB Administrator itself, or via a regular email client (like Windows Mail, or Microsoft Outlook).

Click the **Users** tab on the Email2DB Administrator Ribbon bar, then click **Add User** to create a new user.




Enter the **Username** and **Password**. The username must be unique. Enter the users **Full Name**.

Select the **Organization** that this user is part of. When a user logs into Email2DB they will only see Accounts, Triggers & Users created by users in the same Organization. Click **Add** to create a new Organization.

Select the **Administrator** option if this user has full access to all Message Store folders and can create other users within the selected Organization. If the **System Admin** option is enabled then the user can see all Accounts & Users - regardless of the Organization.

Select the **Allow Access Via POP3/IMAP** option if this user can use an email client to view their messages.

You can now assign any number of **Email Addresses** to this user. Any incoming messages that arrive via the built-in mail server addressed to any of these addresses will be placed in the users folder in the Message Store. Click the **Add** button to add an email address. If you specify just the

alias for an email address (IE: the name without the @ sign and domain) then the mail server will assume the default domain name specified against the Organization. For example, if the hostname is 'email2db.com' and you specify 'sales' for the email address, then Email2DB will accept messages for 'sales@email2db.com' for the user.

Wild cards can be used in email addresses. For example, sales@* would accept any incoming email addressed for the alias 'sales' - regardless of the domain. stephen*@email2db.com would accept emails for stephen@email2db.com, stephen.parker@email2db.com, stephenp@email2db.com etc.

**Message Store Folders**

For non-admin users you can restrict access to folders in the Message Store on a user basis. Select each folder that you grant the user access to. When users view messages either via the Email2DB Administrator, or via an email client, they will only see the folders they have access to.

**Forward Emails**

You can select to forward all inbound email for this user to another address.

Enable the **Forward Emails** option and enter the **Forward To** address. If the **Only Forward** option is enabled then Email2DB will forward the message then delete it. The message will not be saved in the Email2DB users mailbox.

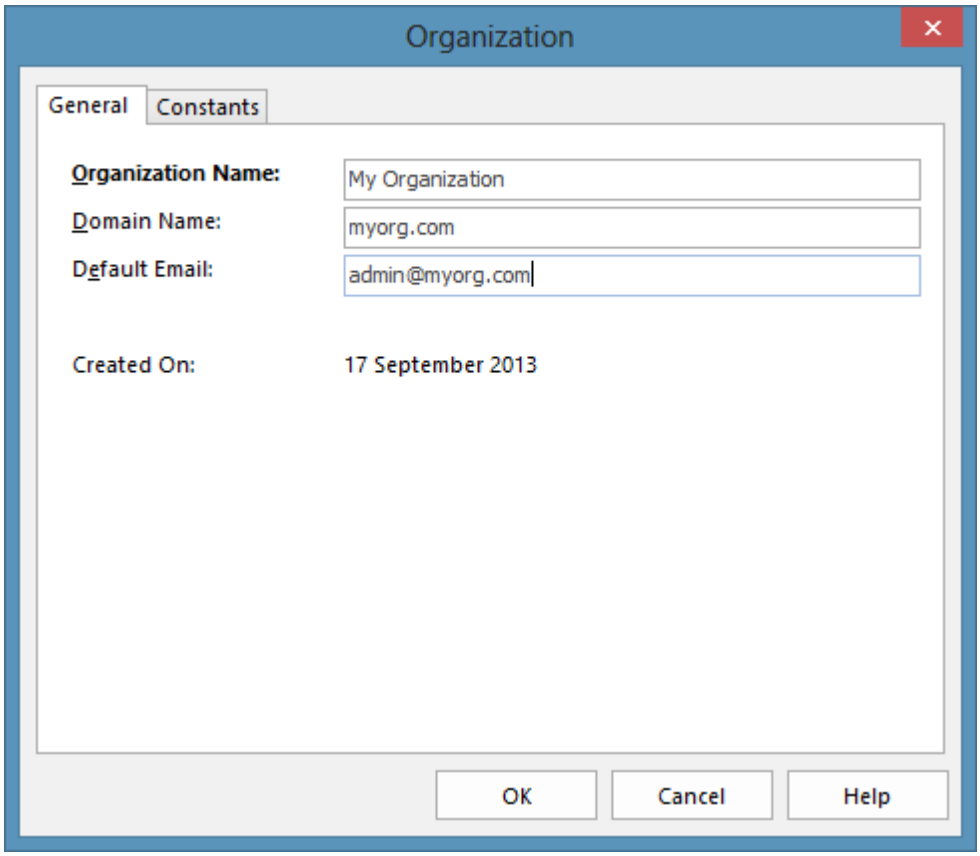
See Also: Organizations

## 11.2 Organizations

Each Email2DB Account and User you create must belong to an Organization. Email2DB creates a default Organization record when it is started for the first time. When a users logs into Email2DB they can only see Accounts & Triggers for the Organization that they belong to.

Under most circumstances you should not need to create additional Organizations - unless you have many users and you want to separate them into groups.

You can create Organizations in the **Program Options - Organizations** tab, or when creating users.

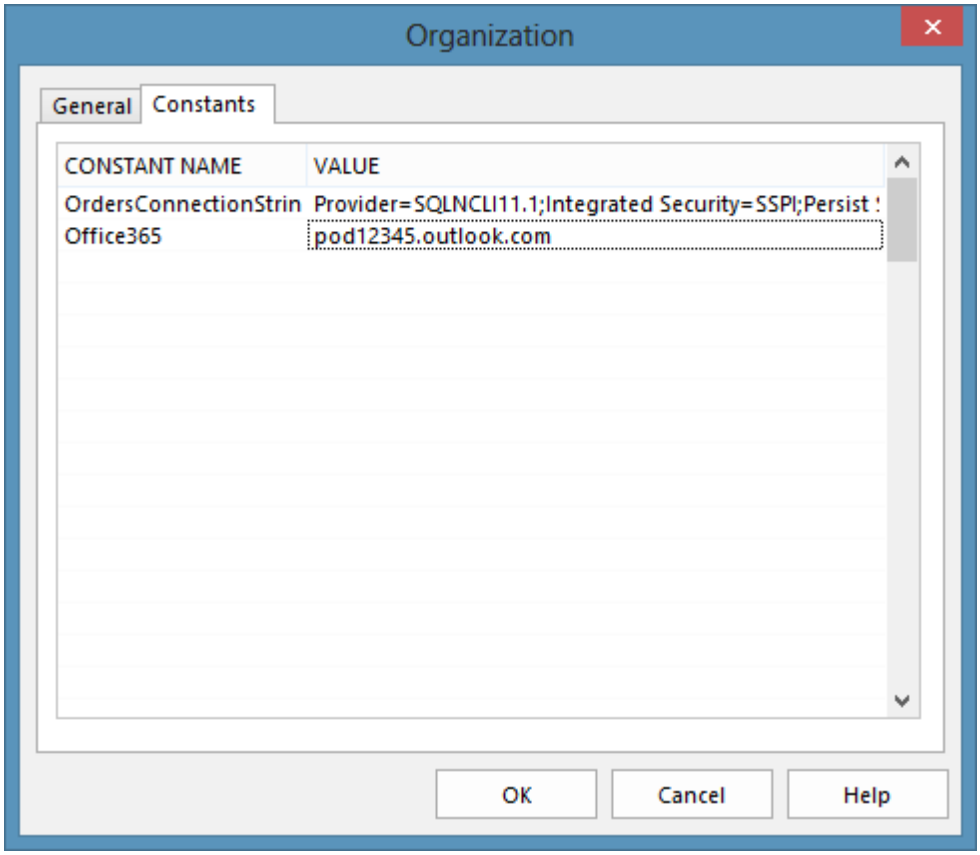



Specify the **Organization Name**, **Domain Name** & **Default Email address**.

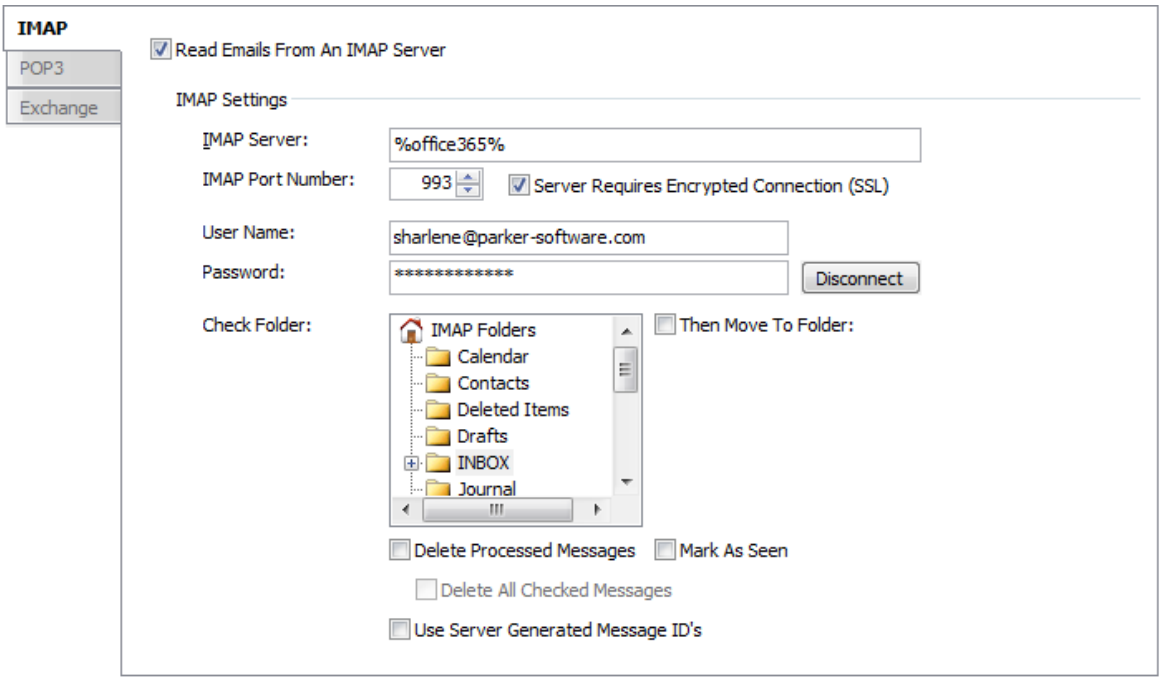
The Organization Name & Domain Name must be unique.

### Organization Constants

You can assign each Organization any number of Constants. A Constant is a name, value pair that you can use as Field Replacements on any of your Trigger Action settings and Account settings.




For example, suppose you create a Constant called 'office365' and assign this the value of your Office 365 server. You can then use this on your Email2DB Account settings:




Unlike Field Replacements, Organization Constants can be used on both Account Properties and Trigger Settings.

See Also: Field Replacements

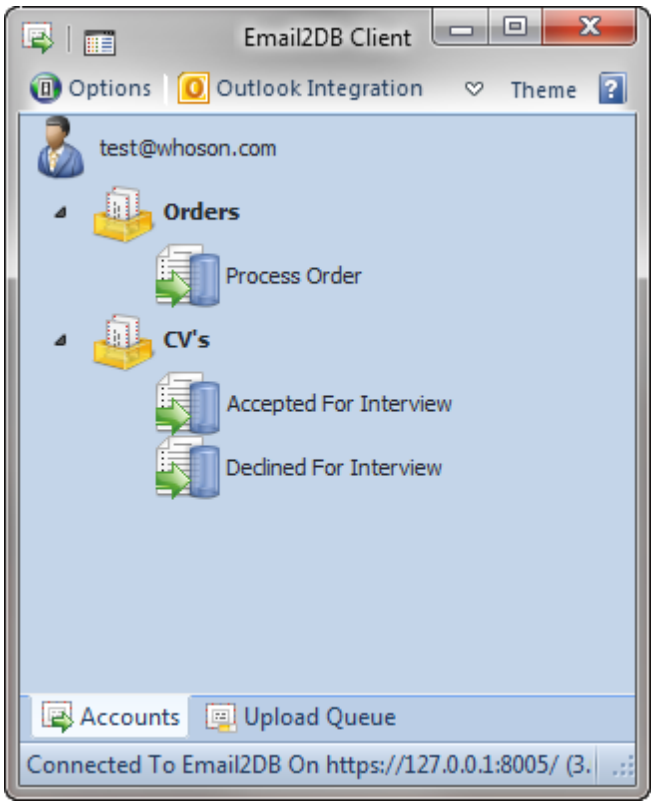
# Part

# 12 The Email2DB Client Application

The Email2DB Client is a stand-alone program that you can install on multiple computers on your network. It connects to the Email2DB Server using a secure HTTPS connection. The number of Email2DB Clients that can connect to the Email2DB Server at the same time depends on your license.

Users can drag and drop email messages, text files or attachments on to Triggers inside the Client. These will be sent to Email2DB for immediate processing. You can drag and drop messages from Microsoft Outlook (MSG format) or any other email client that saves messages as EML files. Any text files dropped onto the Client will be converted into plain text emails. HTML files will be converted to HTML emails. Any other file types will be added as an attachment to a plain text email before being sent to the Email2DB Server.

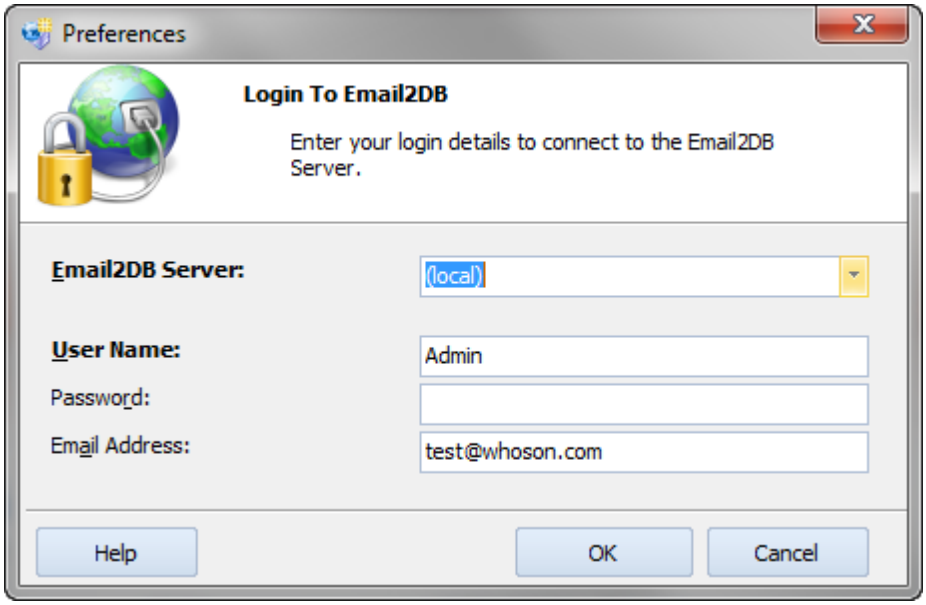
You can also drag files into your user (at the top of the tree). These messages will be saved directly in the Email2DB Message Store against your user account. You can view the message store by right clicking an Account or Trigger, or your User Name and selecting **View Processed Messages** from the popup menu.




The Client is started by choosing Email2DB Client from the **Start** Menu - **Email2DB Version 3** folder. A shortcut is also placed on the desktop during installation.

The Client is installed as part of the main Email2DB Server installation. A separate Client-Only installer is available on our web site.

When the Client starts the user must login.




Enter the computer name/DNS or IP Address of the **Email2DB Server** that you want to connect to. Enter your **User Name** & **Password** and your **Email Address**. Click **OK** to connect.

The Client will then display the Accounts & Triggers that you have access to.

## Outlook Integration

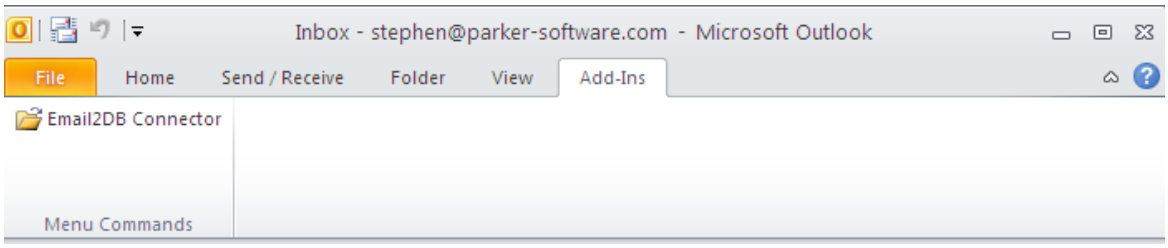
The Email2DB Client includes an add-in for Microsoft Outlook that will automatically read new messages from selected Outlook folders and transfer them to the Email2DB Client. Click the **Outlook Integration** button to configure this.

See: Integrating With Microsoft Outlook

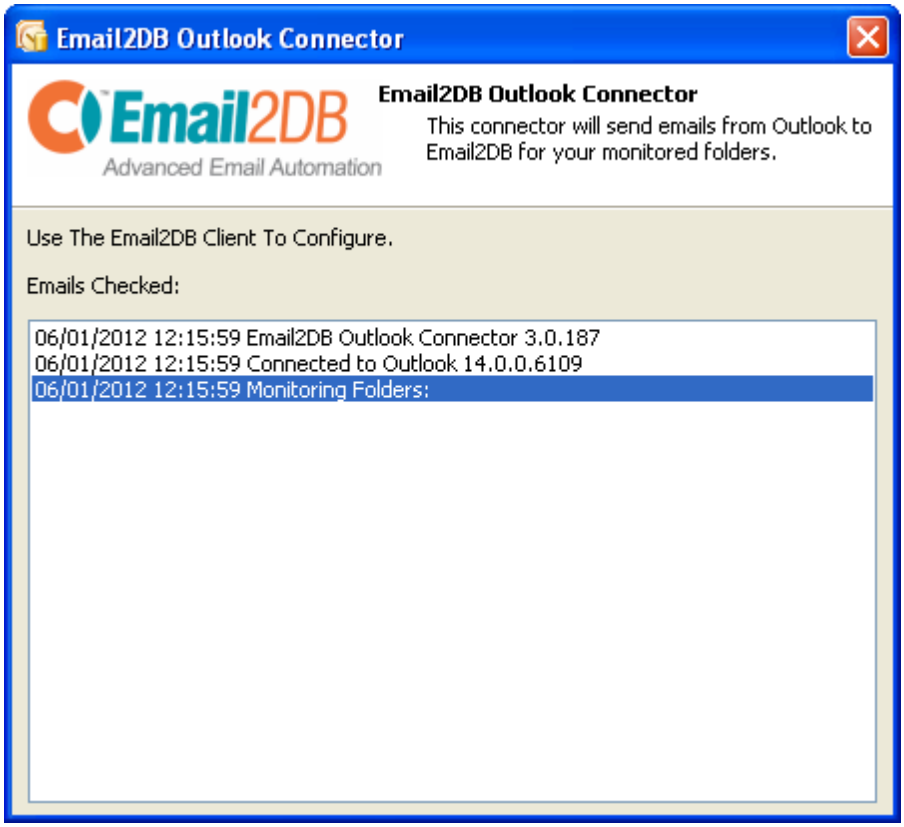
## 12.1 Integrating With Microsoft Outlook

Included with the Email2DB Client is an add-in for Outlook that acts as a 'bridge' between Outlook and the Email2DB Client. This add-in is installed automatically when the Email2DB Client is installed.

If you restart Outlook after installing Email2DB you will see a new option on the **Add-Ins** tab:




Select **Email2DB Connector** to view the **Email2DB Outlook Connector**

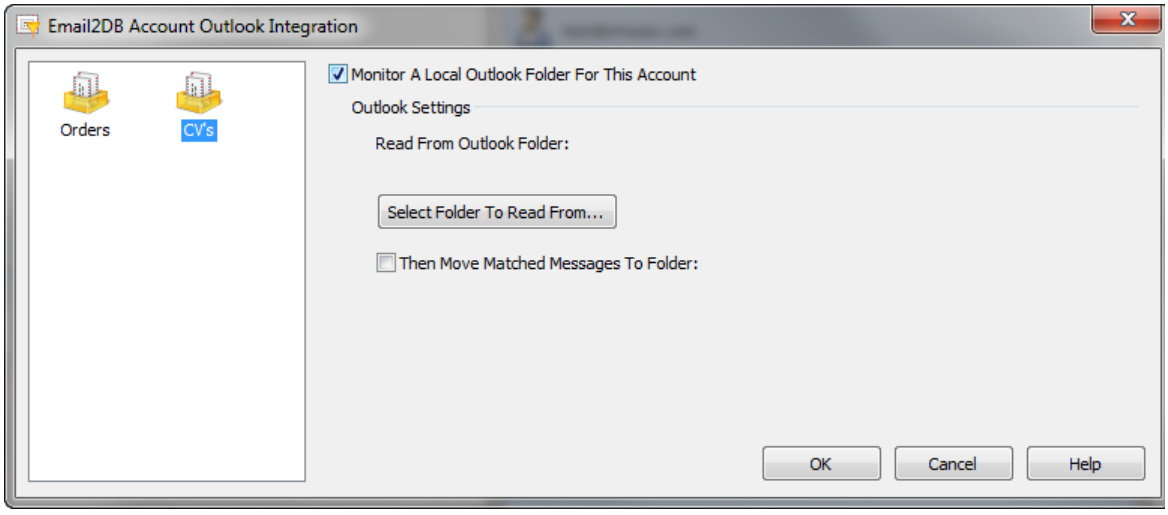



The Email2DB Outlook Connector checks for new email messages in Outlook every xx minutes (this interval is defined on the Email2DB Account Settings). Any new email messages found are passed to the Email2DB Client for processing. Email2DB can read both Mail Items and Post Items from personal and public folders.

Email messages are never altered in Outlook or deleted.

The Outlook Connector is configured using the Email2DB Client.

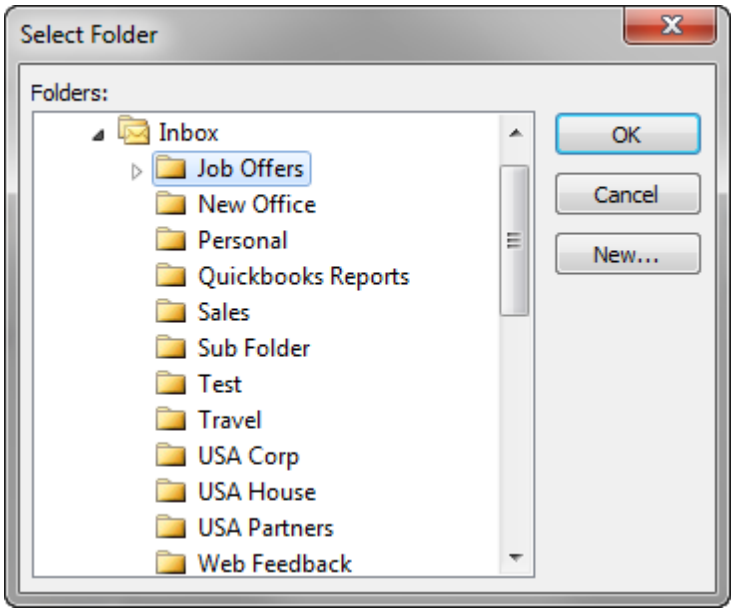
Click button **Outlook Integration** to configure the Outlook Add-In.




Each Email2DB Account is shown. Select an **Account** and enable the **Monitor A Local Outlook Folder For This Account** option.

Click **Select Folder To Read From** button.

Your Outlook folders will be shown.




Select the **Folder** to monitor for the selected Email2DB Account.

You can choose to move processed messages to a different folder in Outlook once they have been sent to Email2DB. Click the **Then Move Matched Messages To Folder** option and select the folder to move the messages to.

Click OK to save the settings.

The Email2DB Outlook Add-In will then monitor the selected folders and move messages to the Email2DB Client automatically. The Email2DB Client will then send the messages to the Email2DB Server for processing.

*Note: The Email2DB Outlook Add-In is a 32 bit application. It will only work on 32 bit Outlook. A 64 bit version is currently in development.*

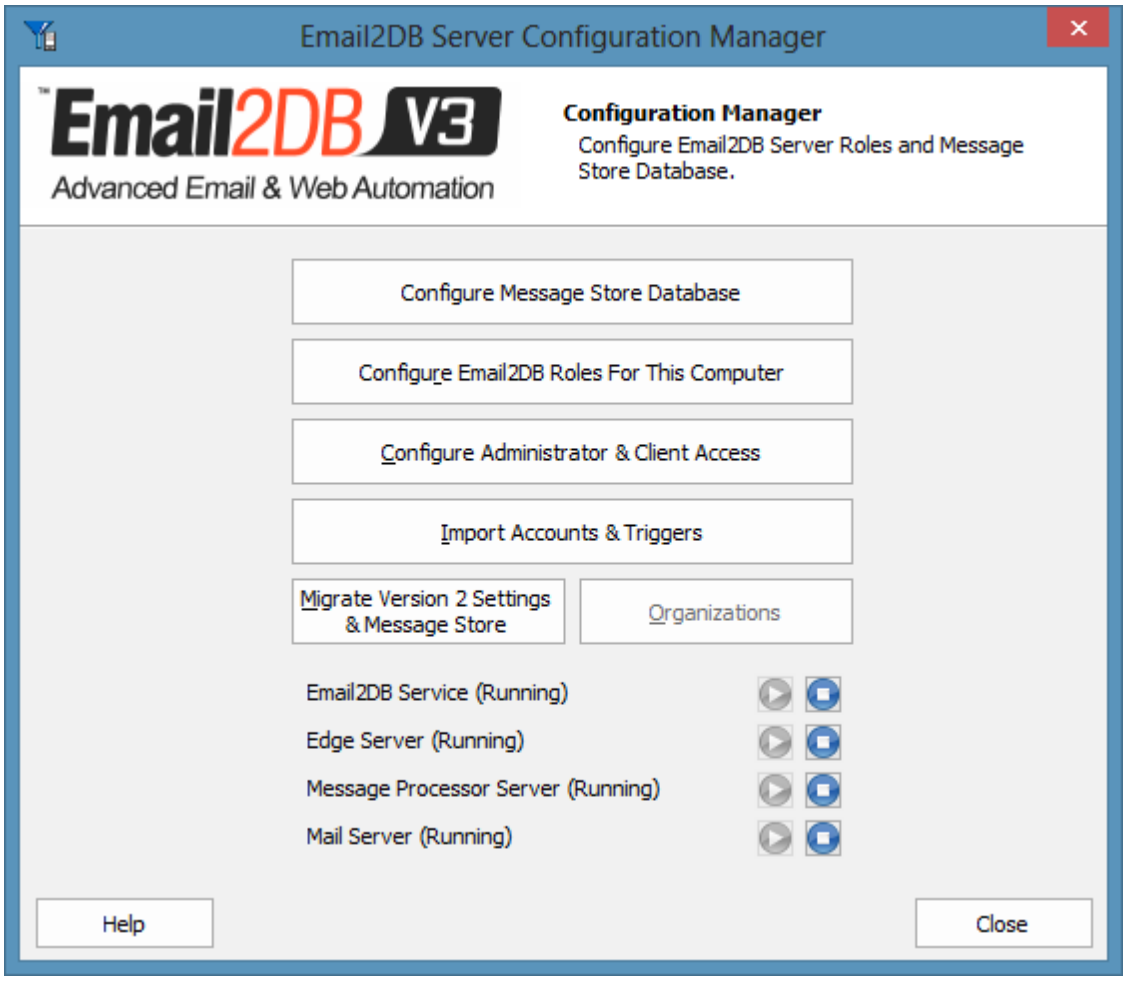
See Also: Outlook PST Export Wizard

# Part XIII

# 13 Email2DB Server Configuration

Included with the Email2DB Server installation is the **Email2DB Server Configuration Manager**. This can be accessed from your **Start** menu - **Email2DB Version 3** folder, or within the Email2DB Administrator **File** menu.

The Email2DB Server Configuration Manager can only be run on the Email2DB Server computer - not a remote workstation.




The Email2DB Server Configuration Manager is used to perform the following tasks:

## Configuring The Message Store Database

This option is used to define what sort of database is used for the Email2DB Message Store and Metadata.

## Configuring Email2DB Roles

This option is used to define what roles this Email2DB node is used for.

## Configuring Administrator & Client Access

This option is used to define the HTTPS port number that the Email2DB Administrator and Remote Clients connect to the Email2DB Server on.

## Importing Accounts & Triggers

This option is used to import Accounts & Triggers from another Email2DB MetaData database. This option can be used to import from Email2DB Version 2 & 3.

### Migrating Email2DB Version 2 Settings & Message Store

This option is used to transfer Email2DB Version 2.x to Email2DB Version 3.

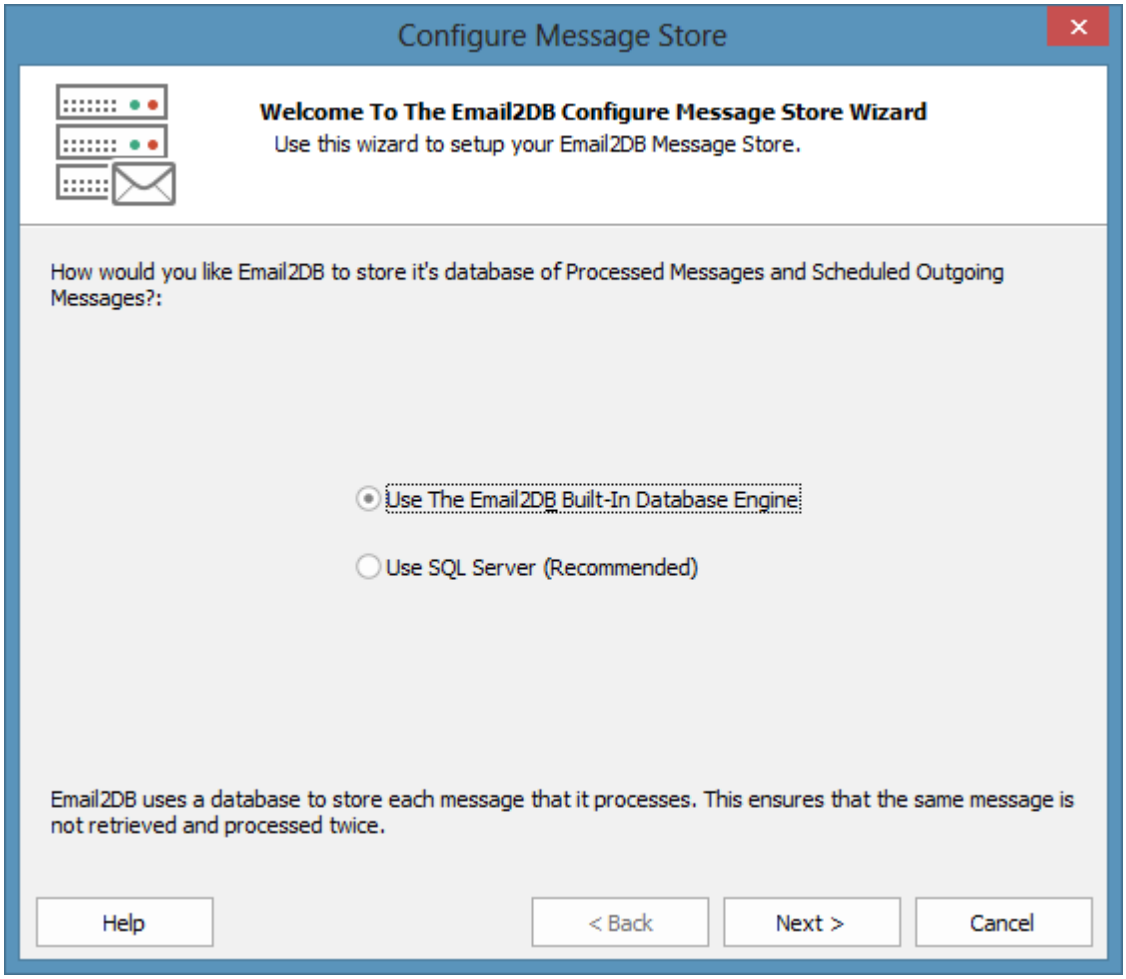
### Message Store Database Maintenance

Used to clean & compress the Email2DB Message Store database when using the built-in database type.

## 13.1 Configuring The Message Store Database

Email2DB can store processed messages either in its own database or in an external SQL Server database.

You can change the format of the Message Store Database by clicking **Configure Message Store Database** from the **Email2DB Server Configuration Manager**. The Email2DB Server will need to be stopped before running this option.




### Using the Email2DB Built-In Database Engine

Email2DB can use its own built-in database engine to store processed messages. This requires no external resources on your PC. It is ideal for small to medium sized businesses.

The built-in Message Store Database is located in the C:\Documents and Settings\All Users \Application Data\Parker Software\Email2DB\folder. It is called **MessageStore.db3**. You can delete this file, provided the Email2DB Administrator and Server are not running. Email2DB will re-create it when it next starts. The built-in database supports unlimited records, however we recommend using the built-in database only for a message store of up to 10GB.

### Using An External Database Server

Email2DB can also use an external SQL Server database for the Message Store. This will provide better performance and allow a Message Store of any size. The following database servers can be used:

- SQL Server 2008/2012
- SQL Server 2008/2012 Express

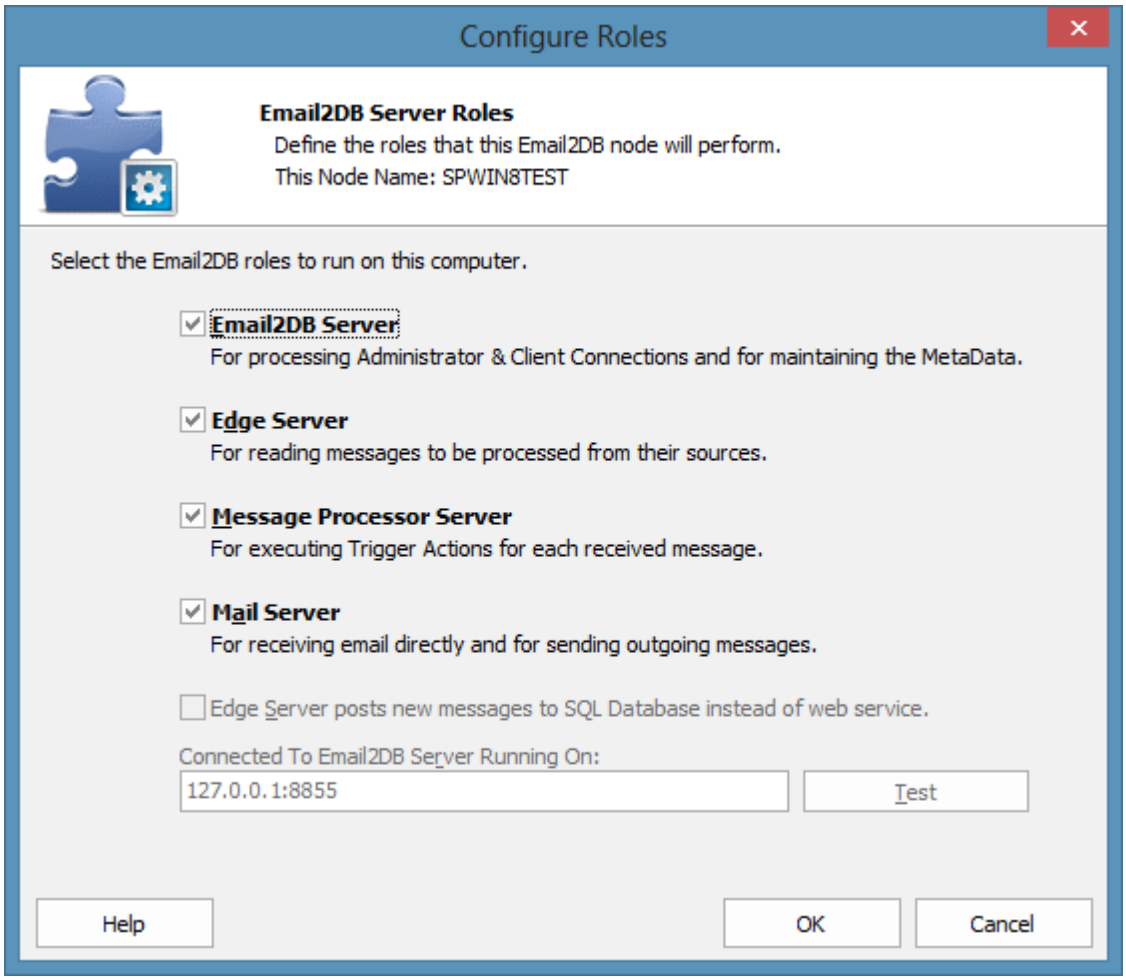
Using an external database is recommended for larger businesses or if you will be processing many thousands of messages. The SQL Database that you plan to use will need to be installed and running before you select this option.

For best performance and ease of setup we recommend using SQL Server 2008/2012 or Express on the same computer as Email2DB.

## 13.2 Configuring Server Roles

This topic applies to the Data Center Edition only.

You can change the Server Role of the Email2DB node by clicking **Configure Roles For This Computer** from the **Email2DB Server Configuration Manager**. The Email2DB Server will need to be stopped before running this option.




The Email2DB Data Center Edition enables you to run Email2DB in a multi-server environment. Connected to an Email2DB Server are any number of 'node' servers. Each node can be configured to run one or more Email2DB Services.

Depending on your requirements you can add additional nodes to distribute the workload.

Select **Email2DB Server** if this computer will act as the main Email2DB Server. At least one server must be configured this way.

Select the additional services that this computer will run:

- **Edge Server** - which reads messages from their sources.
- **Message Processor Server** - which processes the messages read by the Edge Server.
- **Mail Server** - which receives email directly and sends outgoing messages.

If the Email2DB Server is not enabled then you must enter the IP address of the Email2DB Server that this node will connect to. Click the **Test** button to verify the connection.

Click **OK** to configure the Email2DB node. The relevant services will then be installed and started.

## Edge Server & Message Processor Server Message Store Access

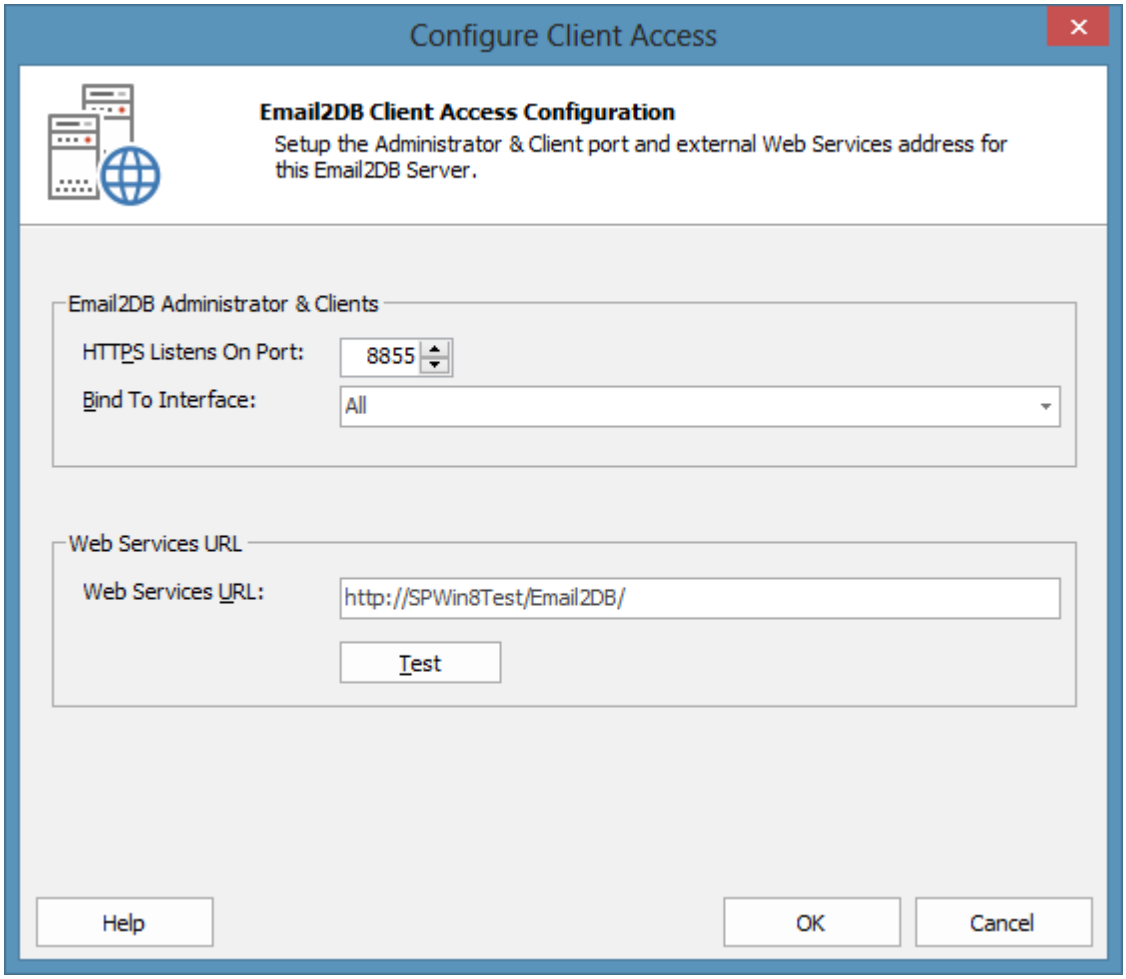
If you select to run an Edge Server on a separate computer from the Email2DB Server, Email2DB can post new messages received either via the Web Services or directly into the SQL Server Message Store. Select **Edge Server posts new messages to SQL Database instead of web service** option to enable this. This will provide the best performance, however your Message Store SQL Server database must be accessible from this computer. If you use the Web Services then the Edge Server does not need Message Store Access.

You can only run the Message Processor Server on remote computers if you have configured the Email2DB Message Store to use SQL Server. The SQL Server database used for the Message Store must be accessible on the computer running the Message Processor Server.

## 13.3 Configuring Administrator & Client Access

The Email2DB Administrator and Email2DB Client applications connect to the Email2DB Server using a secure HTTPS connection. This option is used to change the port number that is used for these connections.

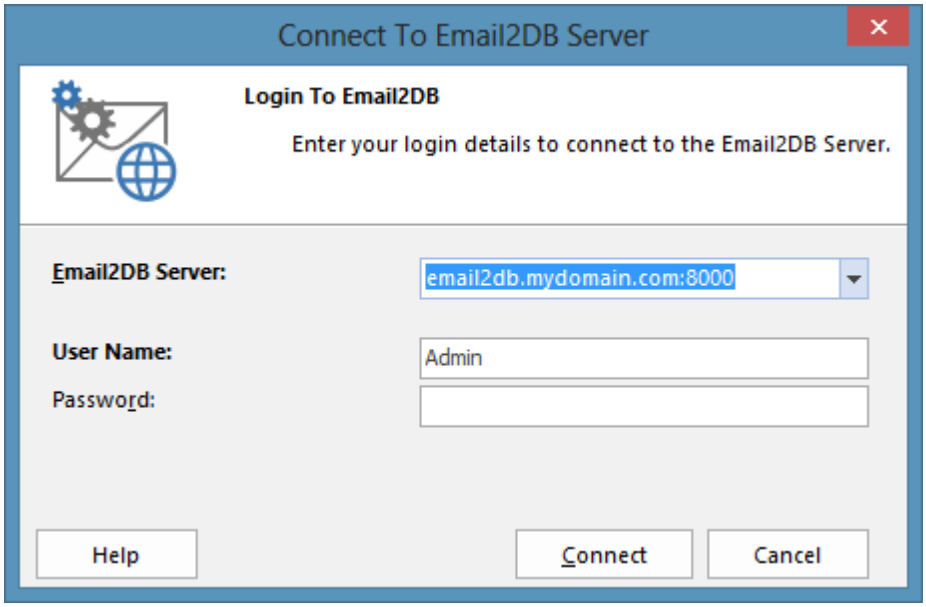
You can change the Client Access Configuration the Email2DB node by clicking **Configure Administrator & Client Access** from the **Email2DB Server Configuration Manager**. The Email2DB Server will need to be stopped before running this option.




**HTTPS Listens On Port**

Specify the port number that the Email2DB should listen on for Administrator & Client connections. The default port number is **8855**. You would normally not need to change this port unless another application is already using it.

If you change the port then users of the Email2DB Administrator & Client will need to specify it in the **Email2DB Server** address when they connect:




Here we are connecting to Email2DB running on 'email2db.mydomain.com' where the port has been changed to 8000.

**Bind To Interface**

By default the Email2DB server listens for client connections on all network interfaces. You can however bind to a specific network interface by selecting the appropriate network card. This can be useful in machines with multiple network interfaces (multihomed).

**Web Services URL**

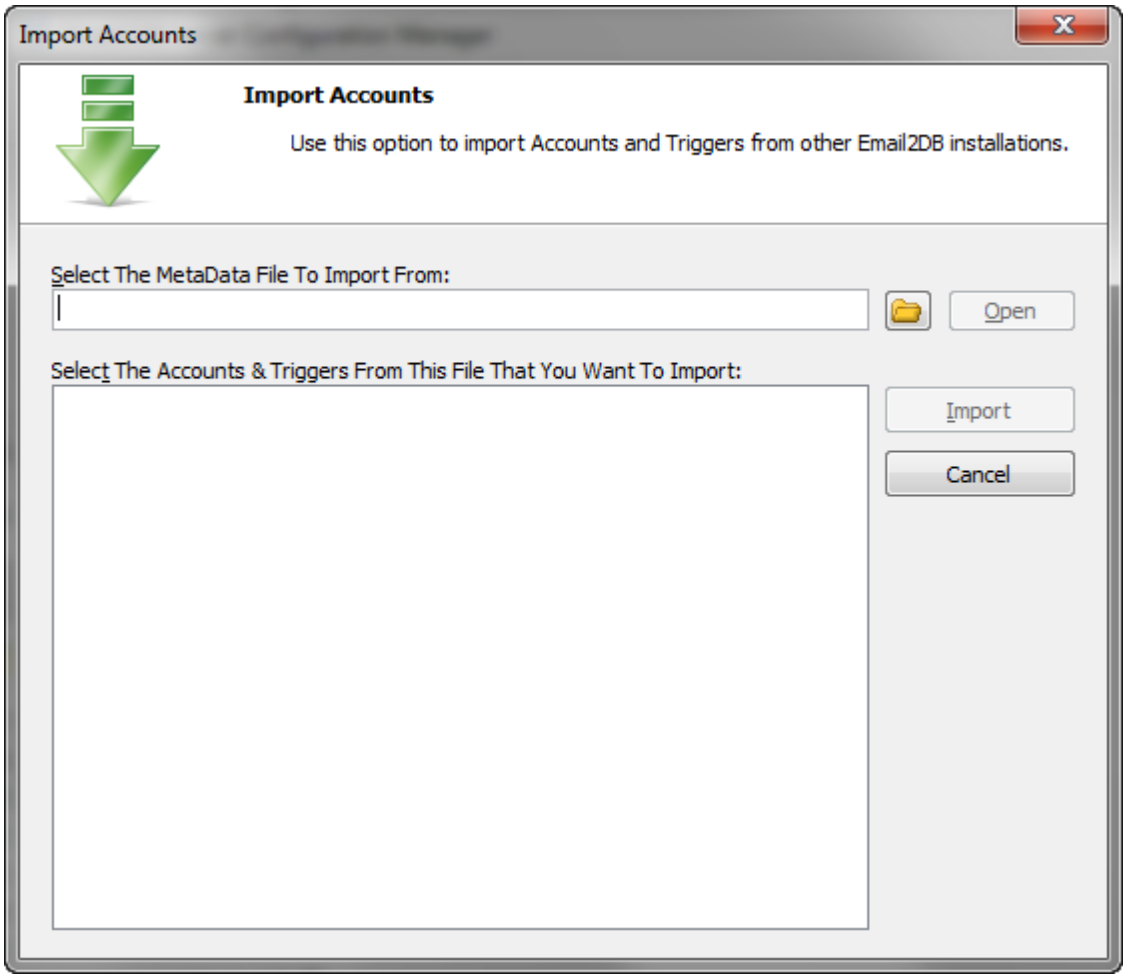
This is the external address for the Email2DB Web Services. During installation the Email2DB Web Services are installed under your default web site in IIS - in a virtual directory called 'Email2DB'. This entry is the PUBLIC address for this virtual directory. If you will be using the Email2DB Administrator or the Email2DB Client over the Internet then the public address for the Email2DB Web Services must be configured correctly.

Click OK to save the settings. You can then restart the Email2DB Server.

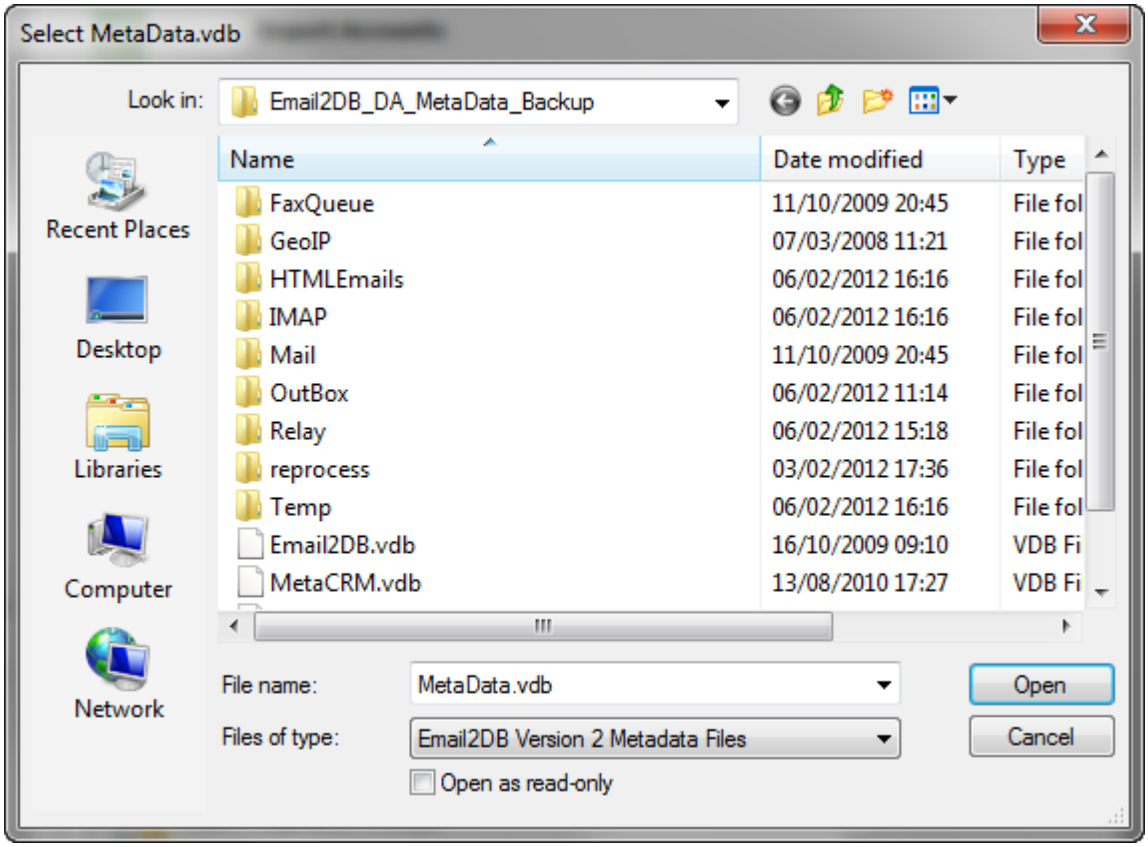
## 13.4 Importing Accounts & Triggers

You can import Email2DB Accounts & Triggers from another Email2DB MetaData file. This includes Email2DB Version 2 MetaData files.

This option can be found by clicking **Import Accounts & Triggers** from the **Email2DB Server Configuration Manager**. The Email2DB Server will need to be stopped before running this option.




Click the **Open Folder** button to select an Email2DB MetaData file. You can import from both Email2DB Version 2 & Version 3 files.

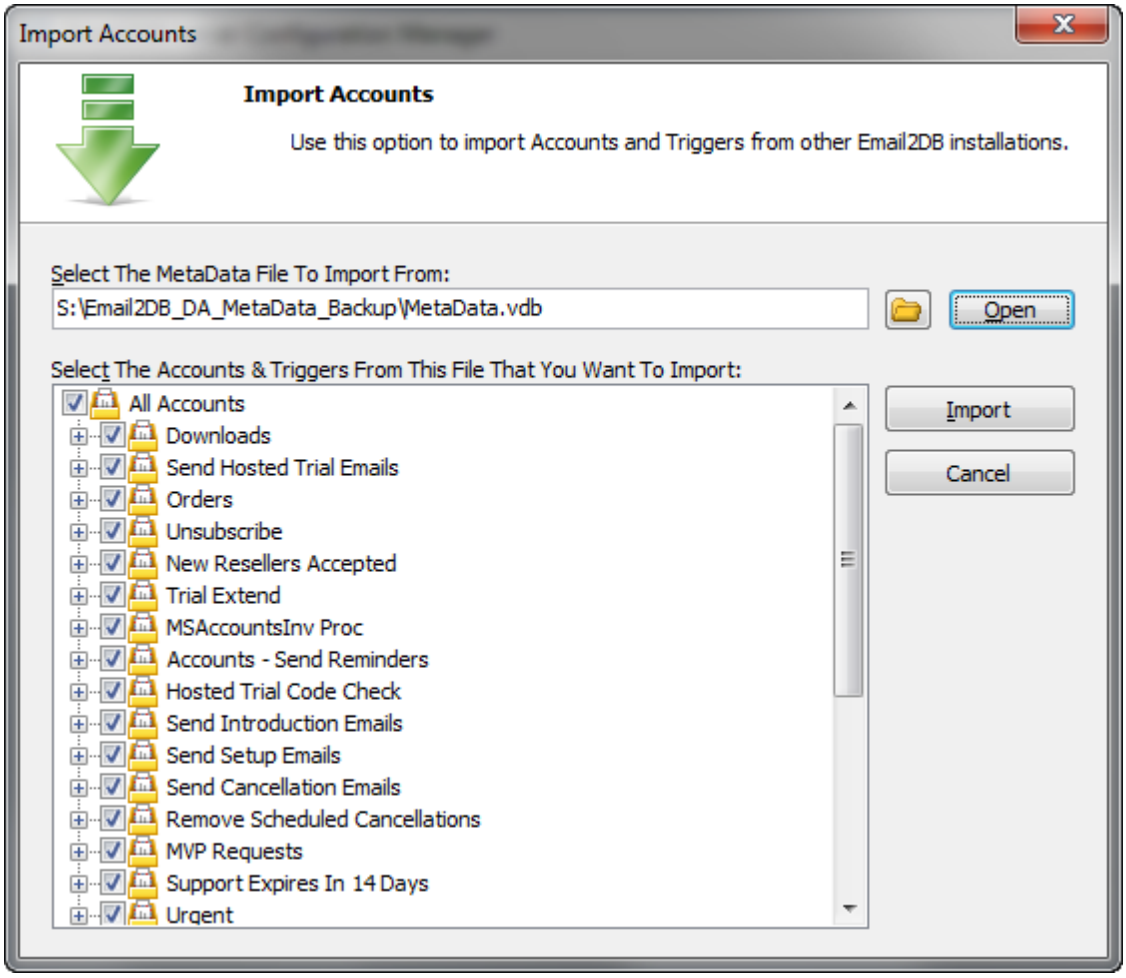



For Email2DB Version 2 - select 'Email2DB Version 2 Metadata Files' from the **Files of type** selection. Then navigate to the MetaData file you want to import from.

*Note: For Email2DB Version 2 this will be called 'MetaData.vdb'. For Email2DB Version 3 it will be called 'MetaData.db3'.*

Click the **Open** button once you have selected the MetaData file.

The Accounts & Triggers will then be read from the selected MetaData file.



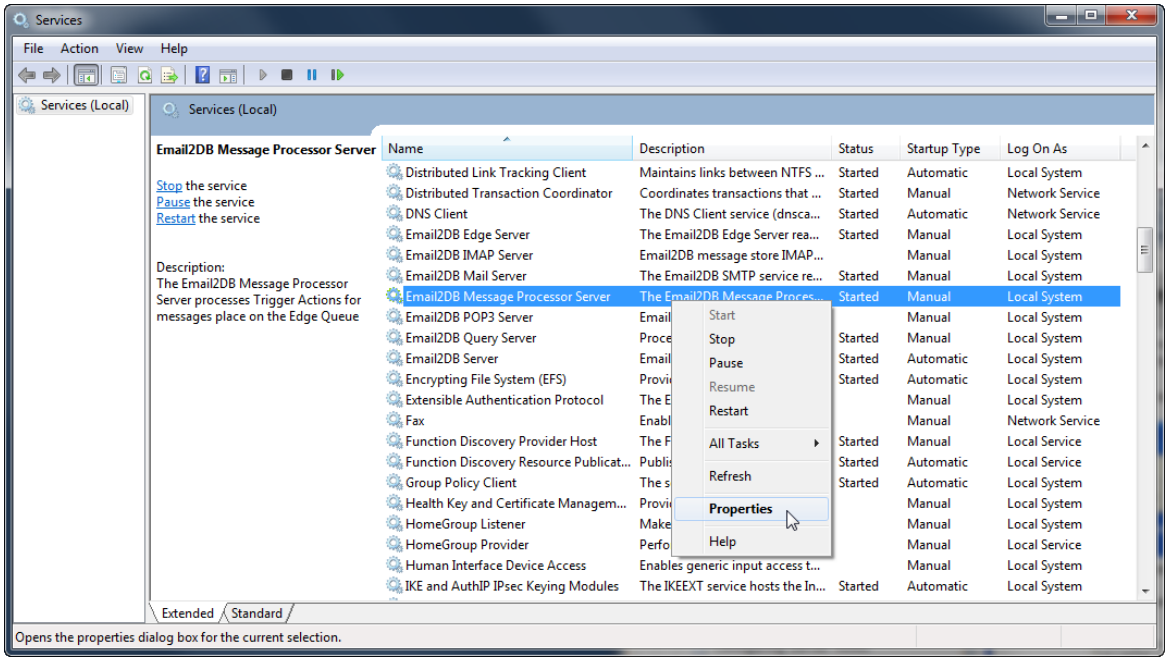

Select the Accounts & Triggers you want to import and click the **Import** button.

The selected Accounts & Triggers will then be added as new Accounts & Triggers to your current Email2DB MetaData file. You will need to restart Email2DB for these to show in your Email2DB Administrator.

## 13.5 Running The Email2DB Service Under A Different User

By default the Email2DB Services run under the SYSTEM account. The SYSTEM account does not have access to any network resources (Printers, Network Drives, Remote Shares etc). Therefore if you want Email2DB to update files, save attachments, print reports etc. to network devices then you must run the Email2DB Message Processor service under a different user.

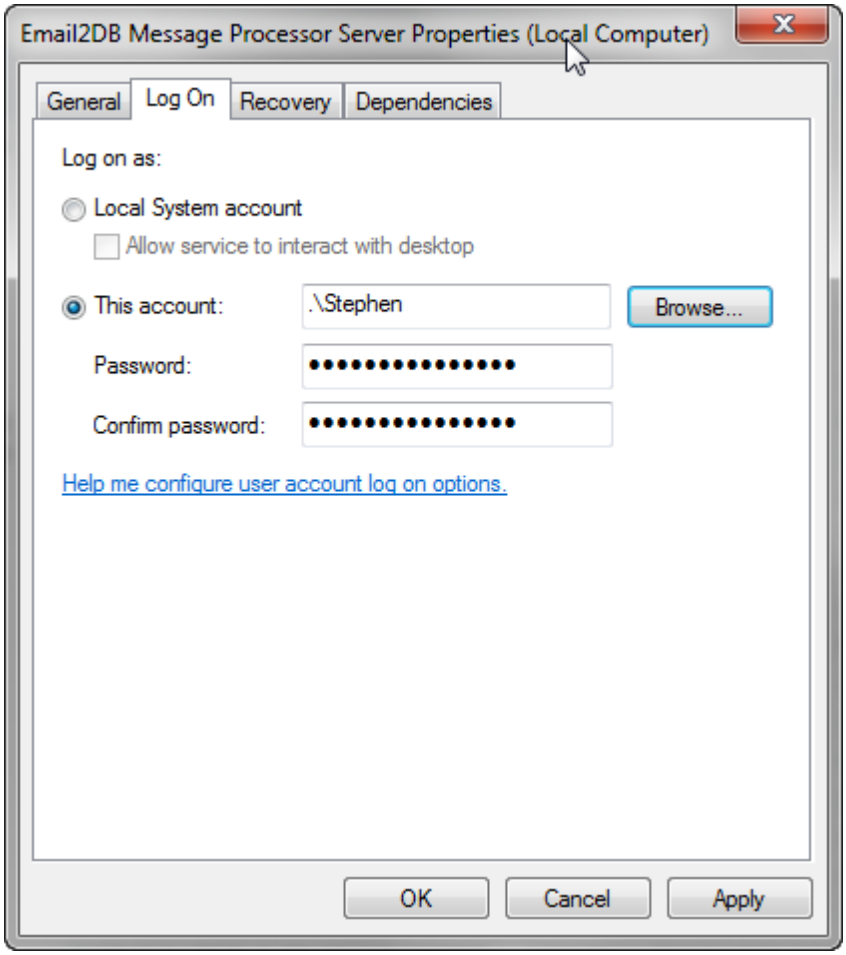
To do this, first stop the Email2DB Services using the Email2DB Server Configuration Manager. Then open the **Services** applet in your **Control Panel** - **Administrative Tools** folder.




Select the **Email2DB Message Processor Server** Service.

On the **Log On** tab un-select Local System Account and select **This Account**

Select a local user account that has access to the network resources you want Email2DB to be able to access. Enter the password and confirm it.




Click **OK**.

Restart the Email2DB Service.

## 13.6 Backing Up Your Settings

### If Using The Built-In Database For The Message Store

Email2DB stores all its settings in the folder C:\Documents and Settings\All Users\Application Data \Parker Software\Email2DB\ or C:\ProgramData\Parker Software\Email2DB on Windows 7/8.

You can open this folder by clicking the **Settings Path** button on the **File** menu.

All Accounts & Triggers are stored in the file **MetaData.db3**

Mail server settings are stored in **MetaMailServer.db3**

Global settings are stored in the file **SETTINGS.XML**.

The message store is stored in the file **MessageStore.db3**. If you delete this file, then Email2DB will re-create a new one when it is next started (you must close Email2DB before deleting it).

To take a complete backup of all your Email2DB settings, simply backup the contents of this folder.

### If using SQL Server For The Message Store

All Accounts & Triggers are stored in a database called **Email2DBV3_Metadata** in the SQL Server database selected when you setup the Message Store.

The message store is stored in a database called **Email2DBV3**.

You must use SQL Server management tools to backup your data.

Mail server settings are stored in **MetaMailServer.db3**

Global settings are stored in the file **SETTINGS.XML**.

### Transferring Settings To Another PC

If you want to setup some triggers to be transferred to another computer, you simply need to copy the MetaData.db3 file to the same location on the other PC.

# Part

# 14 Examples

Updating A Database, Forwarding Confirmation To The Customer and Printing a Report

In this example we will show how to parse an email and update an 'Orders' database... then send a follow up email to the person who placed the order and print a report.

Suppose we receive automated emails from the address 'orders@mydomain.com' that contain the following:

**Subject**:

```
Order received for product "Widgets Version 1"
```

**Body:**

```
Name    = Sharlene Baker
Company = Parker Software Inc
Street  = New Broad Street
ZIP     = 32814
City    = Orlando
Country = United States
State   = Florida
Phone   = 800 680 7712
Fax     = 44 1782 839682
Email   = sharlene@parker-software.com
```

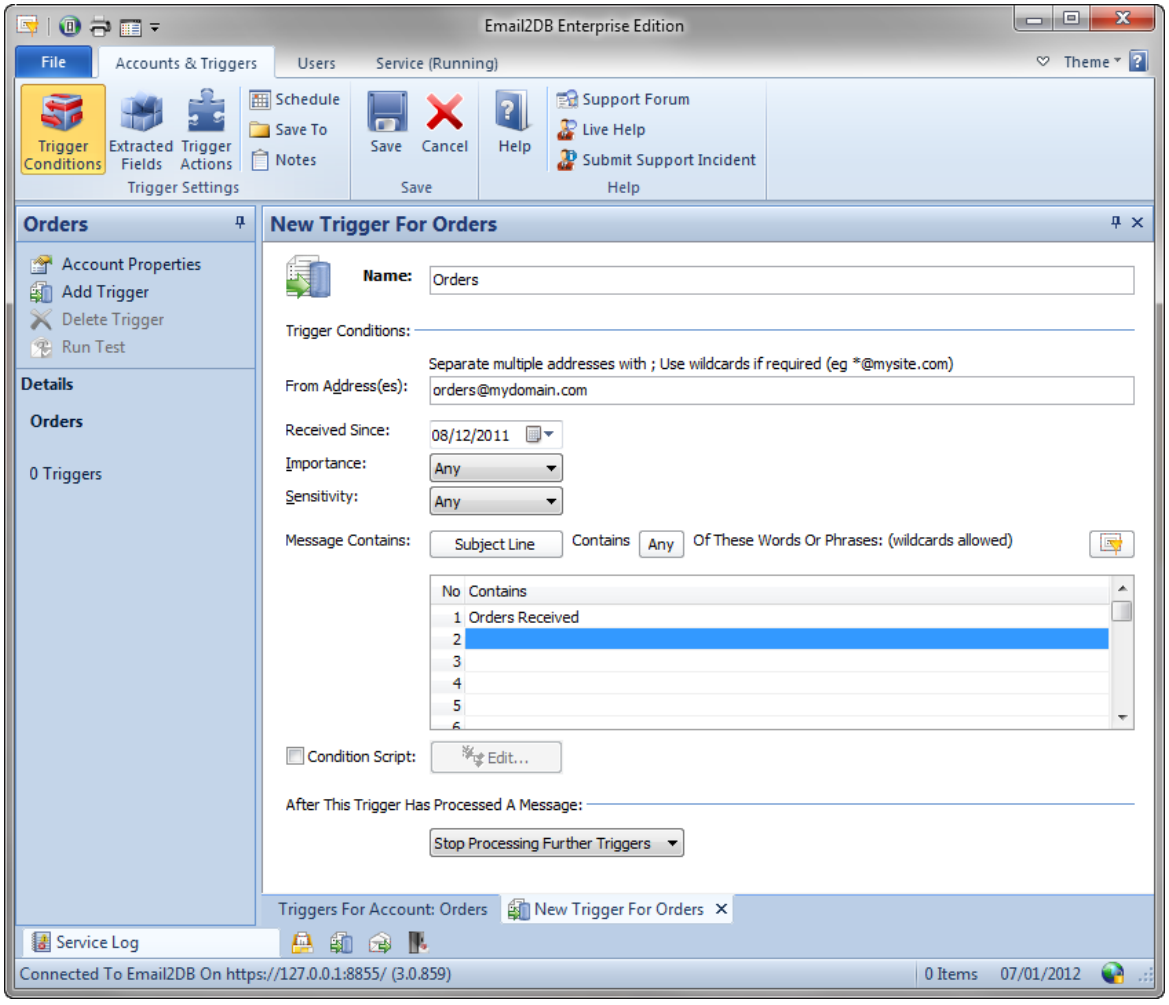
The first thing we need to do is create an Account in Email2DB. Click **Add Account** button.

Enter the Account **Name**. Click the **Get Via Email** button.

On the **POP3** tab, enable the **Read Emails From A POP3 Server** and specify the mail server and mail account details for the POP3 account that receives the above email's. Click **Save** to save the Account. When you save a new Account, Email2DB will ask if you want to create a new trigger now - click Yes.

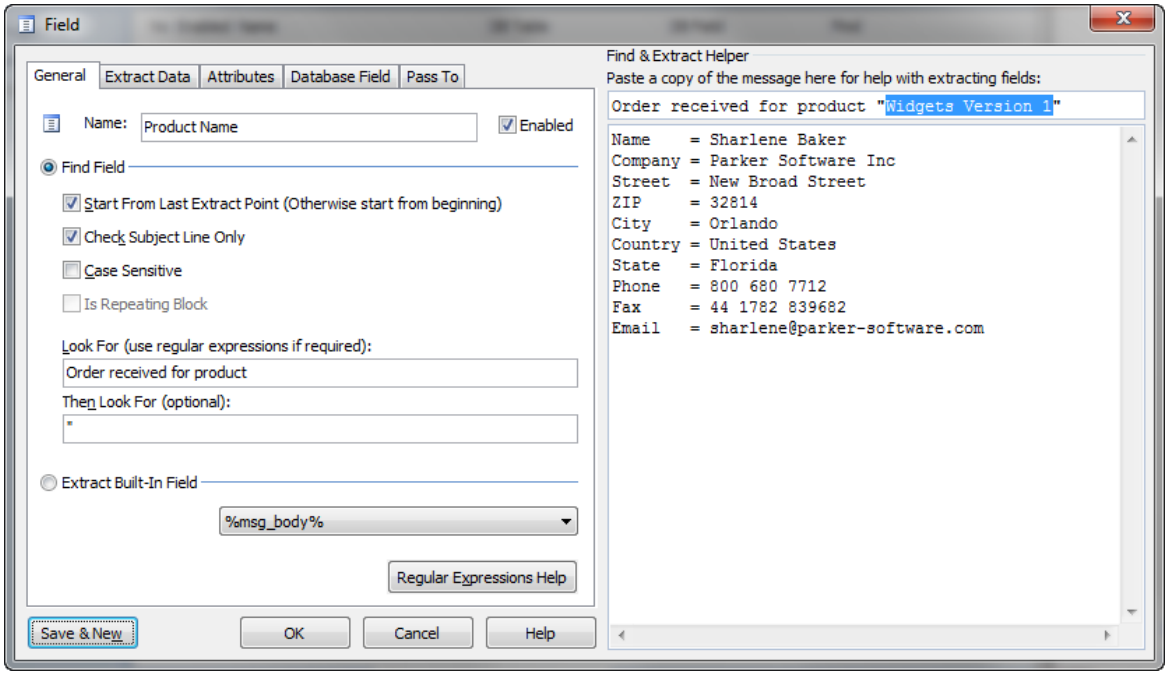
Next we need to create a 'Trigger' so that Email2DB can act on just the Order email's received through this account and ignore the rest.

Since the orders are always received from the same email address (orders@mydomain.com) we can enter it in the **From Address** entry. The Subject line also always contains the words 'Order Received' so we can enter this in the **Message Contains** entry.




We can now proceed to extract fields from the email message. Click the **Extracted Fields** button then click the **Add field** button. When you add your first field it is a good idea to paste a sample email text and subject into the Find & Extract helper field. As you create your extraction fields, Email2DB will highlight what will be extracted in the Find & Extract helper box. For this example, paste the subject and body above into the subject & body fields of the **Find & Extract Helper**.

The first field we want to extract is the Product Name. This is part of the subject line. When Email2DB starts to extract fields it always starts from the 1st character of the subject line before moving to the body text.



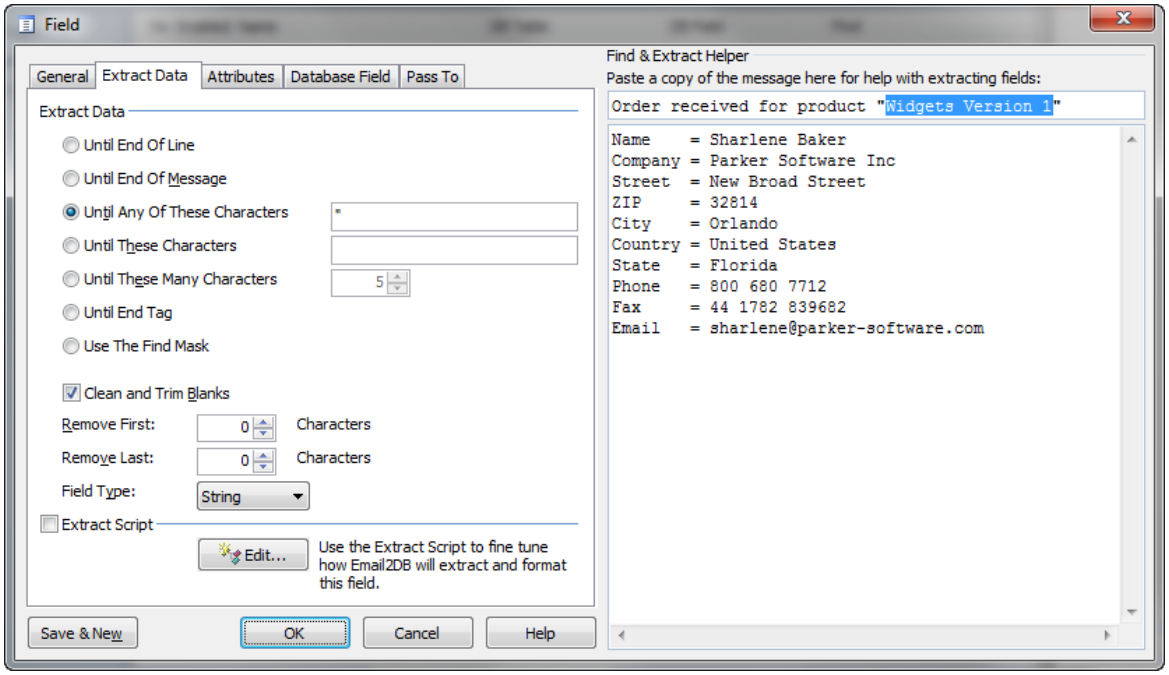

In the **Name** entry enter a name for the field.

The **Start From Last Extract Point** is enabled by default. This tells Email2DB to start searching for the field from the last field position.

Since this field will only appear in the Subject line click the **Check Subject Like Only** option.

In the **Look For** entry enter 'Order received for product'. In the **Then Look For** entry enter " (double quote). This tells Email2DB to search for 'Order received for product' AND THEN the first double quote. The field extraction will start from the next character.

Now click the **Extract Data** tab

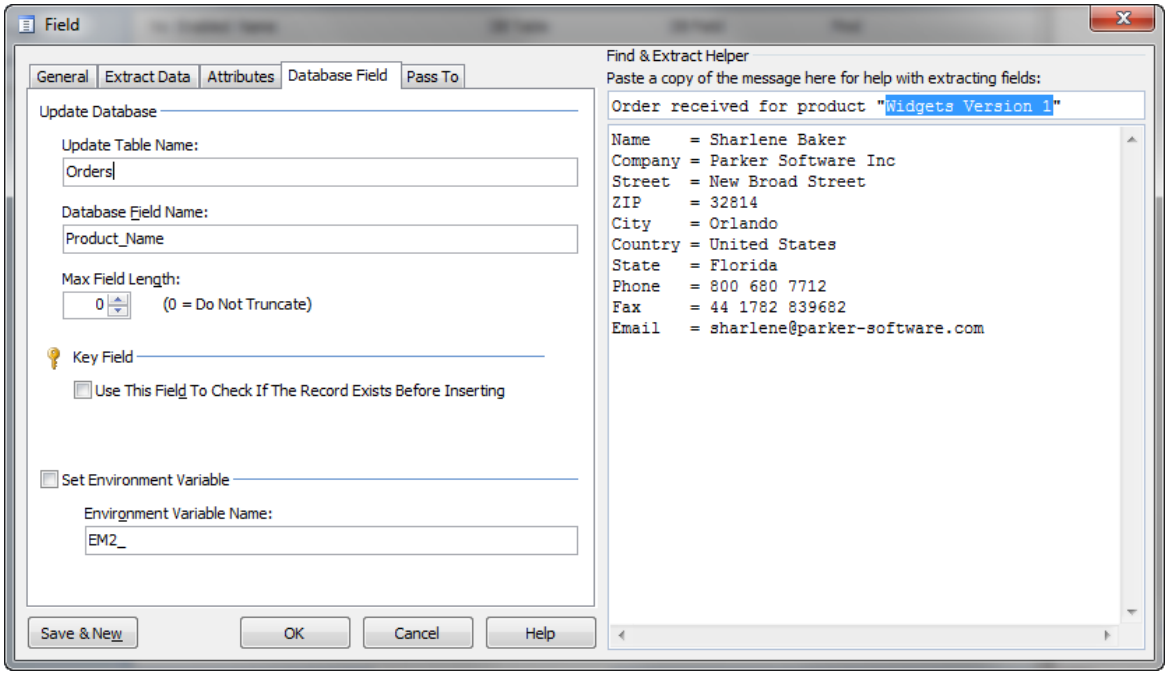


Check the **Until Any Of These Characters** option and enter a double quote in the box.

This tells Email2DB to extract the field up to the next double quote.

Click the **Database Field** tab

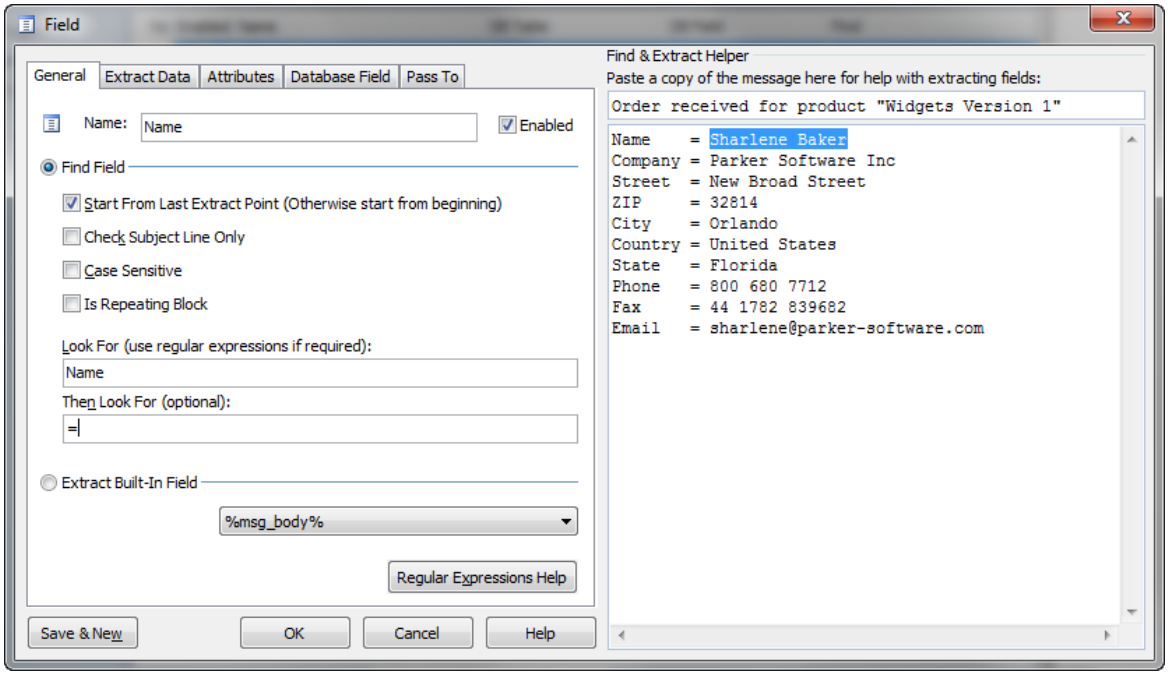
We now need to tell Email2DB the Table and Field name that this field should be inserted into.




Enter 'Orders' for the **Update Table** Name.

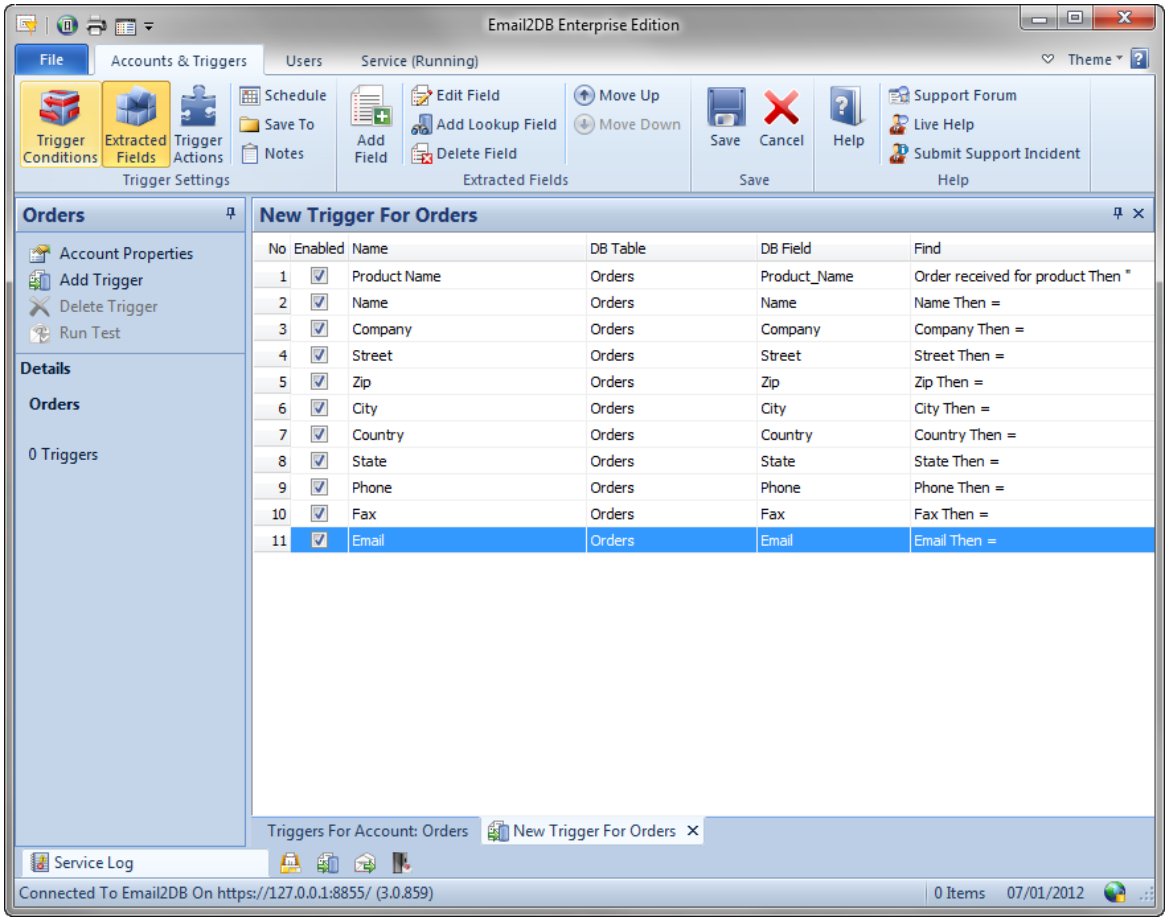
Click **OK** to save the Field.

Click the **Add Field** button to add another field.



The remaining fields are on lines on their own so can extract data **'Until End Of Line'**. We set the **Look For** to the text before the = sign, **Then Look For** to '=' because we don't know how many spaces are between the Look For entry and the start of the field.

Repeat for each of the remaining fields:

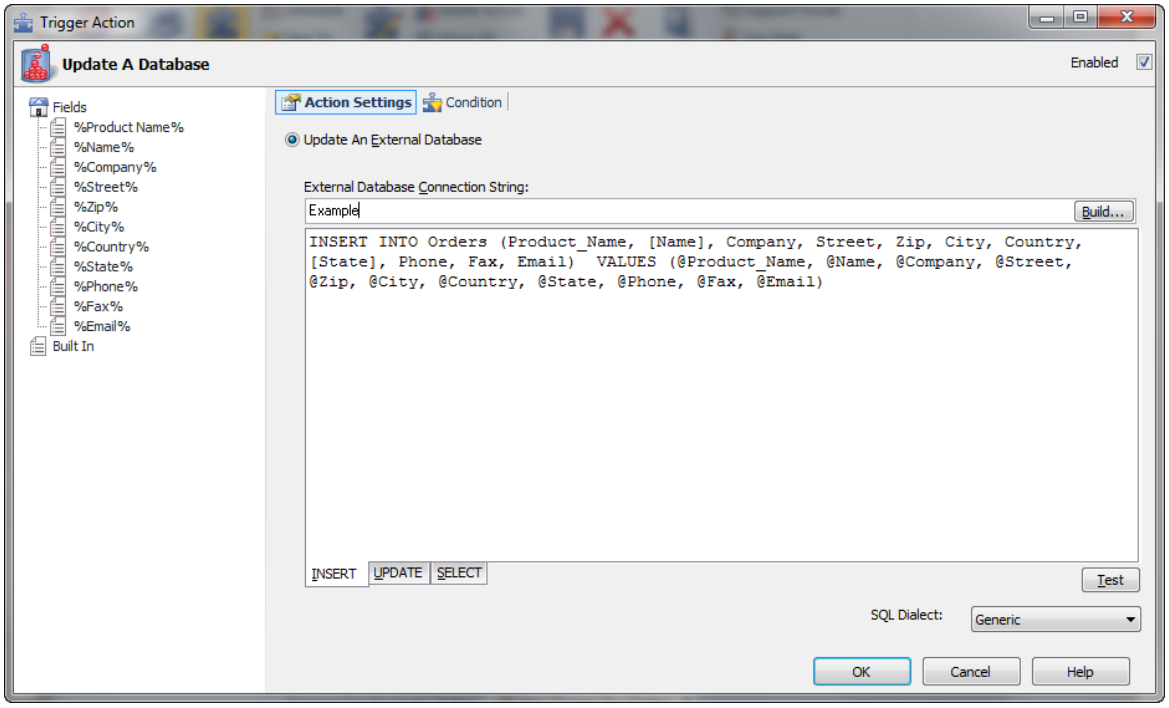



Now click the **Trigger Actions** button.

Here we add all the Actions that we want to perform once Email2DB has received the message and extracted all our fields.

Click the **Update A Database - Direct** Action and drag it onto the Actions list.

The Update A Database Action form will be displayed:



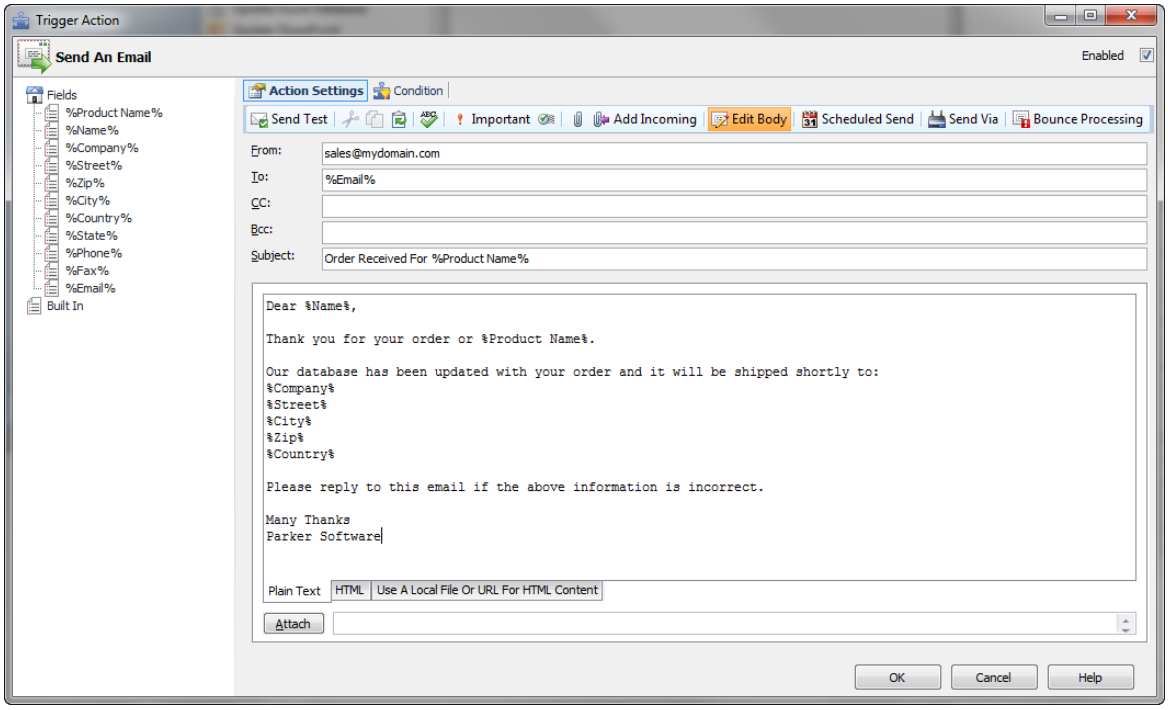

You will see that Email2DB has automatically created an SQL Insert statement using the Table and Field names specified. If we have specified multiple table names Email2DB would have created separate Insert statements for each.

Click the **Build** button to build a connection string for any data source available on the PC. For now just enter 'Example' in the connection string and click **OK**.

We now want to send the customer who placed the order a confirmation email.

Select the **Send Email** tab Action and drag it onto the Actions list. The Send Email Action form will be displayed.

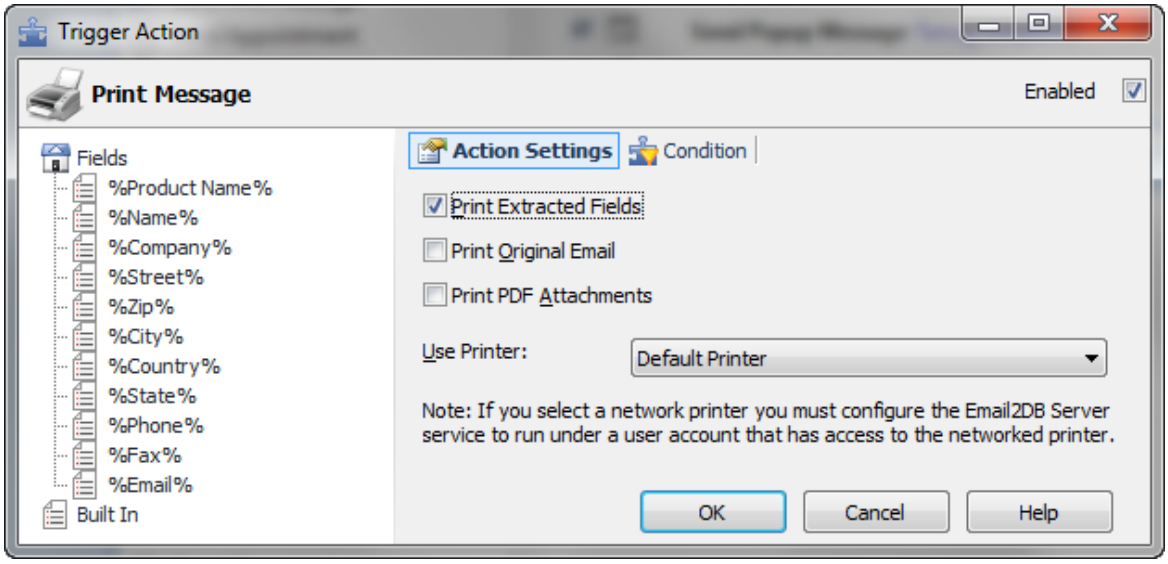
In the **To** entry enter **%Email%** - this tells Email2DB to replace the send to address at run time with the value of the extracted field called 'Email'. You can also drag the field from the Fields tree.




Enter the fields as above. Notice how we can use %field% replacements in any of the Action properties.

Click **OK** to save the email. You can create any number of additional outgoing messages. You can also create 'scheduled emails'. These are outgoing emails that will be sent at some point in the future. You could use this to send a 'how was your order' type message.

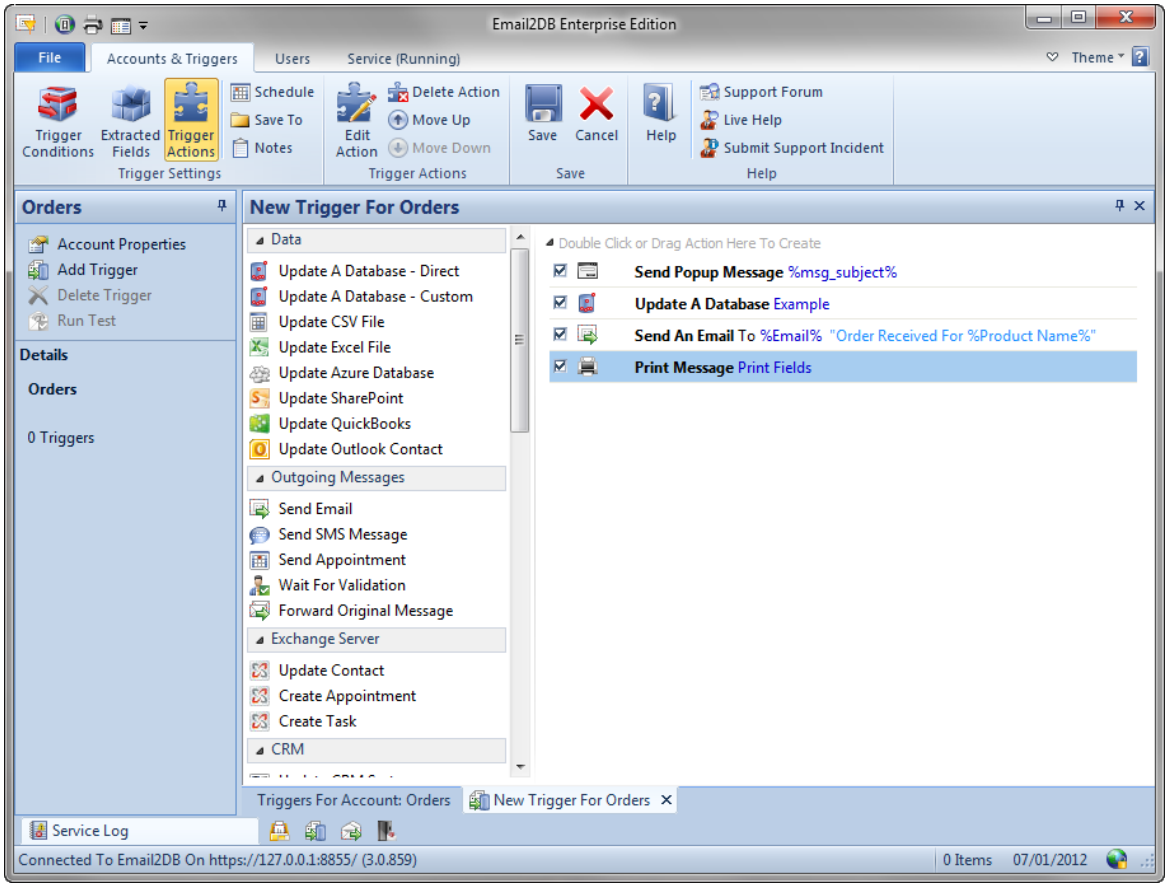
We may also want to print the email as it arrives. Drag the Print Message Action to the Actions list.




Click the **Print Extracted Fields** check box.

Email2DB will then print a report of fields extracted each time an order is received. Click **OK** to save the Action.

Your Actions list will now look like:




Click **Save** to save the Trigger.

The Trigger will be saved. Email2DB is now ready to start receiving orders.

# Part

# 15 Advanced

Regular expressions can be used when searching and extracting fields from message text.

The regular expression syntax used by Email2DB is quite powerful and can be used to match virtually any type of text pattern.

| Char | Description |
|---|---|
| ^ | Beginning of a string. |
| $ | End of a string. |
| . | Any character. |
| [list] | Any character in list. For example, "[AEIOU]" matches any single uppercase vowel. |
| [^list] | Any character not in list. For example, "[^ ]" matches any character except a space. |
| [A-Z] | Any character between 'A' and 'Z'. For example, "[0-9]" matches any single digit. |
| ? | Repeat previous character zero or one time. For example, "10?" matches "1" and "10". |
| * | Repeat previous character zero or more times. For example, "10*" matches "1", "10", "1000", etc. |
| + | Repeat previous character one or more times. For example, "10+" matches "10", "1000", etc. |
| \ | Escape next character. This is required to any of the special characters that are part of the syntax. For example "\.*\+\\" matches ".*+\". It is also required to encode some special non-printable characters (such as tabs) listed below. |

In addition to the characters listed above, there are seven special characters encoded using the backslash. These are listed below:

```
\a Bell (Chr(7))
\b Backspace (Chr(8))
\f Formfeed (Chr(12))
\n New line (Chr(10), vbLf)
\r Carriage return (Chr(13), vbCr)
\t Horizontal tab (Chr(9), vbTab)
\v Vertical tab (Chr(11))

For example
"^stuff"                ' any string starting with "stuff"
"stuff$"                ' any string ending with "stuff"
"o.d"                   ' "old", "odd", "ord", etc
"o[ld]d"                ' "old" or "odd" only
"o[^l]d"                ' "odd", "ord", but not "old"
"od?"                   ' "o" or "od"
"od*"                   ' "o", "od", "odd"
"od+"                   ' "od", "odd", etc
"[A-Z][a-z]*"           ' any uppercase word
"[0-9]+"                ' any stream of digits
"\."                    ' decimal point (needs escape character)
"[1-9]+[1-9]*"          ' any stream of digits not starting with 0
"[+\-]?[0-9]*[\.]?[0-9]*" ' any number with optional sign and decimal point (needs
two escape characters)
"[A-Z0-9]+ [0-9]+[A-Z]+" ' extracts a UK post code
"[a-zA-Z0-9._-]+@[a-zA-Z0-9_-]+\.[a-zA-Z][a-zA-Z.]*[a-zA-Z]" ' extracts any email
address
```

## Additional Email2DB Regular Expressions

In both the Look For and Then Look For fields you can include a number of control characters in addition to regular expressions:

```
<CR>      Carriage return
<LF>      Line feed
<CRLF>    Carriage return/line feed
<TAB>     Tab
<ESC>     Escape
*         When used on its own the * character finds the next
          non-space or control character.
<xxx>     Where xxx is the ASCII character code
```

This can be useful when searching for data. For example, suppose the text contains:

```
Your serial number is:
1234-5678
```

We want to extract the serial number, so we could look for 'Your serial number is:' and then look for '<CRLF>' - because the serial number is on the next line. Another way of doing the above would be to look for 'Your serial number is:' and then look for '*' - which would effectively look for anything after 'Your serial number is:'.

Email2DB regular expressions are compatible with the Microsoft RegEx Library. For more regular expression examples, please see:

http://msdn.microsoft.com/en-us/library/az24scfc(v=vs.110).aspx

http://regexone.com/

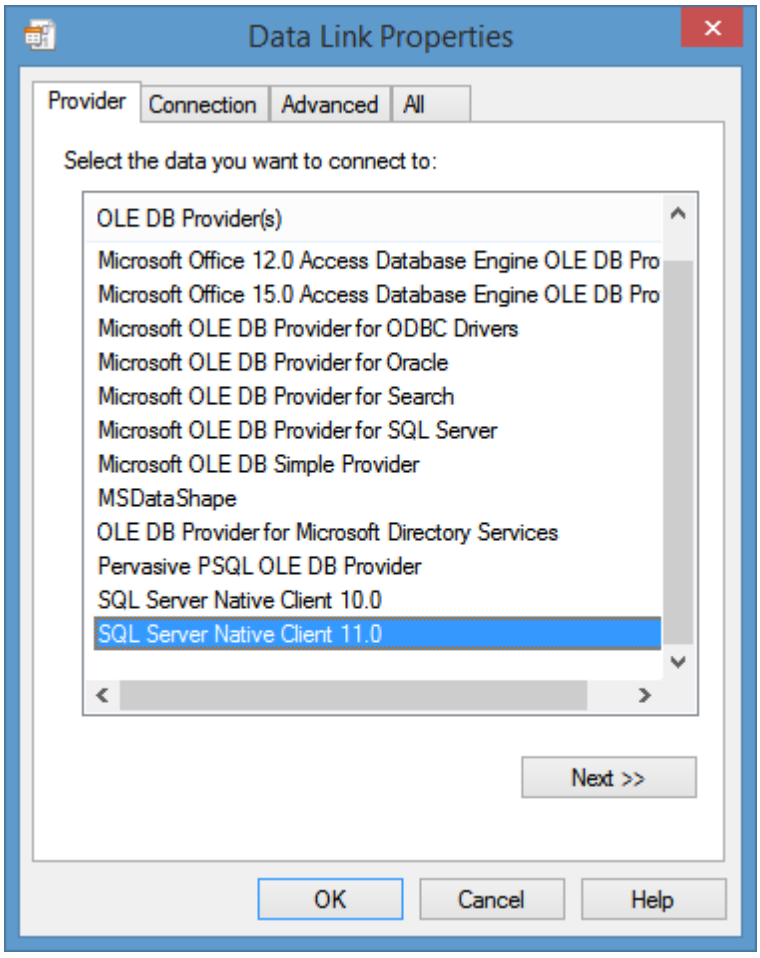
http://en.wikipedia.org/wiki/Regular_expression

## 15.1 Database Connection Strings

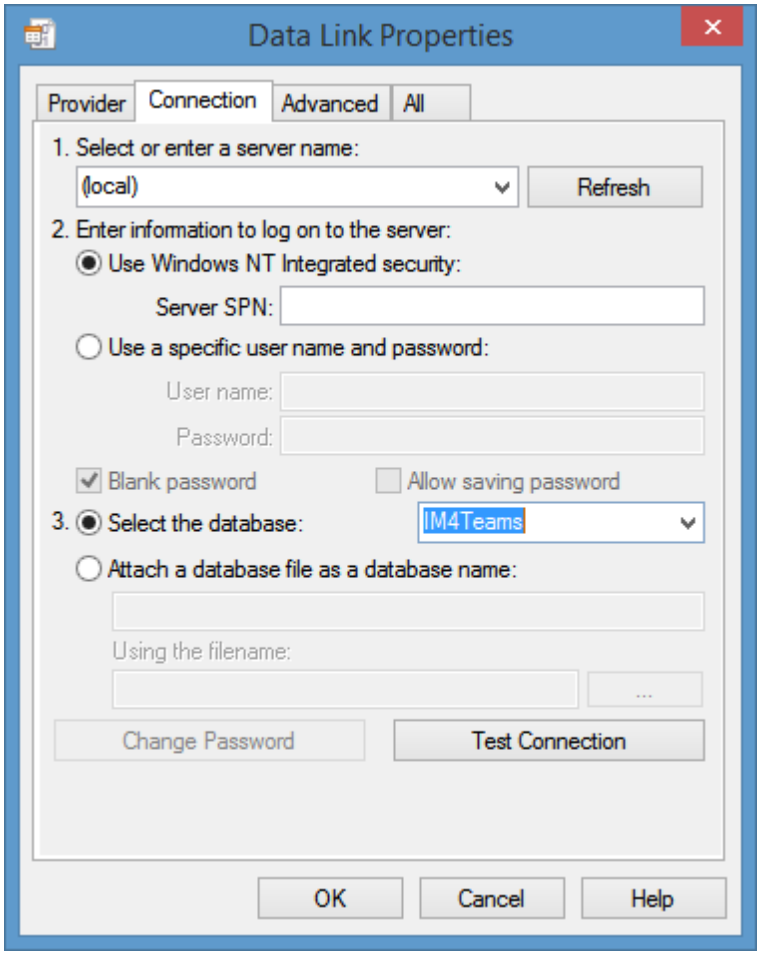
All Email2DB Database Actions require you to enter a 'Connection String'. This provides Email2DB with the information needed to connect to your database.

Connection Strings differ from database type to database type.

You can manually specify the connection string by entering it directly, or you can use the **Build** button to show the **Data Link Properties** form. This is used to build a connection string which is then passed back to the Connection String entry:




The first tab shows all the available database sources on your computer. Select the data source and click **Next**. For example: For SQL Server 2008/2012 select SQL Server Native Client 11.0. If this is not listed you need to install the SQL Server client (this is normally part of the SQL Server setup - however if you are on a remote computer you will need to install the SQL Server client).




You must then select the **server name** for the computer running the database. Click the dropdown menu to view a list.

You then need to provide the authentication settings, either **Windows Integrated Security** or **Use a specific user name and password**. If you do specify a user name/password ensure you enable the **Allow saving password** option - this will ensure the password is passed back to the connection string.

You then need to **Select the database** you wish to connect to.

Click **Test Connection** to verify the database can be opened using the details you have provided.

Click **OK** to save the connection string back to the entry you are using. You can adjust the connection string if required:

```
Provider=SQLNCLI11.1;Integrated Security=SSPI;Persist Security Info=False;User
ID="";Initial Catalog=IM4Teams;Data Source=(local);Initial File Name="";Server
SPN=""
```

## Using Field Replacements Or Constants For Connection Strings

You can use %field% replacements for connection strings. You can also use global constants. Using constants helps when you need to use the same connection string in lots of places. First build your connection string and make sure it works. Then use the Organizations form on the program options form to create a global 'constant' to hold the connection string value. You can then use this constant

in the form %ConstantName% in your connection string entries.

## More Information

For more information about connection strings see:

http://www.connectionstrings.com/ - a reference for connection string formats for most database types.
http://en.wikipedia.org/wiki/Connection_string

Online generator for SQL Server connection string:
http://www.developerfusion.com/tools/sql-connection-string/

## 15.2 Regular Expressions

Regular expressions can be used when searching and extracting fields from message text.

The regular expression syntax used by Email2DB is quite powerful and can be used to match virtually any type of text pattern.

| Char | Description |
|---|---|
| ^ | Beginning of a string. |
| $ | End of a string. |
| . | Any character. |
| [list] | Any character in list. For example, "[AEIOU]" matches any single uppercase vowel. |
| [^list] | Any character not in list. For example, "[^ ]" matches any character except a space. |
| [A-Z] | Any character between 'A' and 'Z'. For example, "[0-9]" matches any single digit. |
| ? | Repeat previous character zero or one time. For example, "10?" matches "1" and "10". |
| * | Repeat previous character zero or more times. For example, "10*" matches "1", "10", "1000", etc. |
| + | Repeat previous character one or more times. For example, "10+" matches "10", "1000", etc. |
| \ | Escape next character. This is required to any of the special characters that are part of the syntax. For example "\.*\+\\" matches ".*+\". It is also required to encode some special non-printable characters (such as tabs) listed below. |

In addition to the characters listed above, there are seven special characters encoded using the backslash. These are listed below:

```
\a Bell (Chr(7))
\b Backspace (Chr(8))
\f Formfeed (Chr(12))
\n New line (Chr(10), vbLf)
\r Carriage return (Chr(13), vbCr)
\t Horizontal tab (Chr(9), vbTab)
\v Vertical tab (Chr(11))

For example
"^stuff"                ' any string starting with "stuff"
"stuff$"                ' any string ending with "stuff"
"o.d"                   ' "old", "odd", "ord", etc
"o[ld]d"                ' "old" or "odd" only
"o[^l]d"                ' "odd", "ord", but not "old"
"od?"                   ' "o" or "od"
"od*"                   ' "o", "od", "odd"
"od+"                   ' "od", "odd", etc
"[A-Z][a-z]*"           ' any uppercase word
"[0-9]+"                ' any stream of digits
"\."                    ' decimal point (needs escape character)
"[1-9]+[1-9]*"          ' any stream of digits not starting with 0
"[+\-]?[0-9]*[\.]?[0-9]*" ' any number with optional sign and decimal point (needs
two escape characters)
"[A-Z0-9]+ [0-9]+[A-Z]+" ' extracts a UK post code
"[a-zA-Z0-9._-]+@[a-zA-Z0-9_-]+\.[a-zA-Z][a-zA-Z.]*[a-zA-Z]" ' extracts any email
address
```

### Additional Email2DB Regular Expressions

In both the Look For and Then Look For fields you can include a number of control characters in addition to regular expressions:

```
<CR>      Carriage return
<LF>      Line feed
<CRLF>    Carriage return/line feed
<TAB>     Tab
<ESC>     Escape
*         When used on its own the * character finds the next
          non-space or control character.
<xxx>     Where xxx is the ASCII character code
```

This can be useful when searching for data. For example, suppose the text contains:

```
Your serial number is:
1234-5678
```

We want to extract the serial number, so we could look for 'Your serial number is:' and then look for '<CRLF>' - because the serial number is on the next line. Another way of doing the above would be to look for 'Your serial number is:' and then look for '*' - which would effectively look for anything after 'Your serial number is:'.

Email2DB regular expressions are compatible with the Microsoft RegEx Library. For more regular expression examples, please see:

http://msdn.microsoft.com/en-us/library/az24scfc(v=vs.110).aspx

http://regexone.com/

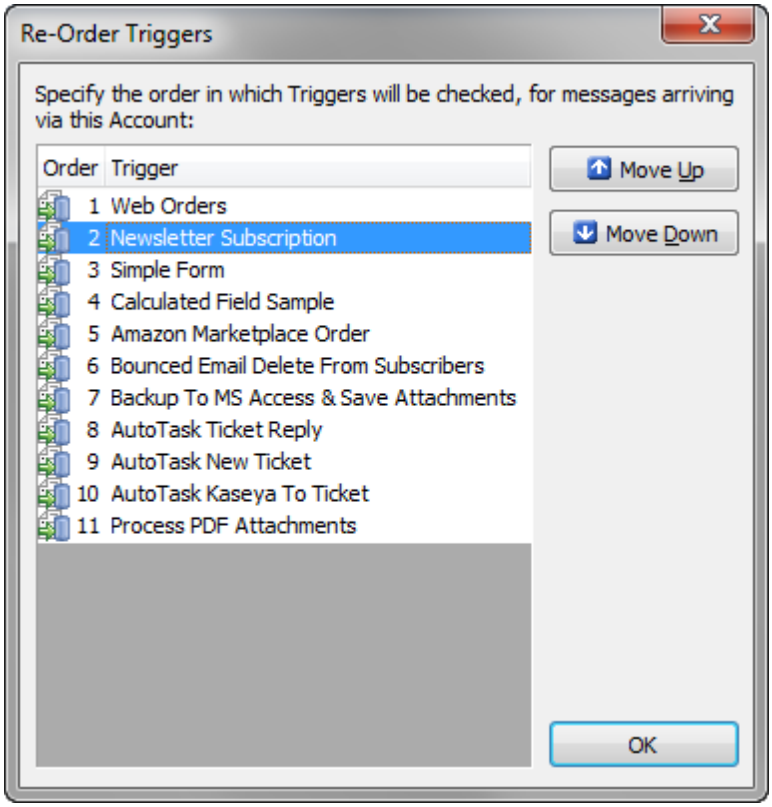
http://en.wikipedia.org/wiki/Regular_expression

## 15.3 Reordering Triggers

When an incoming message is processed for an Account, Email2DB checks each Trigger in turn, to see if it matches the Trigger conditions. The first Trigger that matches is then processed.

By default, Triggers are checked in the order they were created.

You can change the order that Triggers are checked. To do this, select **Reorder Triggers** from the Trigger shortcut (right-click) menu:




Here you can change the order in which triggers are checked for incoming messages.

## 15.4 The Email2DB Web Services

The Email2DB Web Services provide a number of functions required by remote Email2DB Administrator, Email2DB Client and for the Wait For Validation Action.

If you will not be running the Email2DB Administrator or Email2DB Client on remote computers and will not be using the Validation Action then the Email2DB Web Services are not required.

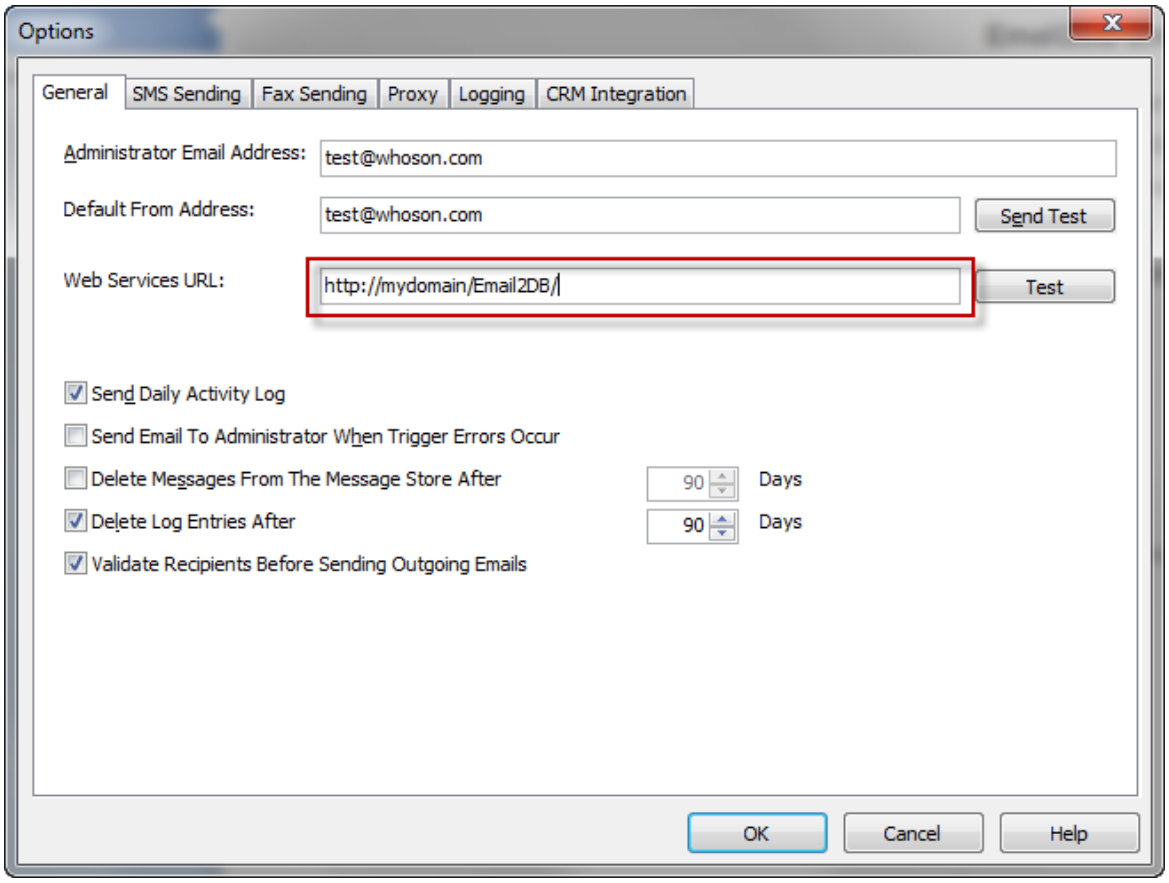
During installation the web services are installed under your Default Web Site in IIS in a virtual directory called 'Email2DB'.
This points to the **C:\Program Files\Email2DBV3\WebServices** folder.

The Web Services use the **.NET Framework 3.5** - which is also installed during the main Email2DB Setup (if its not already installed).

You can move the Web Services to a different web site or virtual directory. If you do move, make sure the .NET Framework version of Application Pool is set to .NET Framework v2.0.xx (which is compatible with .NET Framework 3.5).

### Public Address

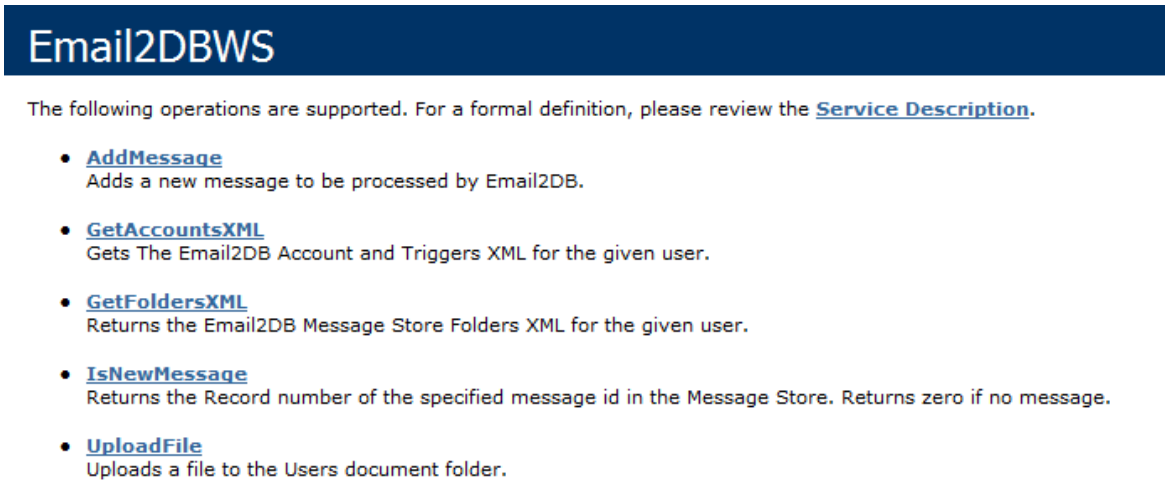
If you will be using the Email2DB Administrator or Email2DB Client on remote computers, or if you will be using the Wait For Validation Action then you need to specify the Web Services URL in your Email2DB Program Options (**File** Tab - **Program Options** in the Email2DB Administrator):




This should be the address of the Web Services when accessed externally.

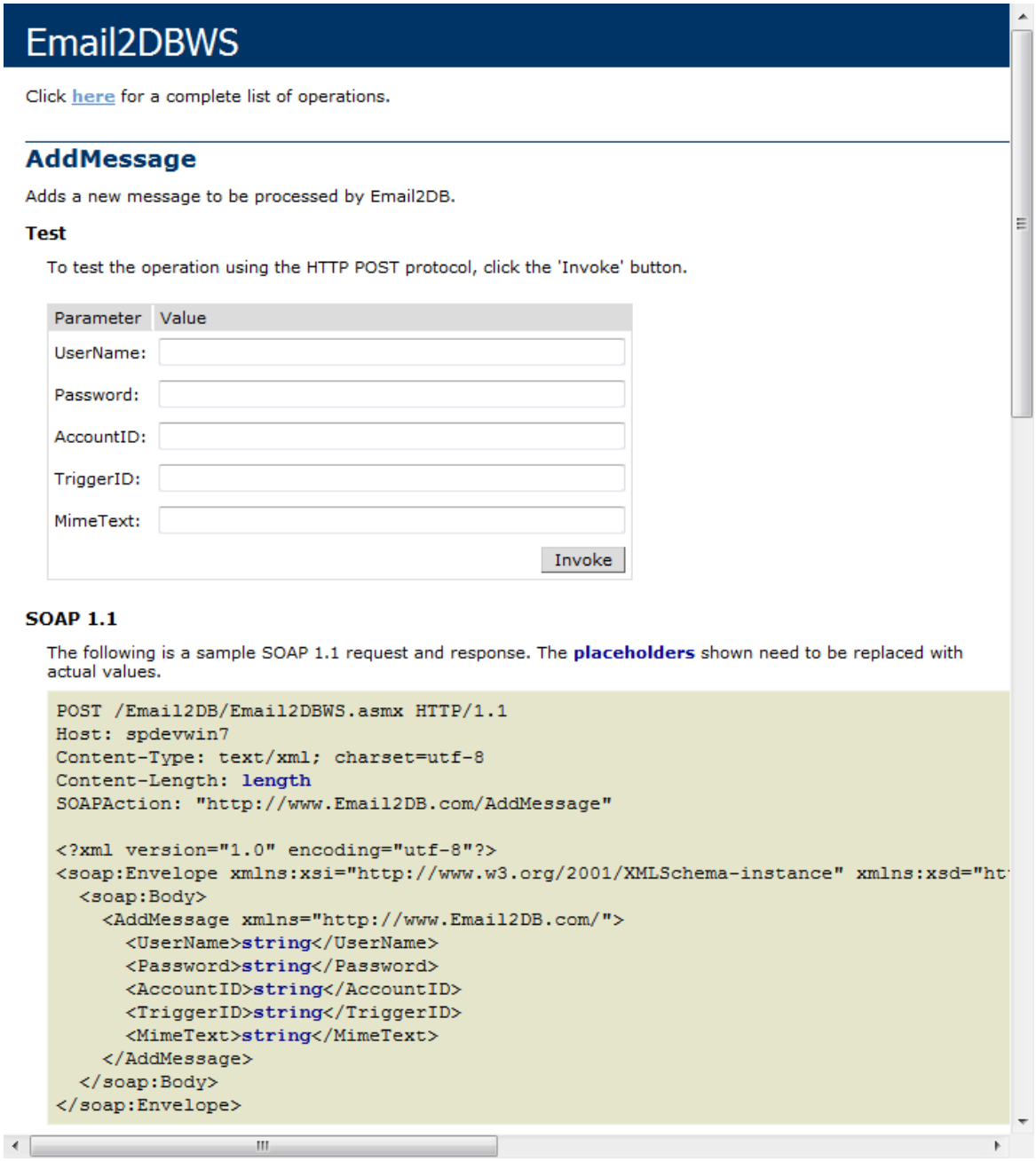
Click the **Test** button to verify the address. This will open your web browser at the address specified.

You should get a web page that looks like this:




## Adding Messages To Email2DB Using The Web Services

The **AddMessage** method can be used to add new messages to Email2DB to be processed by a given Account/Trigger. This enables developers to create their own methods of adding messages to Email2DB. Messages added by the web service will be processed immediately by the Email2DB Message Processor.




You must specify a **UserName** and **Password**. The Password can be sent as a MD5 Hash of the actual password (base64 encoded) or as a plain-text password.

You must also specify the **AccountID**. This is the ID of the Email2DB Account you want to assign the message to. The user must have access to this Account.

The **TriggerID** can be specified if you want the message to be processed by a specific Trigger within the Account. If no TriggerID is specified then Email2DB will assign a Trigger to the message based on the first Trigger for the Account that matches the Trigger Conditions.

The **MimeText** should be set to the full mime-text of the message.

To obtain the Account & TriggerID's you can call the **GetAccountXML** method. This will return the Email2DB MetaData XML for a given user.

# 15.5 Customizing The Validation Response Page

For Validation Actions, the validate.aspx page is called when the validation URL is accessed. This page uses the stylesheet.css file located in the **\Program Files\Email2DBV3\WebServices** folder.

You can edit the **stylesheet.css** file to customize the appearance of the Validation Response page.

## Customizing Strings

Located in the **\Program Files\Email2DBV3\WebServices\Strings** folder is the file **en.xml**

This XML file contains entries for each of the strings used in the validation response:

```
<?xml version="1.0" encoding="utf-8" ?>

<!--
Use this file to change the default text strings used by the validate page.
You can create a separate XML file for each language. Save the file using the 2
digit language code as the file name (eg, fr.xml).
Any string that is blank or not specified will use the default value.
-->

<strings>
       <PageTitle>Email2DB Validate Message</PageTitle>
       <PageHeader>Validate Email2DB Message</PageHeader>
       <ButtonAccept>Accept</ButtonAccept>
       <ButtonNotAccept>Do Not Accept</ButtonNotAccept>
       <ResponseAccepted>Validation complete. Thank you for validating the message.
You accepted the message.</ResponseAccepted>
       <ResponseNotAccepted>Validation complete. Thank you for validating the
message. You did not accept the message.</ResponseNotAccepted>
       <TestResponse>Validation Response Page Test Successful</TestResponse>
       <InputInvalid>The data that you have entered is not valid.</InputInvalid>
       <InputInvalidEmail>The Email Address you have entered is not valid.</
InputInvalidEmail>
       <InputInvalidOption>Please select a value.</InputInvlidOption>
       <InputInvalidRange>The data must be in the range {minvalue} to {maxvalue}</
InputInvalidRange>
</strings>
```

You can edit this file in a text editor to adjust any of the text strings.

## Different Languages

You can also save the file for different languages using the 2 digit language code as the file name (eg, fr.xml for French). The language used will be automatically read from the browser settings of the user viewing the validation page.

## Removing Branding

If you want the Powered By Email2DB or (C) Parker Software items removing or if you want a custom validation page design, please contact our Professional Services team.

## 15.6 Viewing Messages In A Browser

The Email2DB Web Services includes a **viewmessage.aspx** page. This can be used to render any Email2DB Message Store message in a browser.

The built-in field **%msg_viewurl%** returns the full URL to each message. You could create an Action to send this in an email to provide a permanent, browser-viewable link to the message - or save it in a database to use as a link to the message in your own applications.

The ID of the message is passed as **viewmessage.aspx?id={msglinkid}**

The {msglinkid} is a combination of the MessageStore fields 'OrganizationID' and 'Guid' for the message.

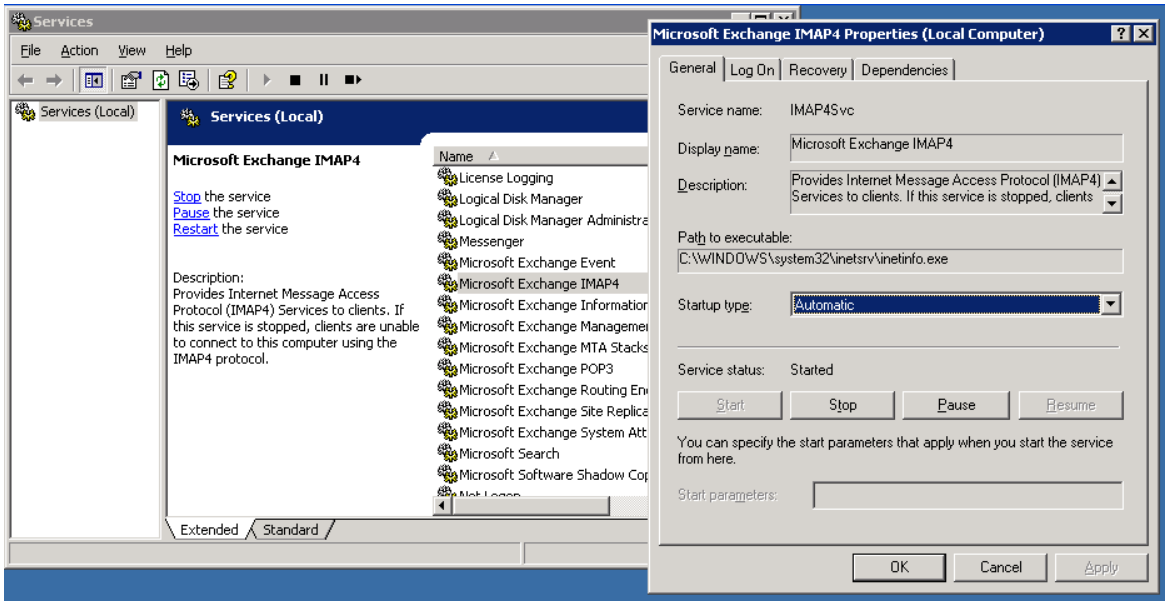
# 15.7 Using Email2DB With Exchange & IMAP

## Exchange 2007/2010

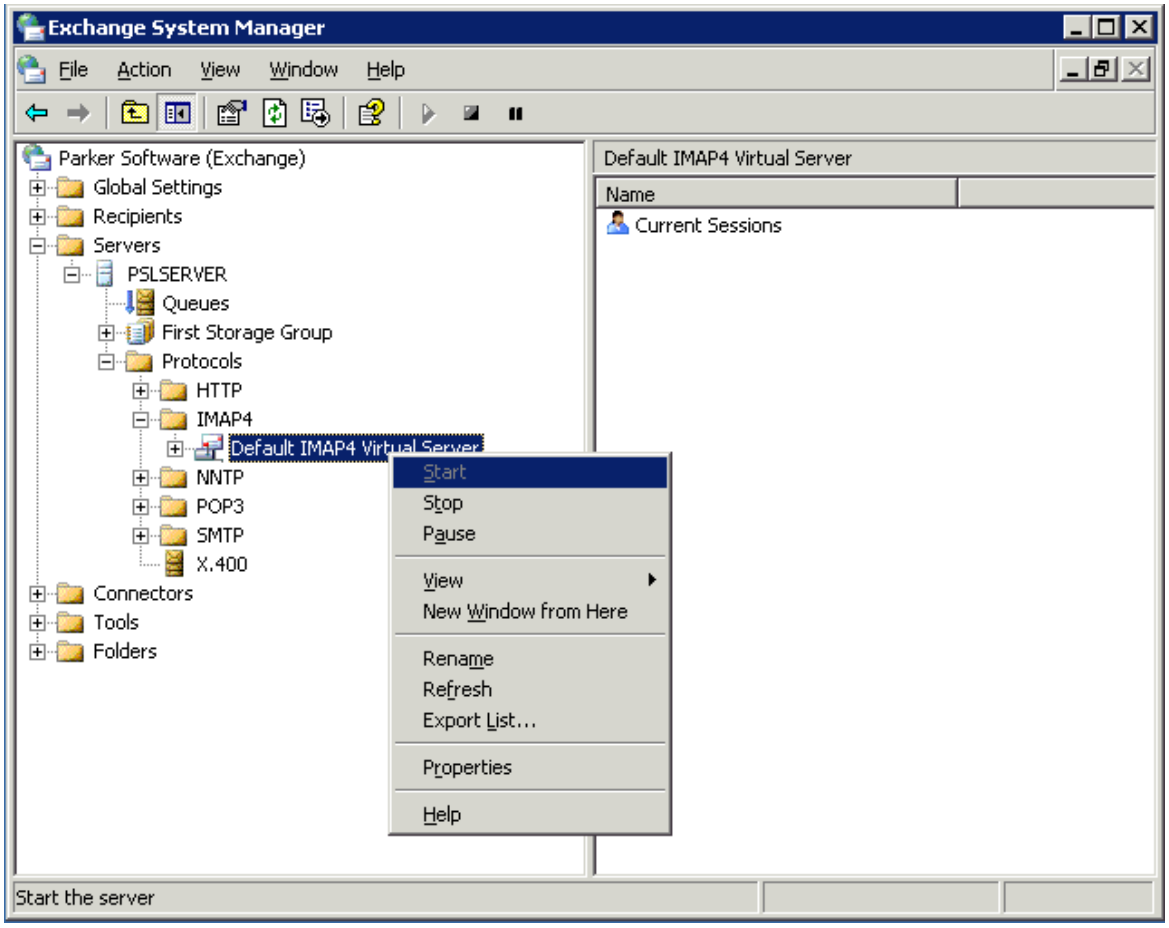
POP3 and IMAP are disabled by default in Exchange 2007. For details on how to enable, open the Management Console, select help and search for 'IMAP enabling'.

## Exchange 2003

By default the IMAP protocol on Exchange Server is disabled and the IMAP Service itself is marked as disabled. If you want to retrieve messages from Exchange Server via IMAP then you will need to enable the IMAP Service AND start the IMAP Protocol:




Set the IMAP4 Service start mode to Automatic using the Windows Service Applet in your Administrative tools.




Then start the IMAP4 Protocol in Exchange Server System Manager

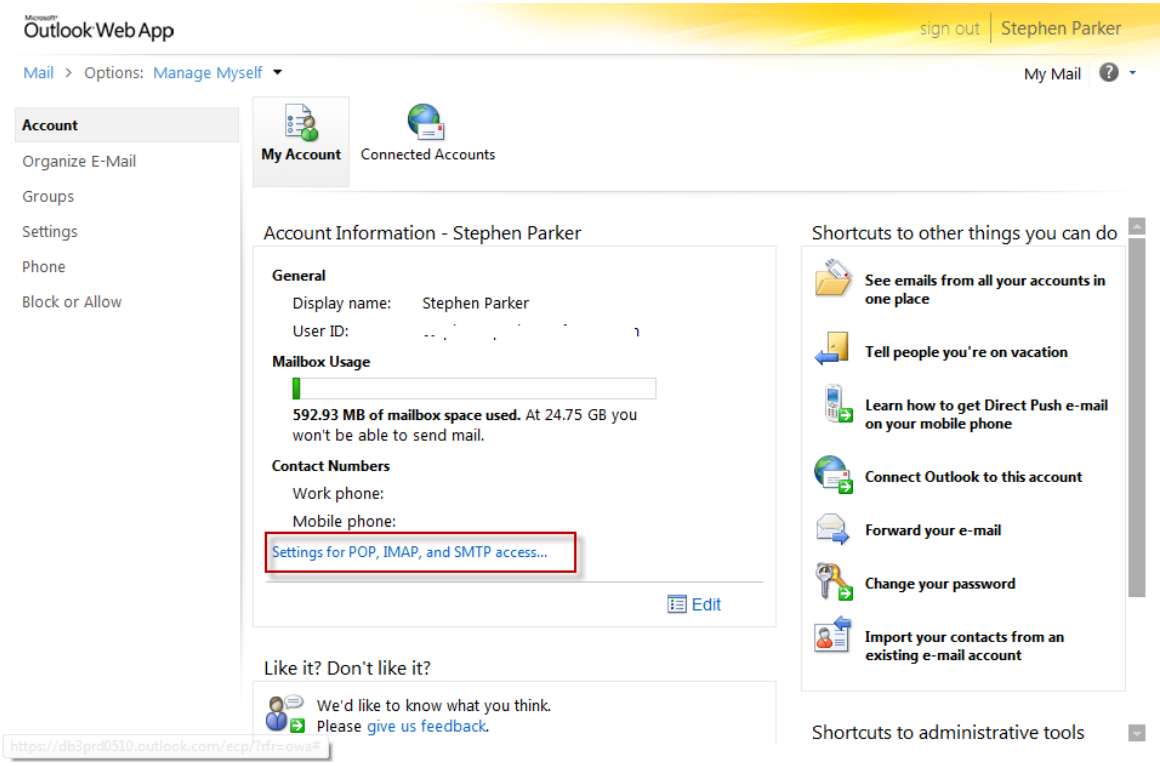
Ensure the IMAP port (143) is open on your server firewall.

You will then be able to retrieve emails from Exchange using IMAP.

## 15.8 Using Email2DB With Office 365

Email2DB can read emails from your Office 365 mailbox - either via IMAP or using Exchange Web Services.

To obtain your server address, login to your Office 365 Web Access. Select **Outlook** - then **Options** - **See All Options**.




Click the **Settings For POP, IMAP and SMTP Access**... link.

The **Server Name** & **Port** for IMAP access will be shown. Use this in the IMAP Settings for your Email2DB Account.

The User Name/Password will be the same as you use to login to Office 365 Web Access.

### Syncing A Mailbox Using Exchange Web Services

If you want to use Exchange Web Services to synchronize a mailbox - you should use the URL:

**https://{servername}/ews/exchange.asmx**

Where {servername} is the Server Name displayed in the settings above.

The same URL can be used for any Exchange Actions (Update Contact, Appointment etc).

### Updating Office 365 SharePoint Sites

Email2DB can update Office 365 SharePoint Sites. To obtain the URL to use login to Office 365 Web Access. Click the **Team Site** link. This will open SharePoint. The URL shown in your browser will be: http://{yourname}/TeamSite

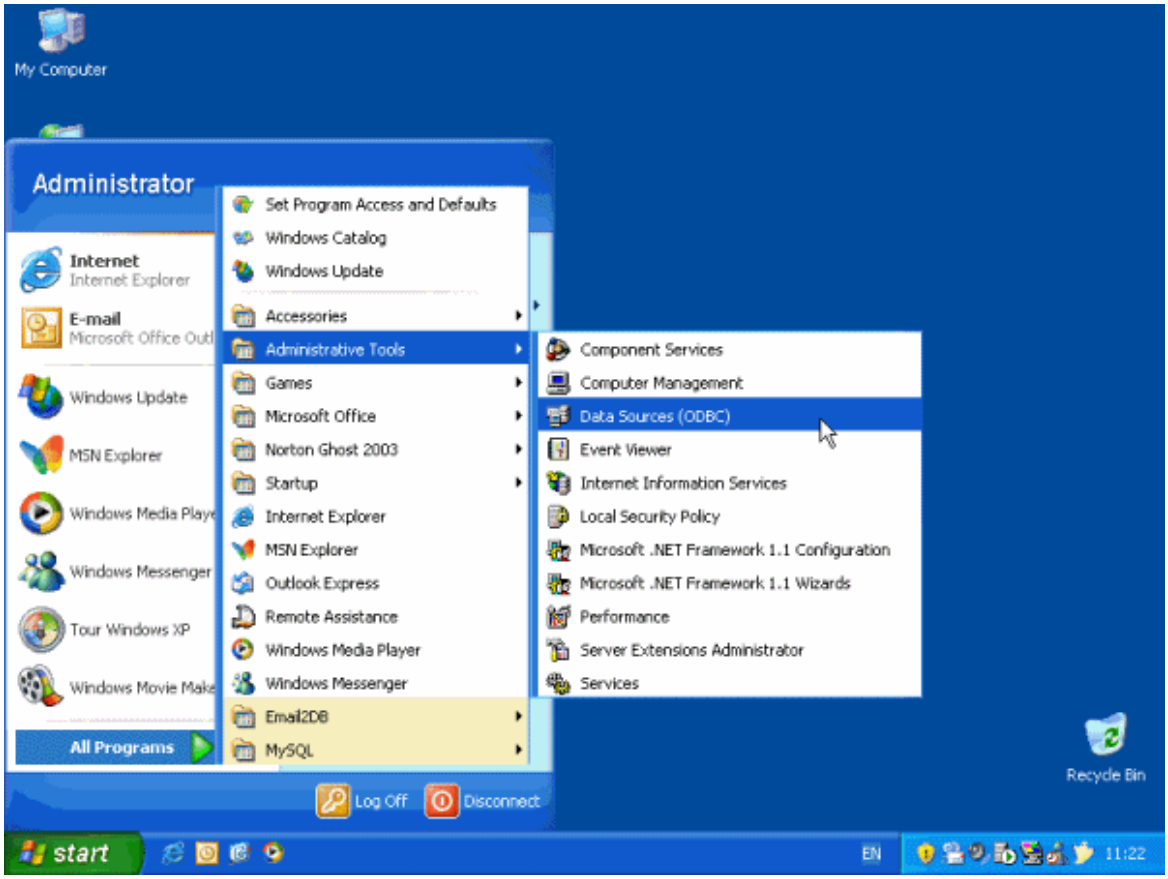
Use the URL here up to the /TeamSite

## 15.9 Updating MySQL Databases Using Email2DB

When updating MySQL databases via Email2DB, you must use the MyODBC Connector. This can be downloaded from the MySQL web site (www.mysql.com). The connector itself can be downloaded here.

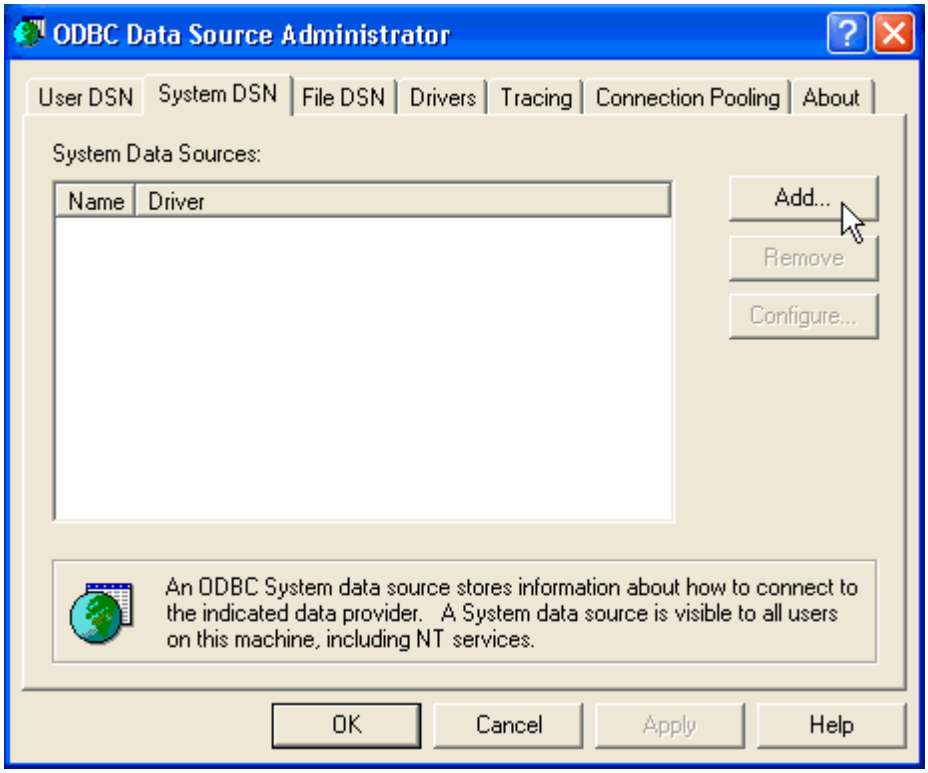
You must the create a DSN (Data Source Name). To do this follow these steps:

From your **Administrative Tools**, choose **Data Sources (ODBC)**:

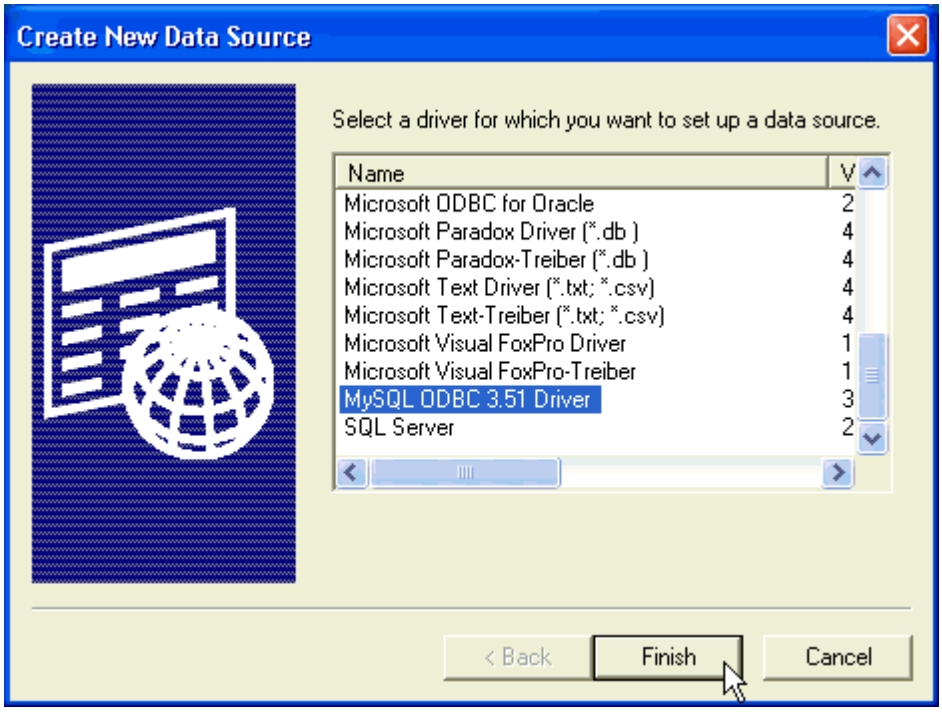



The ODBC Data Source Administrator will then start.

Click the **System DSN** tab:



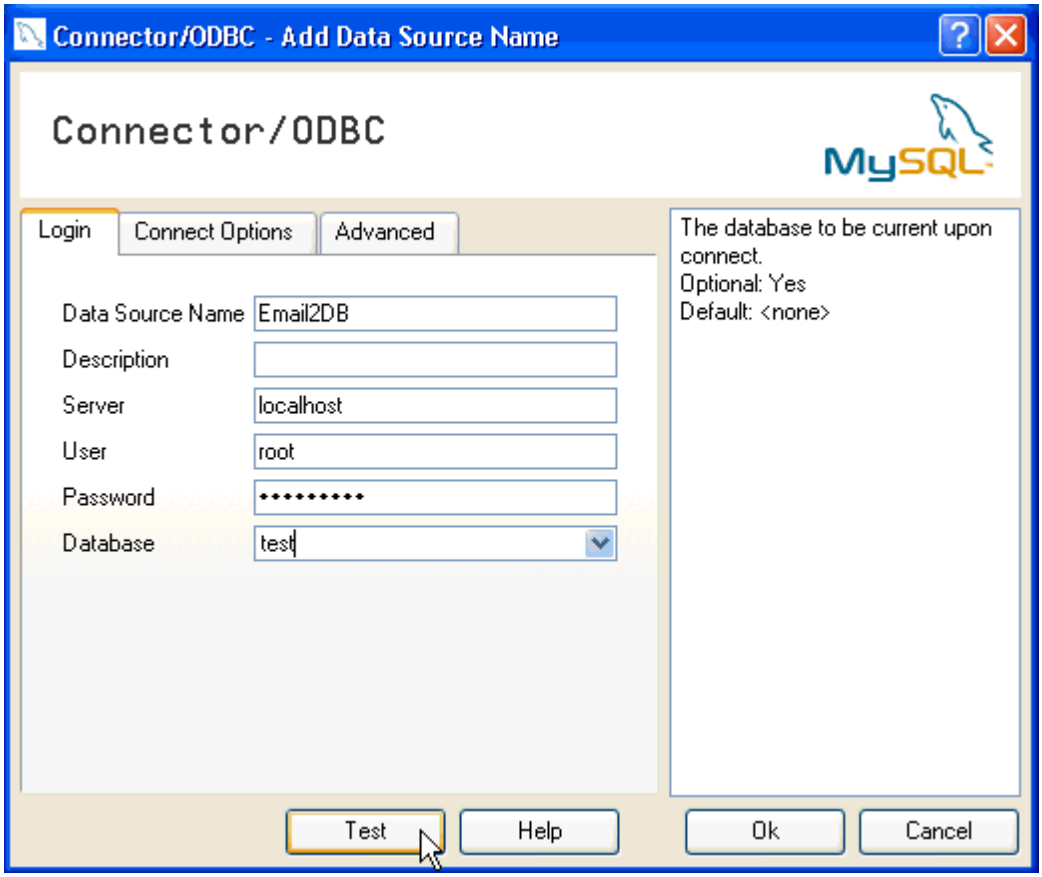

Click the **Add** button:




From the Drivers list select **MySQL ODBC 3.51 Driver** and click **Finish**.

If the MySQL driver is not listed then you do not have the MySQL ODBC Connector installed. If you just installed it, try restarting your PC.

The Connector properties will then be displayed:



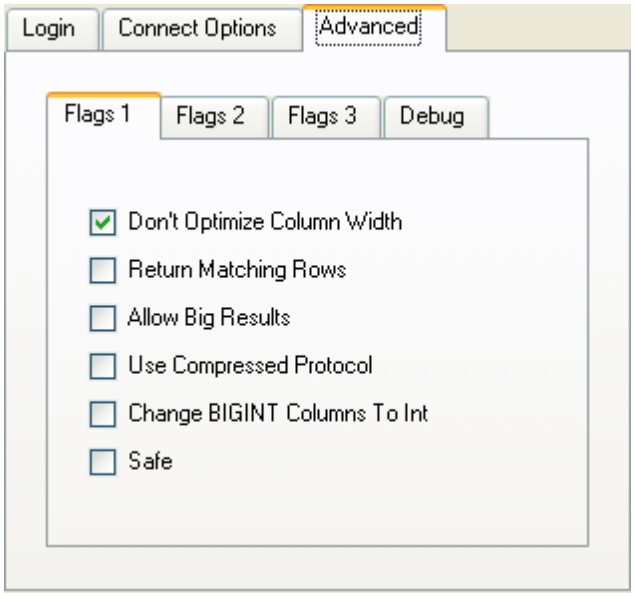

Enter a **Data Source Name** - this can be any descriptive name for your Data Source.

Enter the **Server** host name. If MySQL is running on this PC then enter 'localhost'.

Enter the root **user** name and **password**.

Select the **Database** from the drop down list. This list will show all MySQL databases defined on the MySQL Server that is running on the server specified.

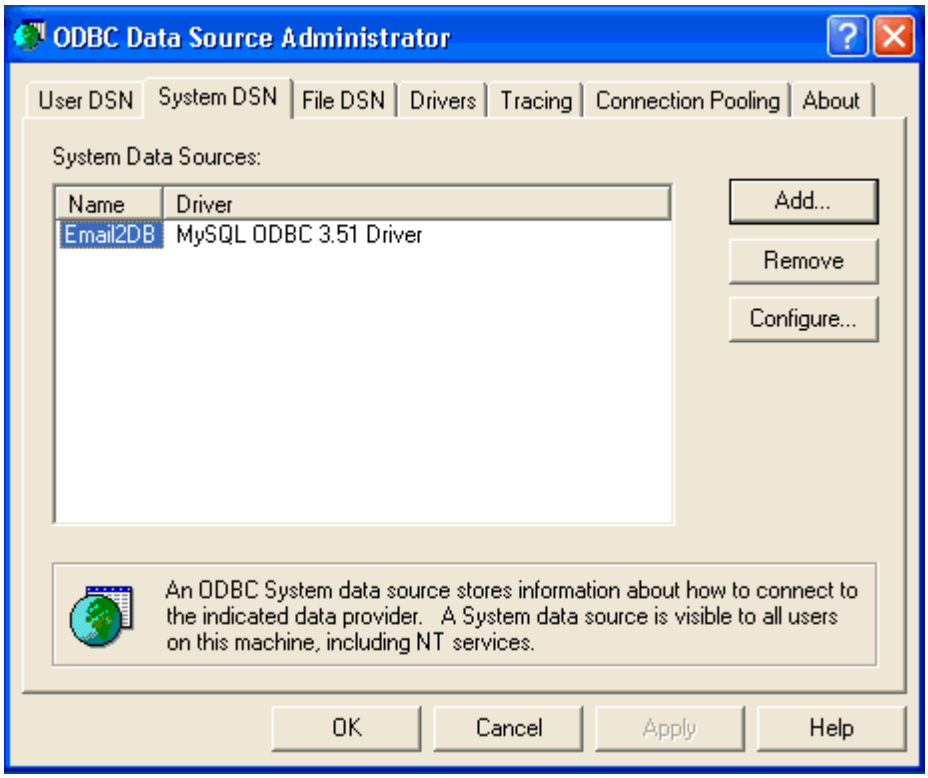
Click the **Advanced** tab.




Make sure the 'Don't Optimize Column Width' option is selected. This is required due to limitations of the MySQL ODBC driver.

Click the **Test** button to verify the connection.

Click OK to save the DSN. The DSN will then be listed in the System Data Sources list:




You will now be able to update the MySQL database via Email2DB.

On the Email2DB **Trigger Actions** area select **Update Database (Direct)** then click the **Build** button to build a connection string.

Select **Microsoft OLE DB Provider for ODBC Drivers** and click **Next**

Select the **Use data source name** option and then select the DSN that you created earlier from the list.

Click the **Test Connection** button to verify we can open the database via ODBC.

Click **OK** to save the connection string:

The Connection String will then be updated. Select MySQL as the SQL Dialect.

The MySQL database will now be updated via Email2DB.

# 15.10 HTML To XML Conversion

The HTML Get Trigger Action and the Set Variable Action both allow the returned HTML to be converted to XML.

Converting to XML will make it easier to extract certain elements from the HTML.

### Example Of HTML to XML Conversion

HTML returned:

```
<html>
	<head>
		<title>This is a test</title>
		<meta http-equiv="Content-Language" content="en-us">
		<meta http-equiv="Content-Type" content="text/html; charset=windows-
1252">
	</head>
	<body>
		<h1>This is the heading</h1>
		<p>Lorem ipsum dolor sit amet, <b>consectetur</b> adipisicing elit,
sed do eiusmod tempor incididunt ut labore <br> et dolore magna aliqua.
		<p>Ut enim ad minim veniam, <a href="http://www.google.com/">quis
nostrud exercitation</a> ullamco laboris nisi ut aliquip ex ea commodo consequat.
	</body>
</html>
```

The converted XML is shown below.

- The XML is written to match the encoding of the HTML. In the HTML above, the charset is windows-1252, so the encoding attribute is set to windows-1252.
- The root node of the XML document is always <root>. The <html> node is found directly underneath. The reason for the "root" node is because you may encounter poorly formed HTML such that it has more than one root-level node.
- All text content is placed under <text> nodes.

```
<?xml version="1.0" encoding="windows-1252" ?>
<root>
	<html>
		<head>
			<title>
				<text>This is a test</text>
			</title>
			<meta http-equiv="Content-Language" content="en-us"></meta>
			<meta http-equiv="Content-Type" content="text/html;
charset=windows-1252"></meta>
		</head>
		<body>
			<h1>
				<text>This is the heading</text>
			</h1>
			<p>
				<text>Lorem ipsum dolor sit amet, consectetur
adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.
				</text>
			</p>
			<p>
				<text>Ut enim ad minim veniam, </text>
				<a href="http://www.google.com/">
					<text>quis nostrud exercitation</text>
```

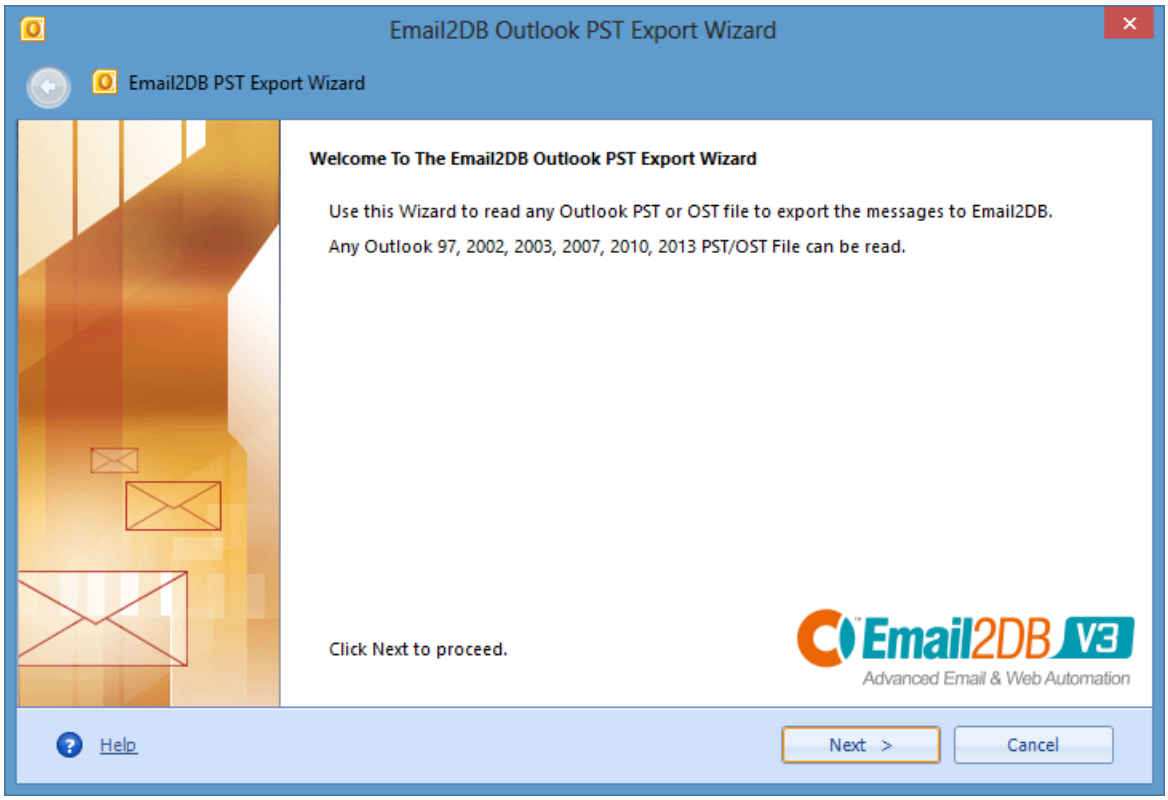
```
                                </a>
                                <text>ullamco laboris nisi ut aliquip ex ea commodo
consequat.
                                </text>
                        </p>
                </body>
        </html>
</root>
```

## 15.11 Outlook PST Export Wizard

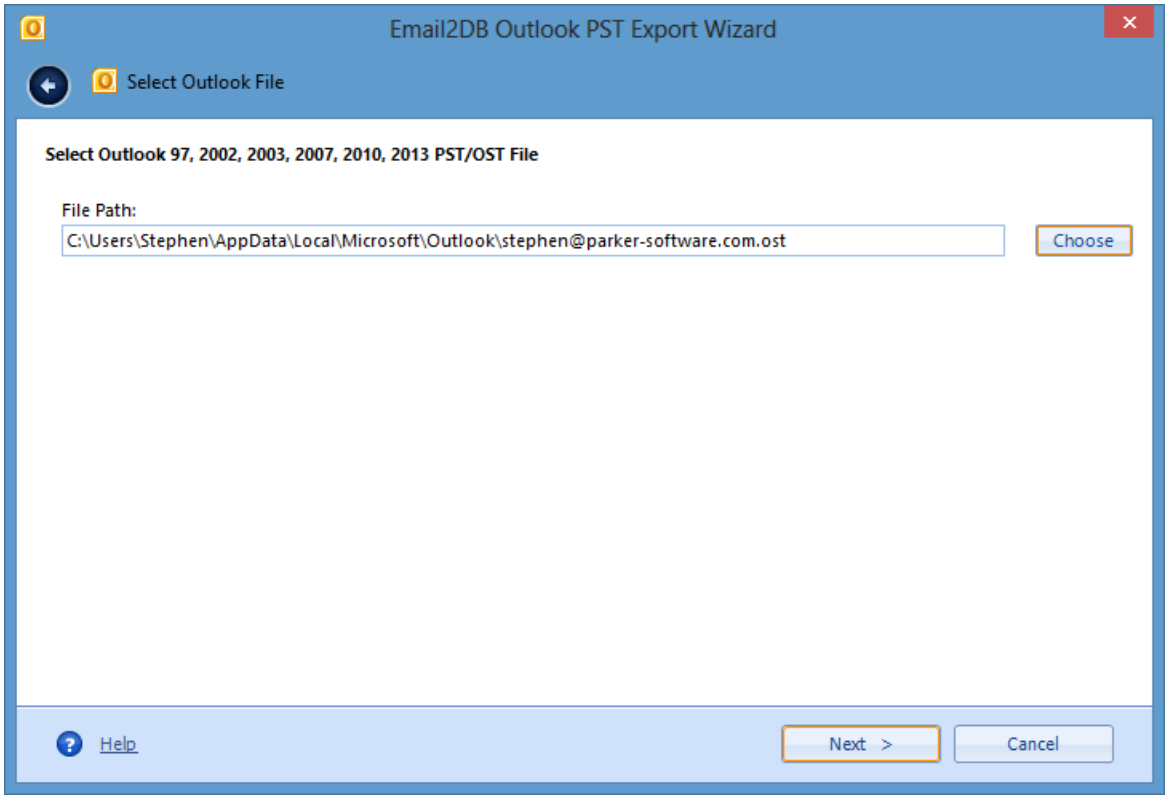
Email2DB includes an Outlook PST Export Wizard. This tool can read any Outlook PST or OST file and export email messages to be processed by Email2DB.

It works with any Outlook 97, 2002, 2003, 2007, 2010 & 2013 PST or OST file. It reads the file directly and does not need Outlook to be installed.

To start the Wizard choose **File** - **Outlook PST File Export Wizard** from the Email2DB Administrator.




Click **Next** to proceed.

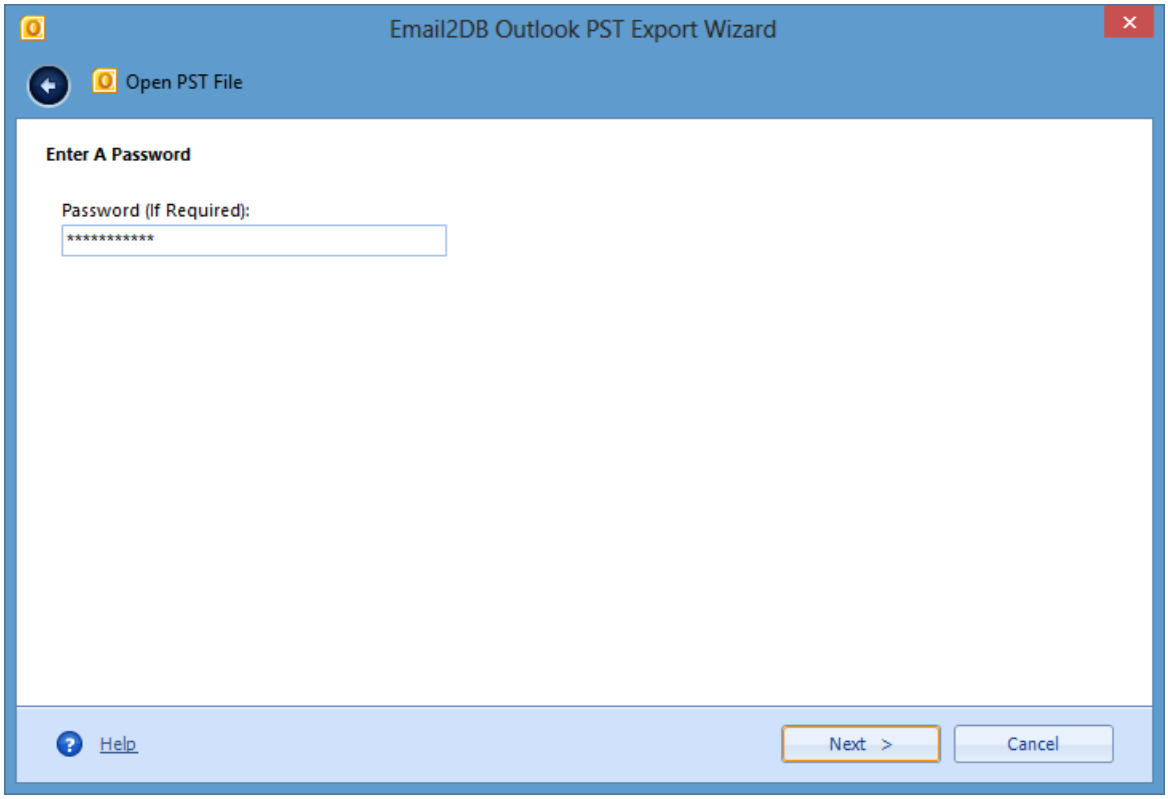



You must then select the PST or OST file to open. Click the **Choose** button to select.

Outlook by default uses the *{user}\{application data}\Local\Microsoft\Outlook* folder to store PST/OST files. This folder may be hidden, so you will need to enable 'Show Hidden Files/Folders' option in Windows first.

If you are using Outlook to connect to an Exchange Server then you will need to select the OST file. Change the **File Type** selector to Outlook OST Files to select.

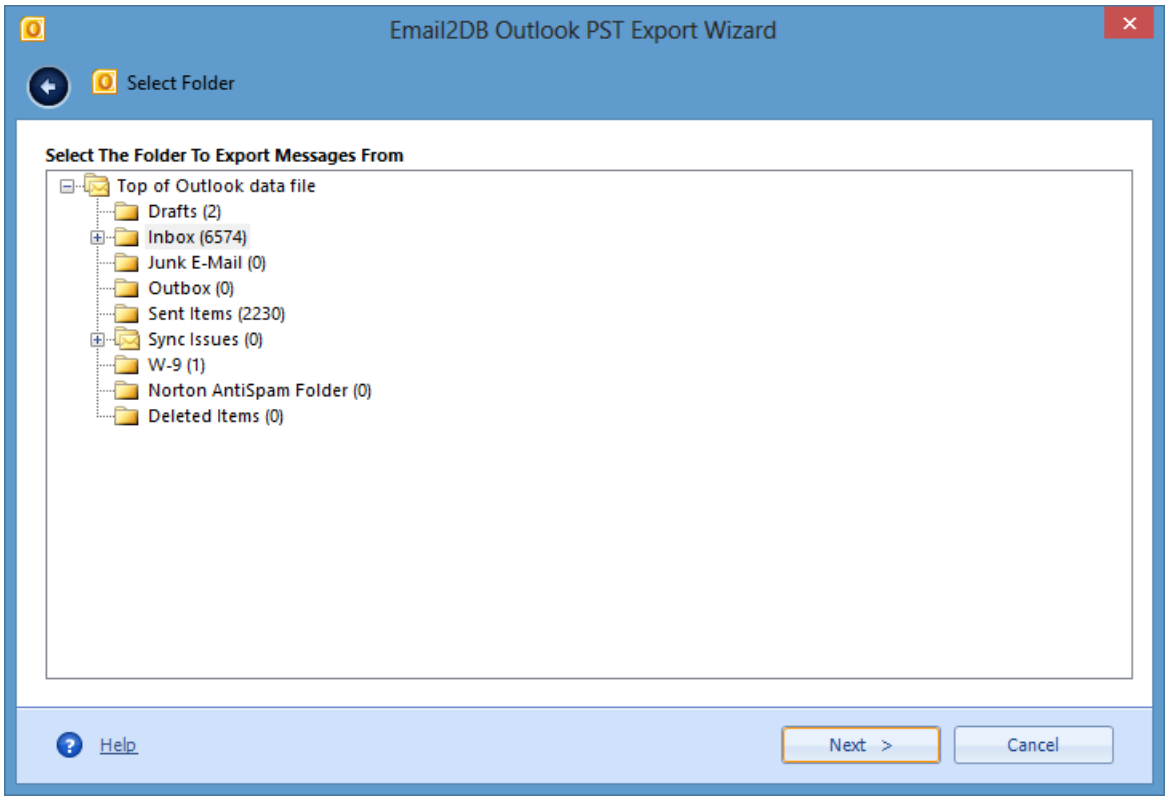
Once you have selected a PST or OST file to open click **Next**.



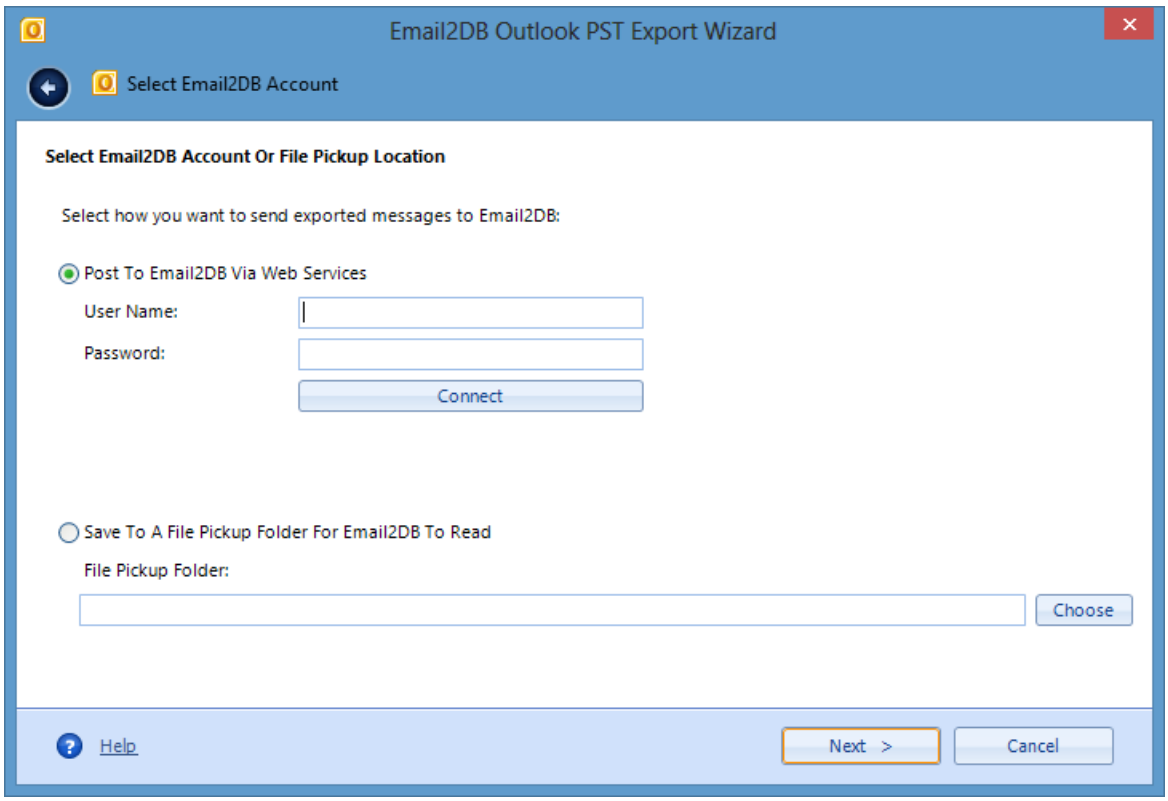

If your PST or OST file is password protected you must now enter the password and click **Next**. If not - click Next without entering a password.

The PST file will then be opened. This may take a minute or so if the file is large.

The folders contained in the PST file will then be displayed:



Select the folder you want to export and click **Next**.




You now have two options to export the messages in the selected folder to Email2DB.

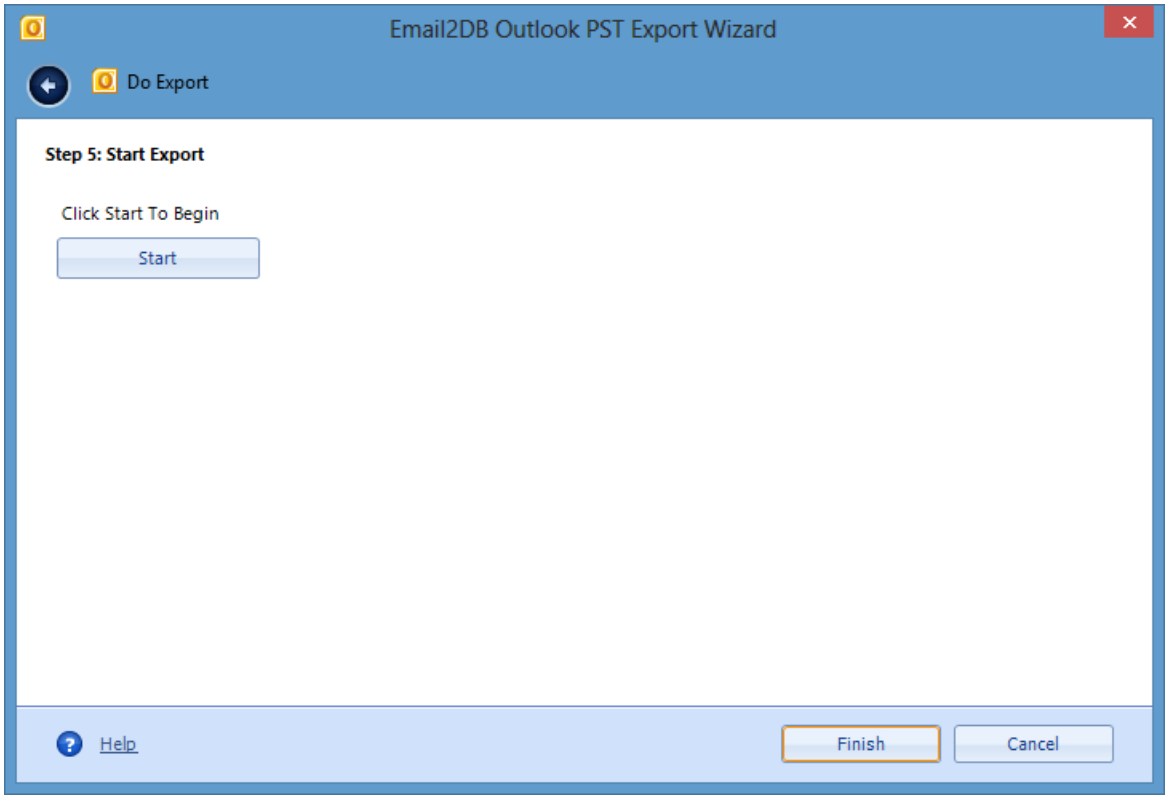
**1. Post To Email2DB Via Web Services**

For this option you need to enter your Email2DB **User Name** & **Password**. Then click **Connect**. The Email2DB Accounts will then be listed. Select the **Account** you want to export the messages to. The Wizard will post each message in the Outlook folder to the Email2DB Web Services for the selected Account. These messages will then be processed by Email2DB.

**2. Save To A File Pickup Folder For Email2DB To Read**

For this option you select a folder on your computer. Each message in the selected folder will be saved as an EML file in the chosen **File Pickup Folder**. You can then create an Email2DB Account with the File Pickup message source. Point the Email2DB Account File Pickup location to the same folder. Email2DB will then read each new EML file and process them as new messages.

Once you have selected the export option click **Next**:




Click **Start** to begin.

During the export process you can click the **Cancel** button to cancel.

## 15.12 Translation Supported Languages

The Translate, Detect & Speak language Actions support the following languages:

| Code | English Name |
|---|---|
| ar | Arabic |
| bg | Bulgarian |
| ca | Catalan |
| zh-CHS | Chinese (Simplified) |
| zh-CHT | Chinese (Traditional) |
| cs | Czech |
| da | Danish |
| nl | Dutch |
| en | English |
| et | Estonian |
| fa | Persian (Farsi) |
| fi | Finnish |
| fr | French |
| de | German |
| el | Greek |
| ht | Haitian Creole |
| he | Hebrew |
| hi | Hindi |
| hu | Hungarian |
| id | Indonesian |
| it | Italian |
| ja | Japanese |
| ko | Korean |
| lv | Latvian |
| lt | Lithuanian |
| ms | Malay |
| mww | Hmong Daw |
| no | Norwegian |
| pl | Polish |
| pt | Portuguese |
| ro | Romanian |
| ru | Russian |
| sk | Slovak |
| sl | Slovenian |
| es | Spanish |
| sv | Swedish |
| th | Thai |
| tr | Turkish |
| uk | Ukrainian |
| ur | Urdu |
| vi | Vietnamese |

# 15.13 Email2DB Basic Scripts

Included with Email2DB is a powerful Scripting module. This allows you to create custom scripts that are executed by the Email2DB Server as it processes messages. Scripts are written using the BASIC programming language. The syntax is compatible with Microsoft Visual Basic 6. This manual is not intended to teach the Visual Basic language (there are dozens of books for that). It is only intended to show how scripts can be used within Email2DB.

Scripts can be as complex as you need and contain many Functions and Sub-Procedures. Scripts can also use external COM objects and ActiveX controls. For example, you could use the ADO objects (ActiveX Data Objects) to access and update data sources.

Within scripts Email2DB allows you to access information about the current email being processed via simple properties.

The Email2DB Server executes three types of scripts:

### 1. Trigger Condition Scripts

Are fired on the Edge Server when Email2DB downloads new messages & checks the message to see if it should be processed. These scripts can be used to fine tune Trigger conditions.

### 2. Field Extraction Scripts

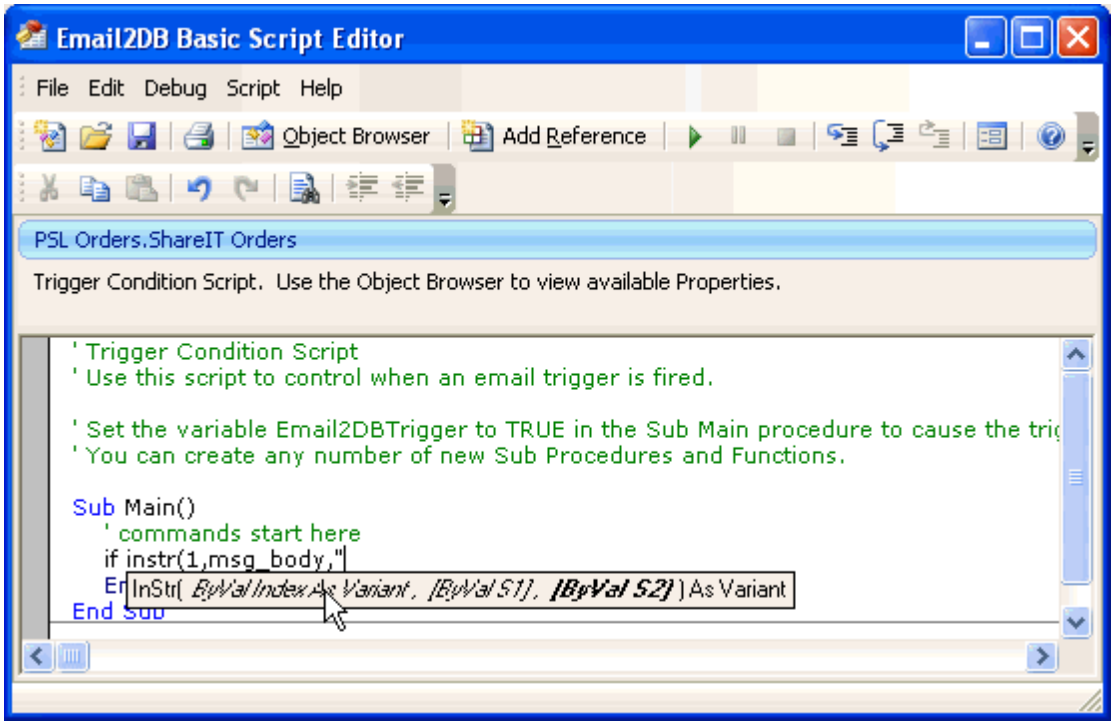
These scripts can be assigned to individual fields that you extract from the Email. They can be used to perform complex field extraction or to create calculated fields or to format fields that have already been extracted.

### 3. Action Scripts

These scripts are fired by the Message Processor Server when the message Actions are processed. These scripts can be used to perform custom actions.

### The Script Editor

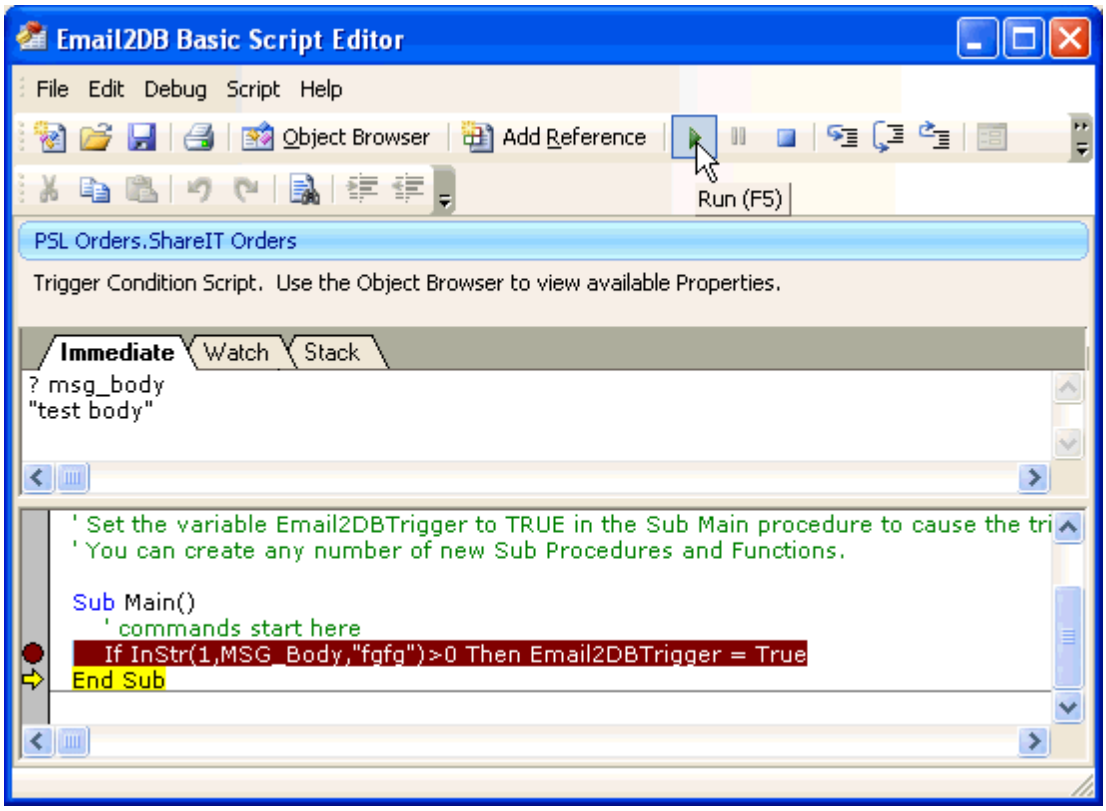
All scripts are edited using the Email2DB Script Editor




The code editing window features 'Intellisense' typing - showing the attributes of the current Basic keyword. You can also press F1 on any keyword to view the language help.

## Debugging

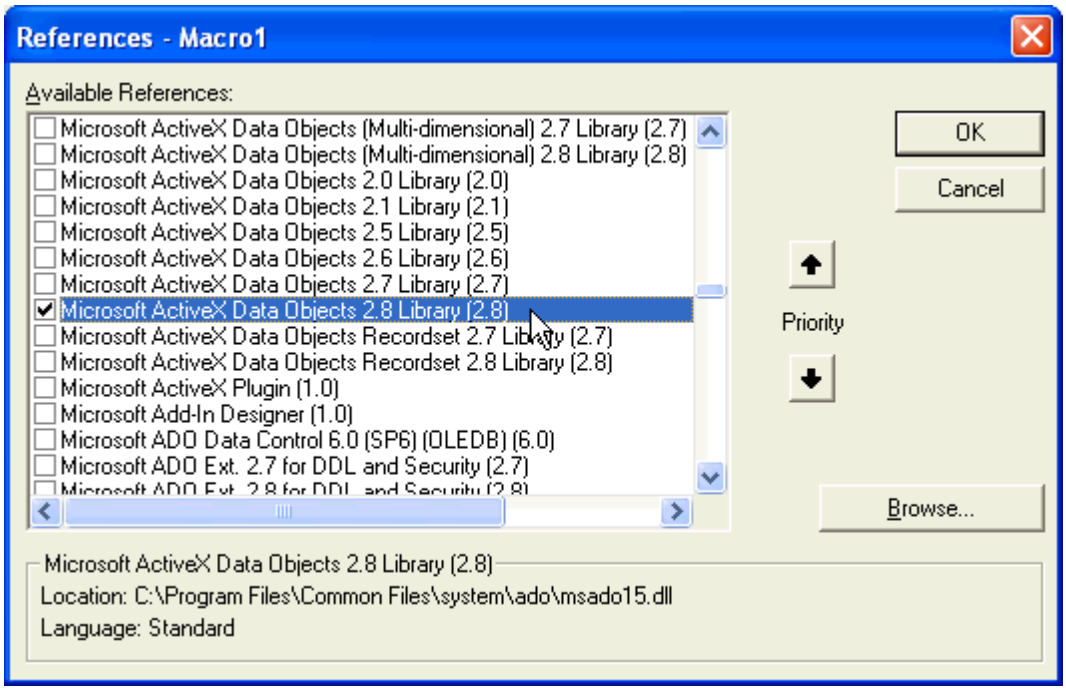
Scripts can be debugged in the code editor. Click in the left margin to set a 'break point' against a specific line and click the **Run** button.



You can step through the code by clicking the **Step Into** button (or pressing F8). In the Immediate tab you can view contents of variables by typing ? {variable}

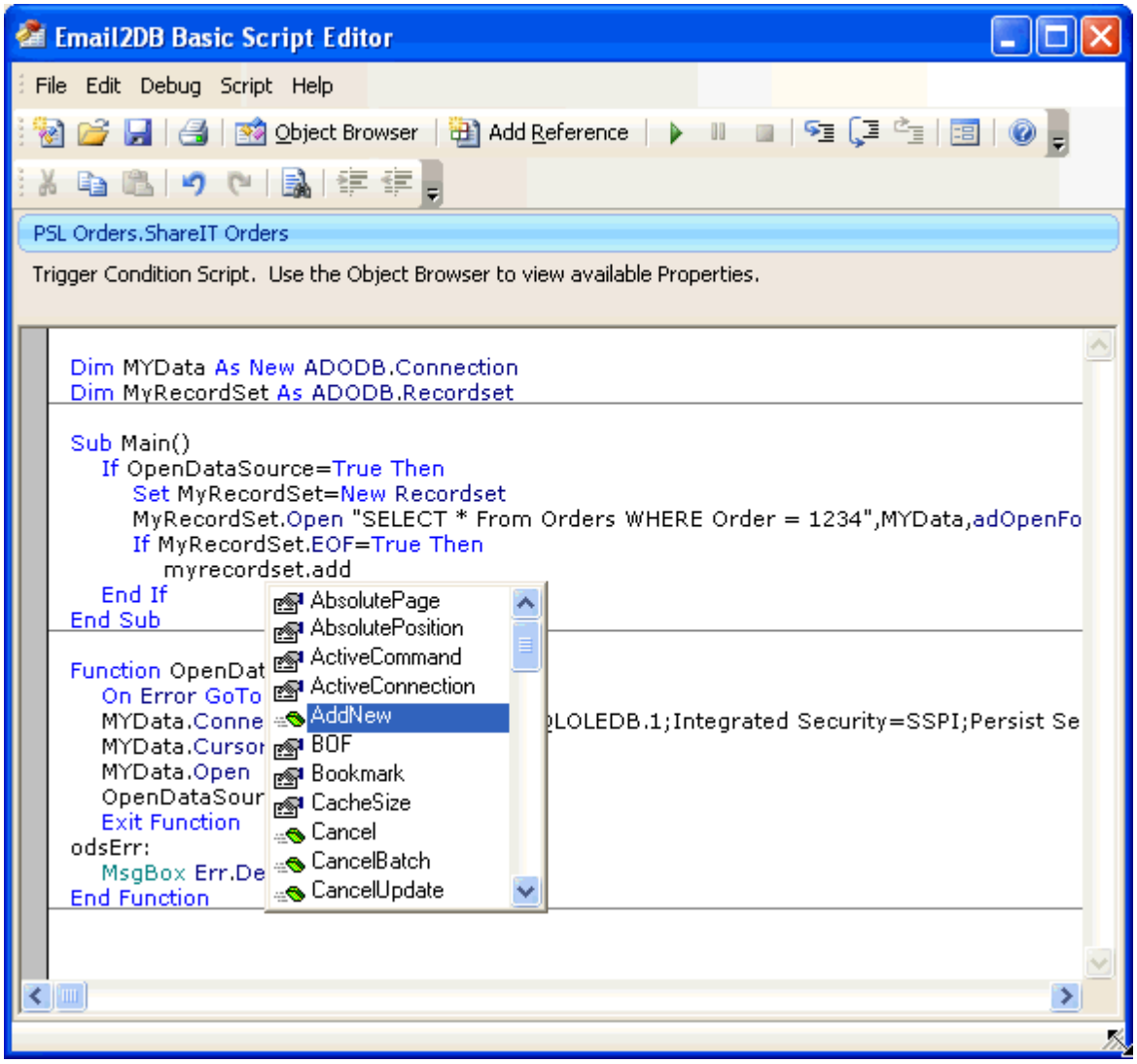
## Adding References

Click the **Add Reference** button to list the available COM objects that you can use in your scripts:




Once added you can create and use the objects.

Below is a sample script that uses the ADO library to open and manipulate a SQL Server database.

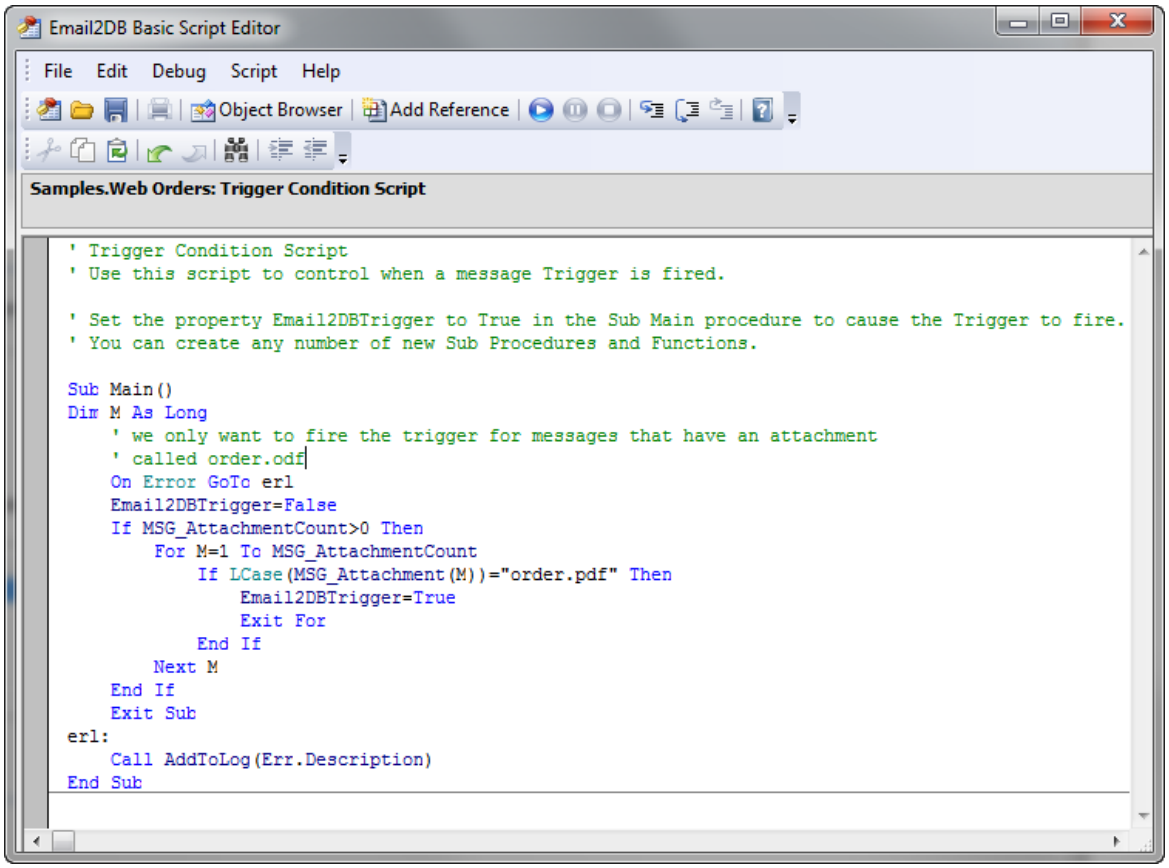


## 15.13.2 Trigger Condition Scripts

In addition to the standard Trigger conditions you can create 'Trigger Condition Scripts'. Trigger Condition Scripts allow you to have complete control of the execution of a Trigger for a specific message. The script is executed by the Edge Server as each message is received for the current account.

For example, you may only want Email2DB to process a message based on very specific criteria that cannot be handled by the standard Trigger Conditions entries.

Check the **Condition Script** option on the Trigger settings and click the **Edit** button to open the Email2DB Basic script editor.




To cause the Trigger to be processed by Email2DB you must set the variable '**Email2DBTrigger**' to **True** in the Sub Main procedure. Before Email2DB fires the trigger condition script it will check the standard trigger conditions first and pre-set the **Email2DBTrigger** property.

You can then write your basic script directly in the code window or open an existing saved script. The script can be as big as you need and contain any number of Sub-procedures and Functions. You can also use the **Add Reference** button to create links to external COM objects.

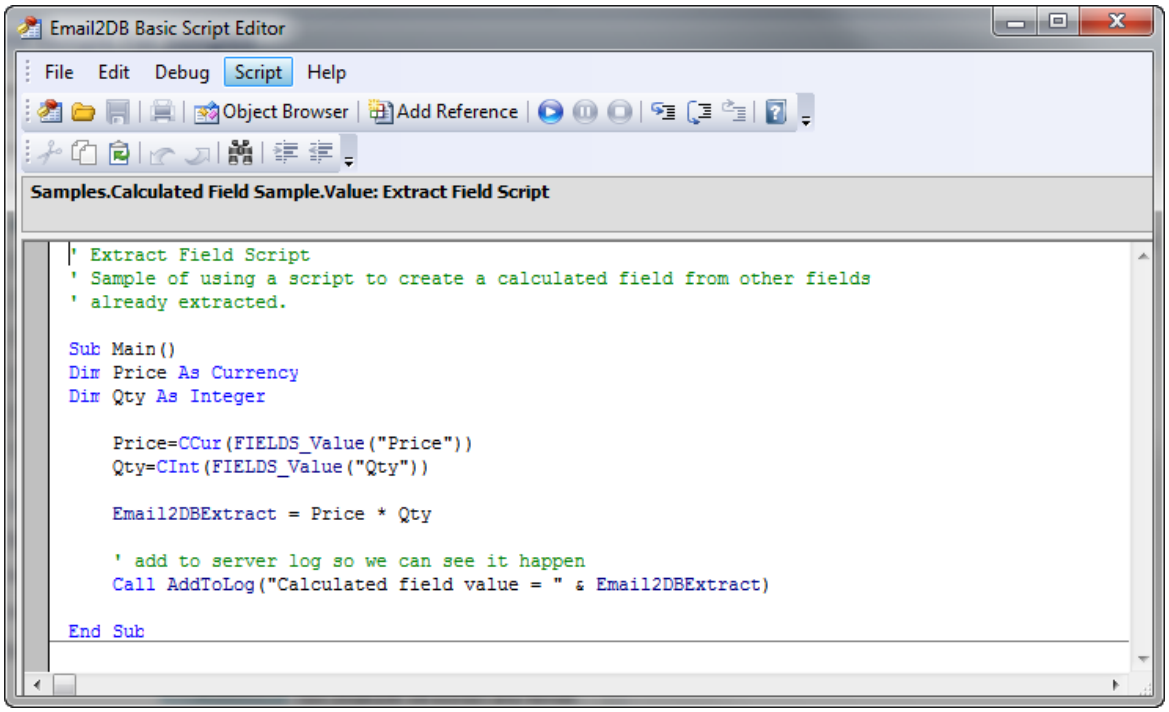
Within the script you can access properties about the current email. Click the **Object Browser** button to view a list of available properties. For example, the property MSG_Body returns the complete email message text. MSG_Header() and MSG_HeaderValue() are arrays containing all headers and header values. MSG_HeaderCount returns the number of headers.

In the above example we are checking the number of attachments for the current email message. Only if the email contains an attachment called 'order.pdf' will the Email2DBTrigger flag be set to True (and therefore the Trigger will be fired).

Within the code editor you can access on-line help for all basic functions. Help is context sensitive. Move your cursor to the beginning of the keyword and press F1.

### 15.13.3 Field Extraction Scripts

Field extraction scripts allow you to both control how a field is extracted from the email and format the extracted data in any way you want.




The Extract Script is fired for each field after Email2DB has extracted the field itself. You can then override what Email2DB has extracted by setting the **Email2DBExtract** variable inside your script to whatever value you want assigning to the field.

Whilst editing an Extracted Field click the **Extract Data** tab then click the **Extract Script** option and click the **Edit** button to edit the script for the selected field.

In the above example we are creating a calculated field from existing extracted fields.

You can examine previously extracted fields by using the **FIELDS_Name()** & **FIELDS_Value()** arrays. The **FIELDS_Count** returns the number of fields extracted. You can also change previously extracted fields by changing the FIELDS_Value() array. The FIELDS_Value() array is an array of strings.

As with other scripts you can access all the properties of the current email message using the Email2DBMessage object. You can use this to write a custom field extraction routines by accessing the message content using:

**Email2DBMessage.Body** - returns the plaintext message.
**Email2DBMessage.HTML** - returns the HTML message (if available),
**Email2DBMessage.MessageText** - returns the raw mime text of the original message.

Click the **Object Browser** button to view a list of available properties, or enter 'Email2DBMessage.' in the code editor to view the 'Intellisense' popup menu for the message object.

Within the code editor you can access on-line help for all basic functions. Help is context sensitive. Move your cursor to the beginning of the keyword and press F1.

### 15.13.4 Action Scripts

Action Scripts are executed when an message passes the Trigger Conditions.

Action Scripts can be used to perform custom actions.

For example, suppose you sell software products. Each order you receive you need to send the customer a unique serial number. You could write an action script to generate the serial number (or read it from a file or database) and send an email to the customer (using the SendEmail function) containing his serial number.

See: Run An Email2DB Script Action

### 15.13.5 Processing Attachments Using Scripts

You can use an Action script to process attachments. You could save attachments to specific folders or filenames on your system, or you could perform other processes.

The **Email2DBMessage.Attachments** collection contains an item for each attachment for the message being processed.

**Email2DBMessage.Attachments.Count** contains the number of attachments.

The **Attachment** object contains the following properties:

**ContentType** Content type (eg: text/html).
**Data** The contents of the attachment returned as a string.
**FileExtension** The extension (eg: PDF)
**FileWithoutExtension** the filename without the extension.
**IsAttached** Boolean - True if the attachment is another message.
**MessageLocation** The tempory location of the attachment whilst the message is being processed.
**Name** The filename.
**SavedTo** If the Attachment Save action has been used this will return the full path and filename where the attachment was saved to.
**Size** The Size in bytes.

Example:

```
Sub Main
Dim A As Integer
Dim MyFile As String

    On Error GoTo erl

    ' do nothing if there are no attachments
    If Email2DBMessage.Attachments.Count = 0 Then Exit Sub

    ' create a folder for the customer
    Call CreateFolder("C:\MyFolders\" & FIELDS_Value("customer"))

    ' save attachments
    For A = 1 To Email2DBMessage.Attachments.Count
       MyFile="C:\MyFolder\" & FIELDS_Value("customer") & "\" &
Email2DBMessage.Attachments(A).Name
       FileCopy Email2DBMessage.Attachments(A).Location, MyFile
       Call AddToLog("Attachment saved to: " & MyFile)
    Next A
    Exit Sub

erl:
    ' record error in Email2DB server log so we can see it
    Call AddToLog("Error saving attachment: " & Err.Description)
End Sub

Sub CreateFolder(ByVal Folder As String)
    On Error Resume Next
    If Len(Dir(Folder,vbDirectory))=0 Then MkDir Folder
```

```
End Sub
```

The above example saves attachments to a folder containing the customer name. We are using a field called 'customer' extracted from the incoming email. The script checks if the folder exists first - if not a new one is created. All attachments in the email are then saved to this folder.

The Attachment.SavedTo property will be updated if Email2DB has saved any attachments via the Attachments action. You could then use this to perform further processing.. Or you can use it to store a pointer saved attachments in a database.

You can also use For .. Each style. For example:

```
Dim Attachment As clsMessageAttachment
For Each Attachment In Email2DBMessage.Attachments
       MyFile = Attachment.Name
Next
```

## Related Items

The **Email2DBMessage.RelatedItems** collection will contain a list of Attachment objects for each related item (embedded images, stylesheets etc).

## 15.13.6 Properties & Methods Available In Scripts

The following properties can be accessed in Email2DB Basic Scripts. For the most up to date list click the **Object Browser** button in the script editor.

**General Properties**

| | | |
|---|---|---|
| Email2DBTrigger | Boolean | Set to False in an Extraction or Action script to cancel execution of the trigger (True otherwise) |
| Email2DBExtract | Variant | Set to a value in an Extraction script to set the Email2DB field value. |
| Email2DBOnLastEmail | Boolean | Returns True if Email2DB is processing the last message in the current 'batch'. Useful if you want to run custom script actions after the last message is processed in each run. |
| Email2DBDeleteEmailFromServer | Boolean | Set to True in a Trigger Condition script if you want to force the incoming email to be deleted from the POP3/IMAP server regardless of the Account settings. |
| Email2DBSend | Boolean | Set to True in a conditional email auto-response script if you want the email to be send, False otherwise. |
| Email2DBBatchCount | Long | Returns the number of messages read in the current connection to the POP3, IMAP, Exchange server. |
| Email2DBDisconnect | Boolean | Set to True in a Trigger Condition script if you want to disconnect from the POP3 or IMAP Server. Use this in conjunction with the Email2DBBatchCount property if you only want to download a certain number of messages during each connection. |

**Current Message Object:**

In scripts you can access the current message using the **Email2DBMessage** object. This object has many properties relating to the current message being processed. For example: Email2DBMessage.Body returns the plaintext message body.

**Email2DBMessage Properties:**

| | | |
|---|---|---|
| Account | Object | Contains a reference to the full Account object for the currently executing Account. |
| Attachments | Collection | Contains an item for each attachment. See: Processing Attachments Using Scripts |
| BCC | String | BCC Addresses. |
| Body | String | The plaintext body. This is the text that Email2DB 'sees' when extracting fields. |
| CC | String | CC Addresses. |
| CharSet | String | The character set assigned to the incoming message. |
| ContentType | String | The content type. |
| Dated | Date | The message date. |
| ExtractedFields | Collection | Contains an item for each extracted field. |
| From | String | The from email address. |
| FromIP | String | the IP of the senders email server. |
| FromName | String | From name (if available). |
| GEOCity | String | GEO-IP City data from the senders IP address. |
| GEOCountry | String | GEO-IP Country data from the senders IP address. |
| GEOOrganization | String | GEO-IP Organization data from the senders IP address. |

| | | |
|---|---|---|
| HeaderName() | String Array | Header name for nth header. Eg: Email2DBMessage.HeaderName(1) |
| HeaderNumber() | Long Array | Returns the header number for a given header: Eg: Email2DBMessage.HeaderNumber("myheader") |
| HeaderValue() | String Array | Returns the header value for any header (number or name): Eg: Email2DBMessage.HeaderValue(2) |
| HTML | String | The HTML portion of the message (if any). |
| Importance | String | Importance. |
| InReplyTo | String | Reply to address (from the in-reply-to header). |
| IsRead | Boolean | True if the message has been read (if available from the originating server). |
| LastError | String | Last error message during Trigger execution (if any). |
| LastErrorNumber | Long | Last error number. |
| MessageHeader | String | The full headers of the incoming message. |
| MessageID | String | The unique message identifier. |
| MessageText | String | The raw mime text of the incoming message (excluding headers). |
| NoHeaders | Long | Number of headers. |
| OrigDate | Date | The original message date (from 'orig-date' header). |
| References | String | Reference ID (from 'references' header). |
| RelatedItems | Collection | Contains an item for each embedded attachment. |
| ReplyTo | String | The Reply To email address. |
| ReturnPath | String | The 'return-path' header. |
| Sender | String | The 'sender' header. |
| Sensitivity | String | Sensitivity. |
| Size | Long | The message size in bytes. |
| Source | String | The Account Source. Values are: IMAP, POP3, EWS, RSS, TWITTER, DBPULL, HTTP |
| Subject | String | Subject. |
| ToAddress | String | The To address. |
| Trigger | Object | Contains a reference to the full Trigger object for the currently executing Trigger. |
| ValidateURL | String | Returns the URL to validate the message if teh Validate Action is being used. |
| WordIndex | String | Returns a sorted, space separated list of unique words found in the message, excluding common words. |
| WordIndexWithCount | String | As above but adds a word count before each word. |
| WordIndexWithCountSorted | String | As above but sorted by word count instead of alphabetic. |

**Current Message Properties (Old V2 Style):**

The following properties can still be used.

| | | |
|---|---|---|
| MSG_From | String | From email address |
| MSG_FromIP | String | The IP address of the sender's server. |
| MSG_FromName | String | From Name (if available) |
| MSG_To | String | To address |
| MSG_CC | String | CC address(es) |

| | | |
|---|---|---|
| MSG_BCC | String | BCC address(es) |
| MSG_ReplyTo | String | Reply to address |
| MSG_ReturnPath | String | Return path (address used for bounce replies) |
| MSG_Subject | String | Subject text |
| MSG_Body | String | The plain Body text |
| MSG_HTML | String | The HTML portion of the email (if any). |
| MSG_Dated | Date | Date received |
| MSG_Importance | String | Importance |
| MSG_Sensitivity | String | Sensitivity |
| MSG_Size | Long | Size in bytes |
| MSG_Headers | String | A string containing the full header (to access individual headers see below). |
| MSG_MimeText | String | The complete MIME encoded message text. This is full message text in raw format. It will include all attachments. |
| MSG_UID | String | Unique message identifier |
| MSG_GEOCity | String | GEO-IP City data from the senders IP address |
| MSG_GEOCountry | String | GEO-IP Country data from the senders IP address |
| MSG_GEOOrganization | String | GEO-IP Organization data from the senders IP address |
| MSG_WordIndex | String | Returns a sorted, space separated list of unique words found in the message, excluding common words. |
| MSG_WordIndexWithCount | String | As above but adds a word count before each word. |
| MSG_WordIndexWithCountSorted | String | As above but sorted by word count instead of alphabetic. |
| MSG_Account | Object | Contains a reference to the full Account object for the currently executing Account. |
| MSG_Trigger | Object | Contains a reference to the full Trigger object for the currently executing Trigger. |
| MSG_HeaderCount | Long | Number of headers |
| MSG_Header() | String Array | Header names |
| MSG_HeaderValue() | String Array | Header values |

**Extracted Field Properties**

| | | |
|---|---|---|
| FIELDS_Count | Long | Number of fields extracted |
| FIELDS_Name() | String Array | Field name |
| FIELDS_Value() | String Array | Field value. Field name and number can be passed.<br>Eg: City = FIELDS_Value("City") or City = FIELDS_Value(2)<br>This property is read/write allowing you to change the content.<br>Eg: FIELDS_Value("City") = "New York" |
| FIELDS_InsertField() | String Array | Returns the SQL database field name that this field is mapped to. Can be accessed by field name or index as above. Read/write |

| | | |
|---|---|---|
| FIELDS_InsertInto() | String Array | Returns the SQL Table name. Read/write |

**SQL Statement Properties**

The SQL statement is built as fields are extracted. You can use the properties below in an Action script to modify the SQL that will be executed against the data source.

| | | |
|---|---|---|
| SQL_SelectStatement | String | Returns the current SQL SELECT statement(s) built during field extraction. Read/write |
| SQL_InsertStatement | String | Returns the current SQL INSERT statement(s) built during field extraction. Read/write |
| SQL_UpdateStatement | String | Returns the current SQL UPDATE statement(s) built during field extraction. Read/write. |
| SQL_Connection | String | Returns the Connection string. Read/write |

## Methods

The following methods are available in scripts for you to use in addition to all the built-in BASIC functions.

| | |
|---|---|
| **SendEmail**( _<br>String ToAddress, _<br>String FromAddress, _<br>String Subject, _<br>String MessageText) | Sends an email via the SMTP mail server configured in the settings.<br>Eg:<br><br>Call SendEmail("test@yahoo.com","test@mysite.com","Test Message","Test") |
| **SMTPSend**( _<br>String From, _<br>String To, _<br>String CC, _<br>String Bcc, _<br>String Subject, _<br>String Body, _<br>Boolean Important, _<br>String Attachments, _<br>String HTML, _<br>Boolean Receipt) | A more advanced send email method.<br><br>Set Important = True to mark the message as important. Set Receipt to True to request a read receipt to be sent to the 'From' address. Set the HTML parameter if you want the mail to be sent in HTML format.<br><br>The Attachment parameter is a string containing a comma separated list of files to attach. |
| **SendScheduledEmail**(_<br>Date SendOnDateTime, _<br>String From, _<br>String To, _<br>String CC, _<br>String Bcc, _<br>String Subject, _<br>String Body, _<br>Boolean Important, _<br>String Attachments, _<br>String HTML,<br>Boolean Receipt) | Adds an email to the Scheduled Send database. This method allows you to create email messages that will automatically be sent at some point in the future.<br><br>The parameters are as above.<br><br>Set the SendOnDateTime parameter to the date and time that you want this email to be sent.<br><br>The Attachment parameter is a string containing a comma separated list of files to attach. If you are attaching files, ensure that the files will exist on the future date. |

| | |
|---|---|
| **SMSSend**( _<br>String SMSFrom, _<br>String SMSTo, _<br>String SMSMessage) | Sends an SMS message. Specify the phone/pager from and to numbers in full international format without the leading +, eg: 44779912345678.<br><br>Note: You must setup an SMS account to use and specify the SMS account user name & password in the Email2DB main settings before SMS messages can be sent. |
| **DeleteScheduledEmail**(<br>String ToAddress) | Deletes pending messages from the Scheduled Send database with the give To Address.<br><br>This method is useful in 'bounced email' triggers. It allows you to remove pending messages for emails that have bounced. |
| **RunProcess**( _<br>String FileName, _<br>String Parmeters, _<br>Optional Boolean WaitForFinish) | Runs the specified process and passes the specified parameters. If the WaitForFinish parameter is True then Email2DB will wait for the process to complete before continuing.<br><br>Eg:<br>Call RunProcess("C:\Accounts\Neworder.exe","%orderno%") |
| **SetEnvironmentVar**( _<br>String VarName, _<br>String Value) | Sets the given environment variable to the give value.<br><br>Eg:<br>Call SetEnvironmentVar("orderno","%order%") |
| **AddToLog**( _<br>String Message) | Adds a message to the Email2DB Service log. This method is useful for debugging so you can track what your scripts are doing. |
| Boolean **SetRegistry**( _<br>String Section, _<br>String Setting, _<br>Variant Data) | Saves a setting to the registry. Settings are stored in HKEY_LOCAL_MACHINE\Software\ParkerSoft\Email2DB<br><br>You can create sub-sections beneath this.<br>Use the GetRegistry function to retrieve settings. |
| Variant **GetRegistry**( _<br>String Section, _<br>String Setting, _<br>Variant Default) | Gets a value from the registry. Values will be retrieved from the HKEY_LOCAL_MACHINE\Software\ParkerSoft\Email2DB section. |
| String **HTTPGet**( _<br>String URL) | Reads a web page from the given URL. The web page is read via HTTP and returned as a string value. |
| Variant **AccountVariable**<br>(String Name) | Gets or sets an Account-wide variable. This variable will be visible to any script executing within the same Email2DB Account. Variables created with the Set Variable Action can be accessed this way.<br><br>Eg:<br>LastCustomer = AccountVariable("Customer") |
| Variant **GlobalVariable**<br>(String Name) | Gets or sets a system-wide global variable. This variable will be visible to all other scripts during the current Email2DB Server session. This is useful for maintaining counters, or values that must remain between executions of scripts. |

| | |
|---|---|
| | Eg:<br>LastSerial = GlobalVariable("LastSerial")<br>LastSerial = LastSerial + 1<br>GlobalVariable("LastSerial") = LastSerial |
| Boolean **GetGEOIP**( _<br>String IPAddress) | Gets the GEO-IP information for a given IP address. The GEO-IP data is returned in the GEOIP Structure. Access the GEOIP structure after calling this method to access the returned GEO-IP data.<br><br>Eg:<br>If GetGEOIP("87.123.12.1") = True Then<br>MsgBox "GeoIP Data: " & GEOIP.Country, GEOIP.City, GEOIP.Organization<br>End If |
| **PassToTrigger**( _<br>String TriggerName, _<br>String Headers, _<br>String BodyText) | Executes a given Trigger with the given message body text and headers. This method enables you to pass any message text to any trigger for subsequent execution. You must pass the Trigger 'Name'. If the same Trigger name exists for multiple Accounts then the last Trigger found for the given name will be used.<br><br>Eg:<br>Call PassToTrigger("Orders Trigger","From: me@somewhere.com","execute this") |
| **PrintPDFFile**( _<br>String FileName, _<br>Optional String PrinterToUse) | Prints the given PDF file to the printer specified. |
| **ExtractPDFFileToText**( _<br>String PDFFileName) As String | Creates a plain text version of the given PDF File. Plain text is returned as a single string. |
| Long **LookFor**( _<br>String FindIn, _<br>String FindWhat, _<br>Optional Long StartFrom, _<br>Optional Boolean CaseSensitive, _<br>Optional Boolean UpTo) | Searches string FindIn for occurrence of FindWhat. FindWhat can make use of Email2DB Regular expression patterns. Returns the position within the string.<br><br>The search starts from the beginning of the string unless StartFrom is specified.<br><br>If UpTo is True then the position returned will be at the end of FindWhat.<br><br>If FindWhat is * then first non control character is searched for.<br><br>See: Regular Expressions |

## 15.13.7 Mail Server Event Scripts

The Email2DB mail server can fire scripts when emails are received or relayed.

The script exposes the current message via the MSG object. This contains properties for all parts of the message.

### Incoming Messages

This script can be used to perform your own checks on incoming messages, you can make changes to the message (via the MSG object) and you can reject it (by setting theEmail2DBAccept flag to false).

Enable the Fire Script When A Message Arrives to enable the incoming message script. Click Edit to edit the script.

### Relayed Messages

This script fires whenever the mail server relays a message (IE, sends a message to another mail server). You can make changes to the message (via the MSG object) and you can cancel the send by setting the Email2DB Email2DBAccept flag to false.

## 15.14 Developing Custom Interfaces To Email2DB

The Email2DB Administrator connects to the Email2DB Server using a secure HTTP connection. The Email2DB Server acts as a web server for reading and updating Email2DB Accounts & Triggers (MetaData).

You can use this web server to create your own version of the Email2DB Administrator to view & update your Accounts & Triggers and for viewing the Email2DB Message Store.

Any platform & development language can be used - it just needs to be able to read and post web pages over a secure SSL connection.

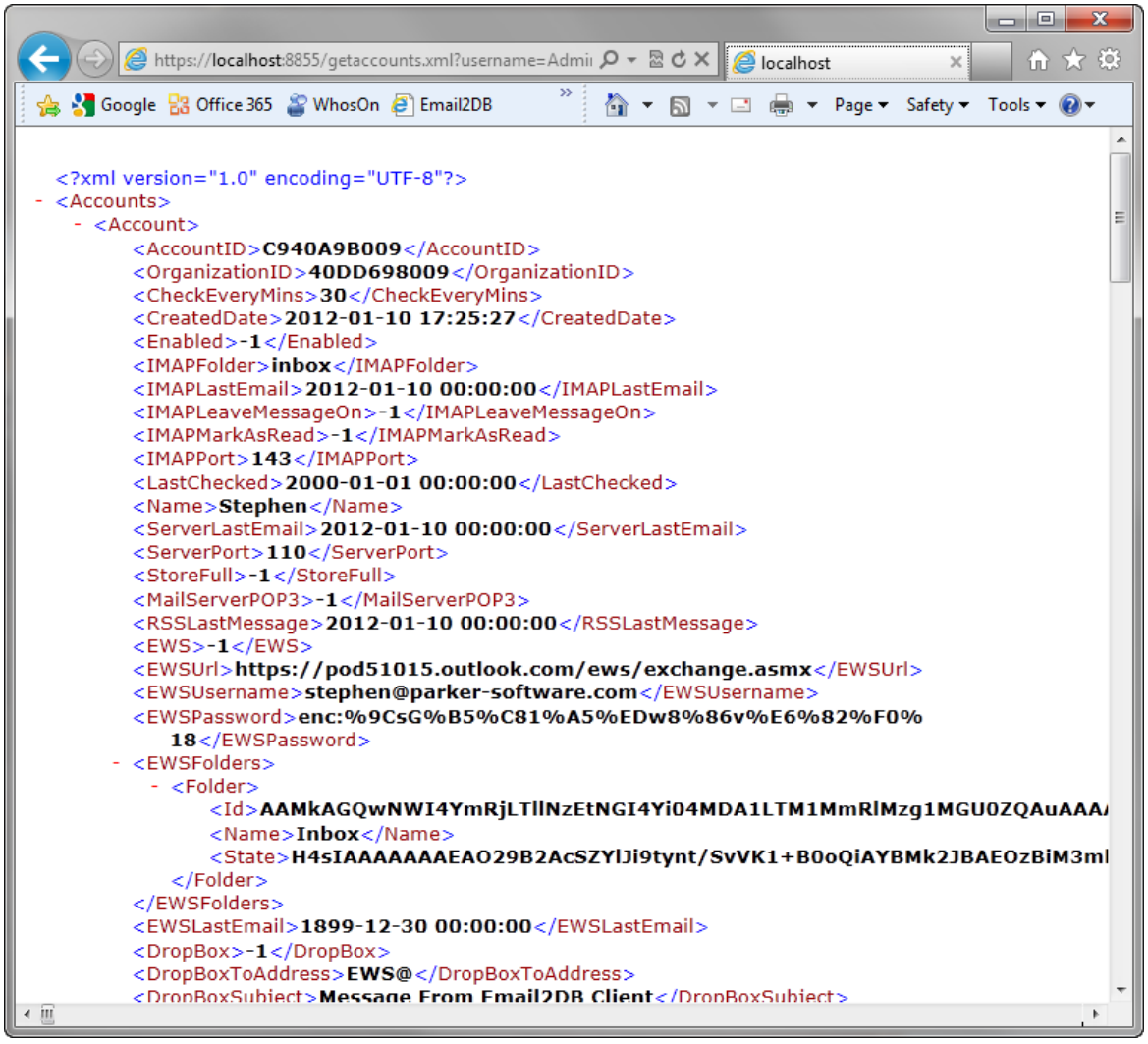
You can test the Email2DB Web Server via your web browser.

Open a web browser and navigate to: `https://{Email2DBServer}:8855/getaccounts.xml?username={username}&password={passwordhash}`

The **Email2DBServer** is the host name or IP address of the computer running Email2DB. If this is the local machine then 'localhost' can be specified.

The **username** & **password** parameters MUST be specified on every GET and POST. The password MUST be specified as the MD5Hash (base64 encoded) of the user's actual password.

The web browser will then display the XML returned:




The Accounts returned will only be the Accounts that the given user has access to.

All passwords used on any Email2DB Settings will always be encrypted in the XML.

### 15.14.1 Reading Accounts & Triggers

Email2DB Accounts & Trigger settings can be retrieved from the server using the following page requests.

The username & password parameters must be passed with each request in the format: &username={username}&password={password}. Where the password is the Base64 encoded, MD5 hash of the password.

| Page Request | Parameters | Returns |
|---|---|---|
| getallaccounts.xml | | XML for all Accounts for the given user. |
| getaccount.xml | accountid={accountid} | XML for the given Account ID and user. |
| gettriggers.xml | accountid={accountid} | XML for all Triggers for the given Account ID and user. |
| gettrigger.xml | accountid={accountid} &triggerid={triggerid} | XML for the given Account & Trigger ID's and user. |
| gettriggerconditions.xml | accountid={accountid} | XML for all Triggers for the given AccountID and user. This request returns just the Trigger details for each Trigger. It does not include all of the Actions and Extracted Fields. This will reduce the size of the XML considerably. You can use it when creating a browser application. You could then use the gettrigger.xml request when you need to retrieve the full trigger settings. |

Example:
`gettrigger.xml?accountid=C940AB009&triggerid=A167B29823&username=Admin&password=`

Each Account has a unique AccountID field. This will be returned in the XML for each Account in the <AccountID></AccountID> tag. Each Trigger has a <TriggerID> field.

## 15.14.2 Posting Accounts & Triggers

You can update any Email2DB Account or Trigger using a HTTP POST.

The content type should be set to text/xml. Any XML tags that are missing will be assumed to equate to Zero for Numeric properties, False for Boolean and Blank for Strings.

If an AccountID or TriggerID is passed then the existing Account or Trigger record will be updated. Otherwise a new Account or Trigger will be added.

| POST Request | Parameters | Returns |
|---|---|---|
| postaccount.xml | | <result>OK</result> |
| posttrigger.xml | accountid={accountid} | <result>OK</result> |

### 15.14.3 Management Requests

| Page Request | Parameters | Action | Returns |
|---|---|---|---|
| deleteaccount.xml | accountid={accountid} | Deletes an Account. | text/plain OK or Error:{msg} |
| deletetrigger.xml | accountid={accountid} &triggerid={triggerid} | Deletes a Trigger from the given Account. | text/plain OK or Error:{msg} |
| enableaccount.xml | accountid={accountid} | Enables an Account | text/plain OK or Error:{msg} |
| disableaccount.xml | accountid={accountid} &triggerid={triggerid} | Disables an Account | text/plain OK or Error:{msg} |
| enabletrigger.xml | accountid={accountid} &triggerid={triggerid} | Enables a Trigger | text/plain OK or Error:{msg} |
| disabletrigger.xml | accountid={accountid} &triggerid={triggerid} | Disables a Trigger | text/plain OK or Error:{msg} |
| checknow.xml | accountid={accountid} [&triggerid={triggerid}] | Forces Email2DB to check for new messages for the specified Account and optional Trigger | text/plain OK or Error:{msg} |

# Part

# 16 Best Practices

This topic describes some of the Email2DB best practices highlighted by the Email2DB support team:

**1. Email2DB Message Store**

We recommend using SQL Server 2008 or 2012 Express for the Email2DB Message Store. This should be installed on the same computer as the Email2DB Server. A standard installation of SQL Server 2008/2012 Express will suffice, you should not need to change any of the default installation properties.

If you are using the Data Center Edition of Email2DB then the database will need to be accessible from any Edge Server or Message Processor Server nodes. You need to setup your database security accordingly. Parker Software Professional Services can assist with the configuration.

**2. Email Access**

The fastest way to process new emails is to have them sent directly to the Email2DB Mail Server. These will be processed as they are received.

If you are reading emails from another mail server then IMAP should always be used if it is available. This will provide the best overall performance.

# Part XVI

# 17 Troubleshooting

*Q. My database isn't getting updated, but the Test button works?*
A. This is probably a security issue. If the database is on a separate PC to Email2DB then the Email2DB service may not have permission to access it. Try creating a specific user in your database and then specifying the user name and password on the connection string. Note, that when you click the Test button the database is opened under your 'user' context. When the Email2DB Service tries to open it the database is opened under the service account context.

*Q. The CSV File doesn't get updated?*
A. Run the Email2DB Service under a user that has permission to write to the folder where the CSV file should be saved. See: Running the Email2DB Service Under A Different User.

*Q. The Report Does Not Get Printed When I Use A Network Printer?*
A. See: Running The Email2DB Service Under A Different User.

# Index

## - A -



## - B -



## - C -

# - D -



# - E -

# - F -



# - G -



# - H -



# - I -



# - J -



# - L -



# - M -

# - N -



# - O -



# - P -



# - Q -



# - R -

# - S -



# - T -



# - U -

# - V -



# - W -



# - X -