



**realisable**

**REALISABLE IMAN**

**QUICK START GUIDE**

© 2011 Realisable Software Ltd. All rights reserved.

Realisable Software, Realisable Software logos, and all Realisable Software product and service names are registered trademarks or trademarks of Realisable Software, Ltd., or its affiliated entities. All other trademarks are the property of their respective owners.

Realisable Software Ltd.

Publisher

No part of this documentation may be copied, photocopied, reproduced, translated, microfilmed, or otherwise duplicated on any medium without written consent of Realisable Software Ltd.

Use of the software programs described herein and this documentation is subject to the Realisable Software License Agreement enclosed in the software package.

# CONTENTS

Document Purpose .....	1
Install Prerequisites .....	2
Software requirements .....	2
IMan Client Requirements .....	14
Installation .....	15
Creating IMan IIS Application Pool .....	15
Installing Realisable IMan .....	17
IMan Database Setup .....	22
Sage300 (Accpac) - Setting Security .....	26
Setting Up IMan .....	28
Using the Integration Designer .....	34
Sample Integrations Walkthrough .....	36
SAMACCIMP (Sample Sage300 O/E Order Import) & SAMS200IMP (Sample Sage200 SOP Order Import) .....	36
SAMACCEXTRCT (Sample Item & Price Extract) & SAMS200EXTRT (Sample Sage200 Item & Price Extract) .....	45
SAMACCARINV (Sample Sage300 A/R Invoice Import) .....	48
SAMACCARRCP (Sample Sage300 A/R Receipt Import) .....	49
SAMACCCPTIME (Sample Sage300 Canadian Payroll Import) .....	50
SAMACCPJCTIME (Sample Sage300 PJC Timecard Import) .....	54
SAMACCINTCO (Sample Sage300 Intercompany Processing Automation) .....	55
SAMS200INVIMP (Sample Sage200 Sales Invoice Import) .....	71
SAMS200CONSPRNT (Sample Sage200 SOP Invoice & Consolidated Billing) .....	74
SAMSAGECRMIM (Sample Sage CRM Opportunity Quote Import) .....	81
SAMSAGECRMEX (Sample SageCRM Export) .....	90
Scheduling and Running Integrations .....	92

## DOCUMENT PURPOSE

---

This document describes the steps required to install, setup and run IMan. This document is not a complete instruction manual and the Administration Guide, User Guide and various application specific documents should be consulted.

The latter section of the manual provides a walkthrough of the sample integrations bundled within IMan.

# INSTALL PREREQUISITES

---

## SOFTWARE REQUIREMENTS

---

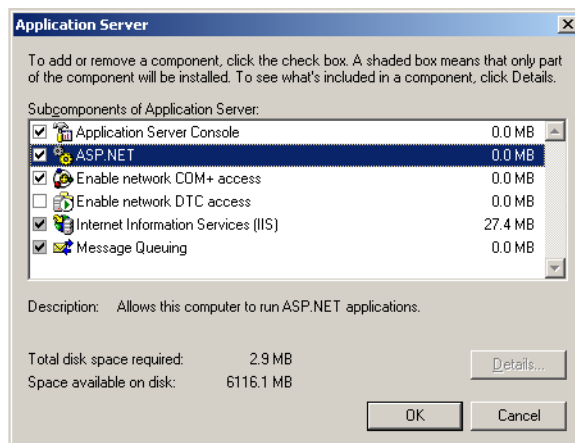
The following section describes system software and windows components that must be installed on the computer on which Realisable IMan is installed.

Realisable will support installations on Microsoft Windows Server 2003, 2008 & 2012, Windows Vista, Windows 7 & Windows 8 as per the specifications below.

We highly recommend for live deployments that IMan is installed on a Server (2003, 2008, 2012) product.

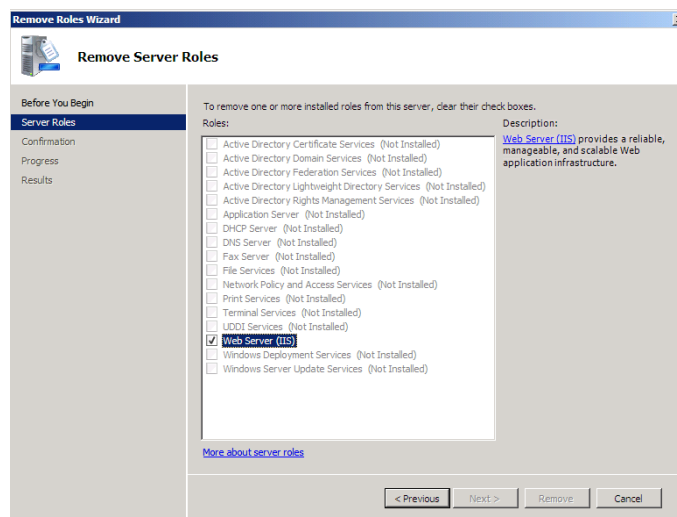
### Windows Server 2003

Operation System	Windows Server 2003 Service Pack 2
Memory	Minimum 1GB; more required depending on job complexity and dataset size.
.NET Version	.NET Framework 3.5 Service Pack 1 or Later .NET Framework 2.0 Service Pack 2 or Later
XML	MSXML 6.0 SP2 (As of writing this now part of Windows Updates)
Windows Components	Application Server Application Server Console ASP.NET Internet Information Services (IIS) -> World Wide Web Service Message Queuing -> Common (Active Directory Integration Not Required)



## Windows Server 2008

Operation System	Windows Server 2008 (R2) Service Pack 2
Memory	Minimum 1GB; more required depending on job complexity and dataset size.
.NET Version	.NET 3.5 Framework Service Pack 1 or Later .NET Framework 2.0 Service Pack 2 or Later
Roles	Web Server (IIS)



### Common HTTP Features

- Static Content
- Default Document
- HTTP Errors

### Application Development

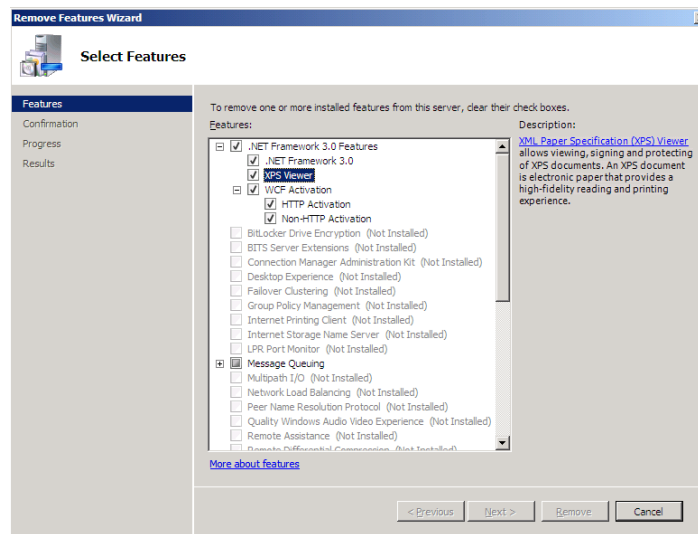
- ASP.Net
- .NET Extensibility
- ISAPI Extension
- ISAPI Filters

#### Management Tools

- IIS Management Console
- IIS 6 Management Compatibility (All)

#### Features

#### .NET Framework 3.5.1 (All)



#### Message Queuing

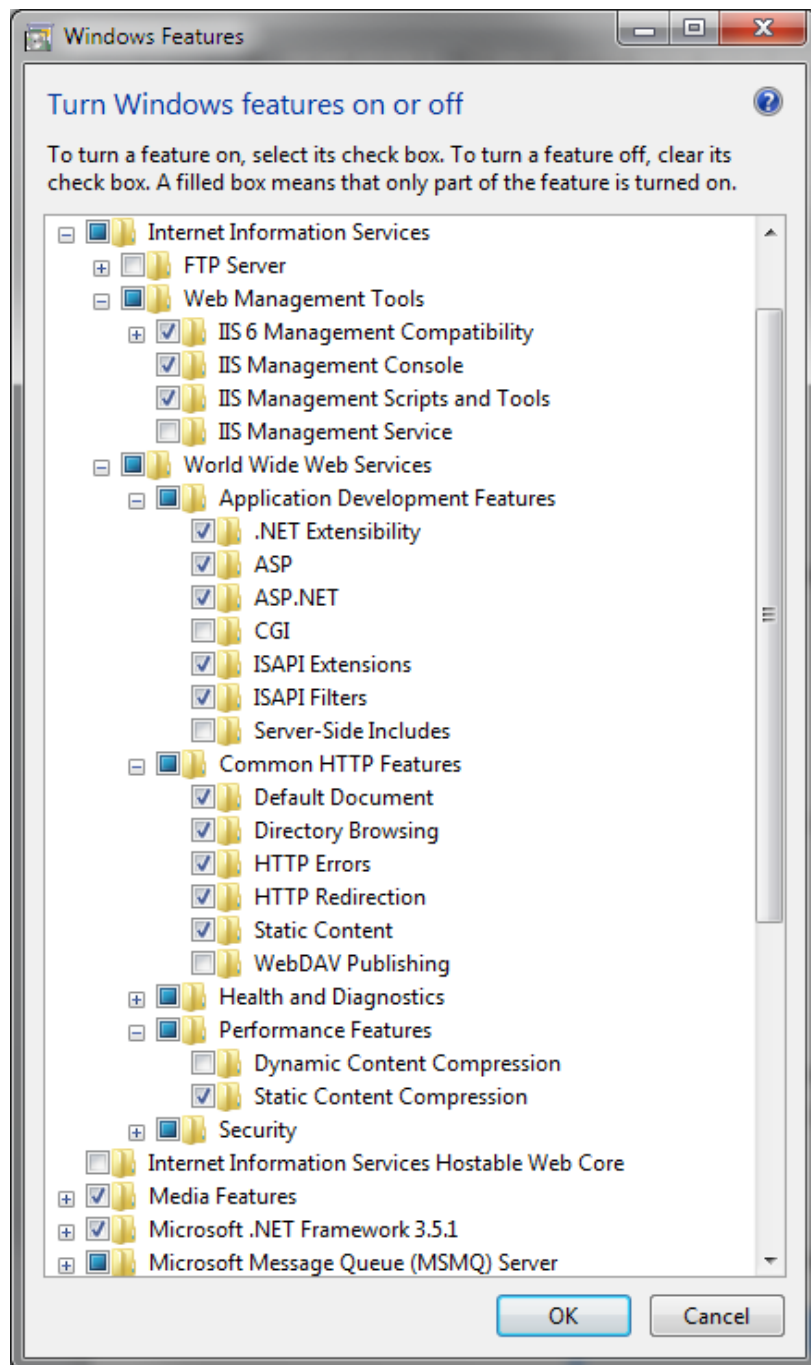
- Message Queue Server (Active Directory Integration Not Required)

#### Windows Process Activation

- All

## Windows 7 & Windows Vista

Operation System	Windows Vista SP2 or Windows 7 SP 1
Memory	Minimum 1GB; more required depending on job complexity and dataset size.
.NET Version	.NET 3.5 Framework Service Pack 1 or Later .NET Framework 2.0 Service Pack 2 or Later
Windows Features	Internet Information Services





#### Common HTTP Features

- Static Content
- Default Document
- HTTP Errors

#### Health & Diagnostics

- No Requirements

#### Performance

- Static Content Compression

#### Security

- No Requirements

#### Application Development

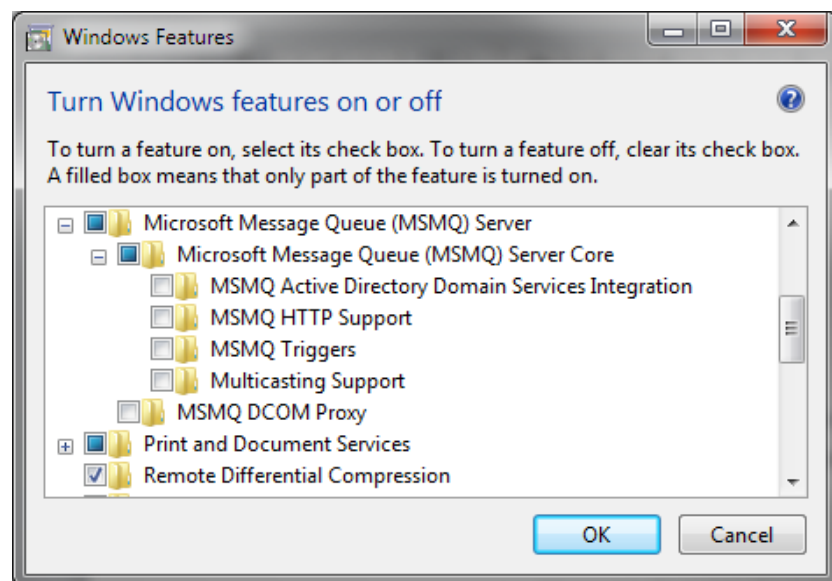
- .Net Extensibility 3.5
- ASP.Net 3.5
- ISAPI Extension
- ISAPI Filters

#### Management Tools

- IIS Management Console
- IIS 6 Management Compatibility (All)

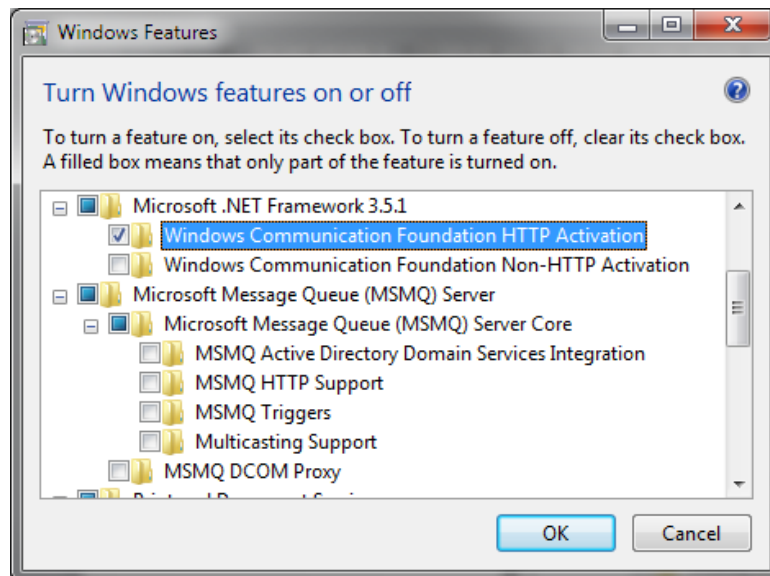
#### Message Queuing

- Message Queuing Server (Active Directory Integration Not Required)



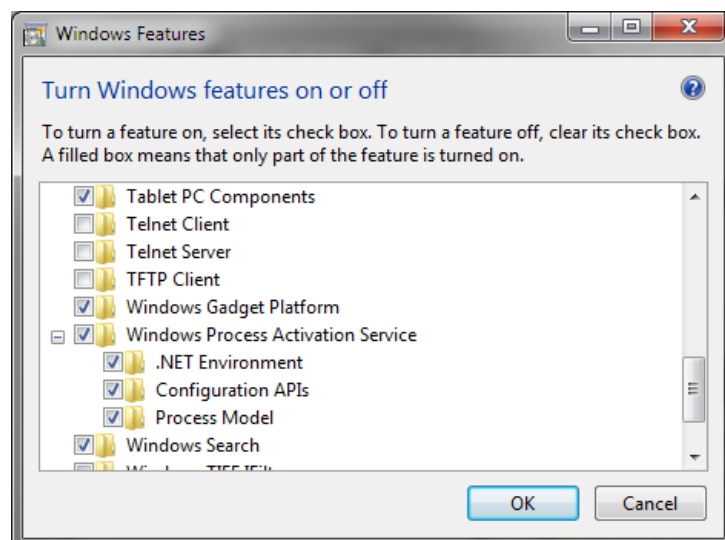
## Microsoft .Net Framework 3.5.1

- Windows Communication Foundation HTTP Activation



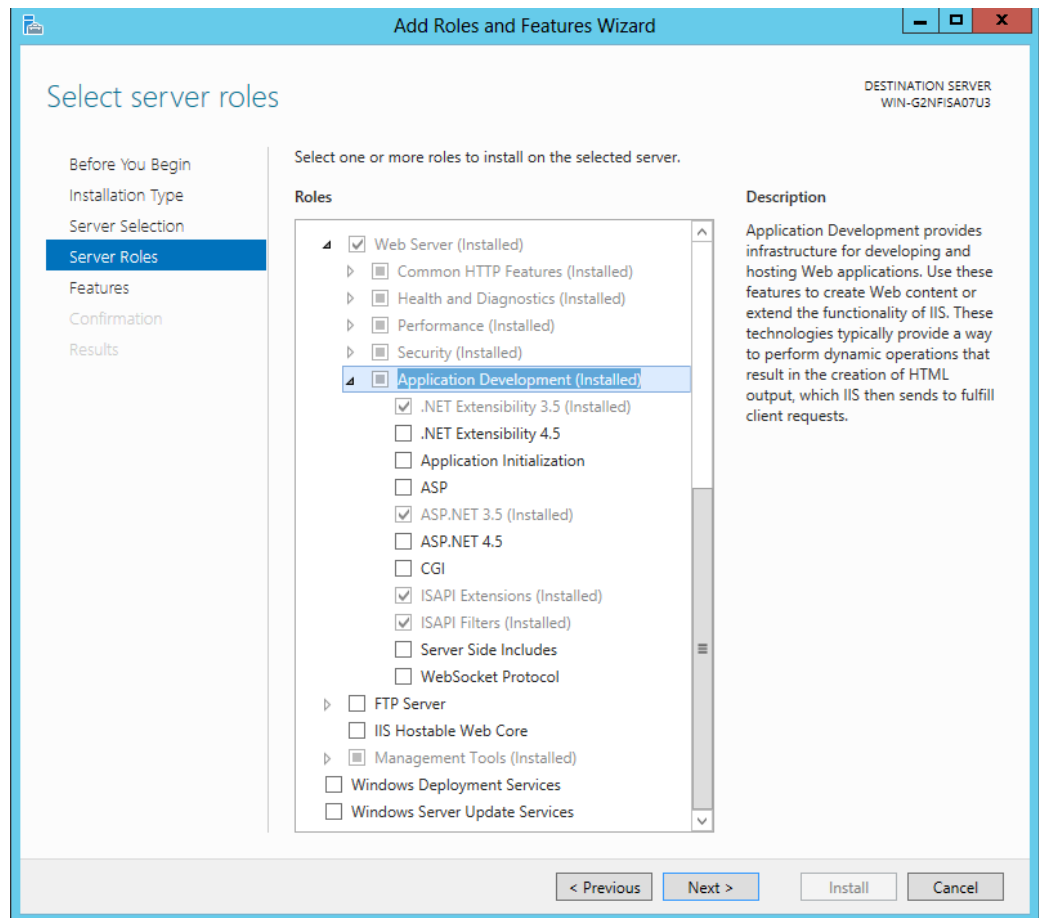
## Windows Process Activation

- All



## Windows Server 2012

Operation System	Windows Server 2012
Memory	Minimum 1GB; more required depending on job complexity and dataset size.
.NET Version	.NET 3.5 Framework (See below).
Roles	Web Server (IIS)



### Common HTTP Features

- Static Content
- Default Document
- HTTP Errors

### Health & Diagnostics

- No Requirements

### Performance

- Static Content Compression

## Security

- No Requirements

## Application Development

- .Net Extensibility 3.5
- ASP.Net 3.5
- ISAPI Extension
- ISAPI Filters

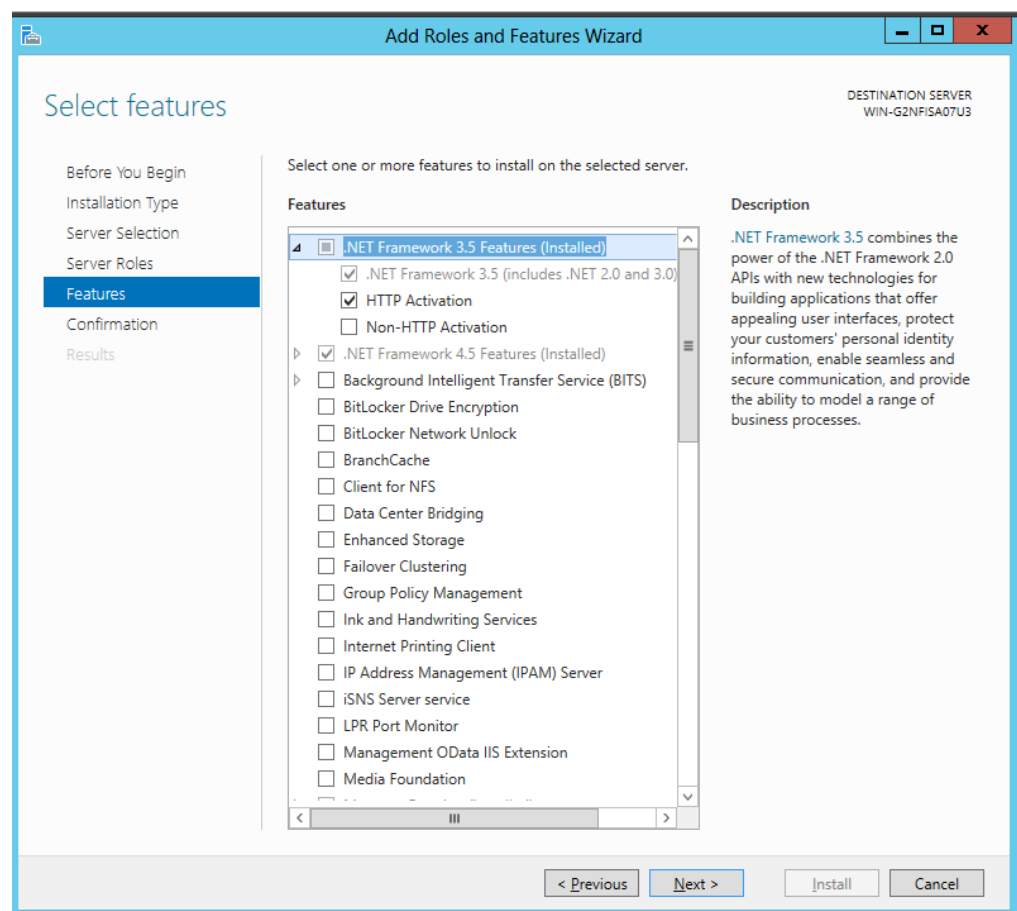
## Management Tools

- IIS Management Console
- IIS 6 Management Compatibility (All)

## Features

### .NET Framework 3.5

- .NET Framework 3.5
- HTTP Activation



## Message Queuing

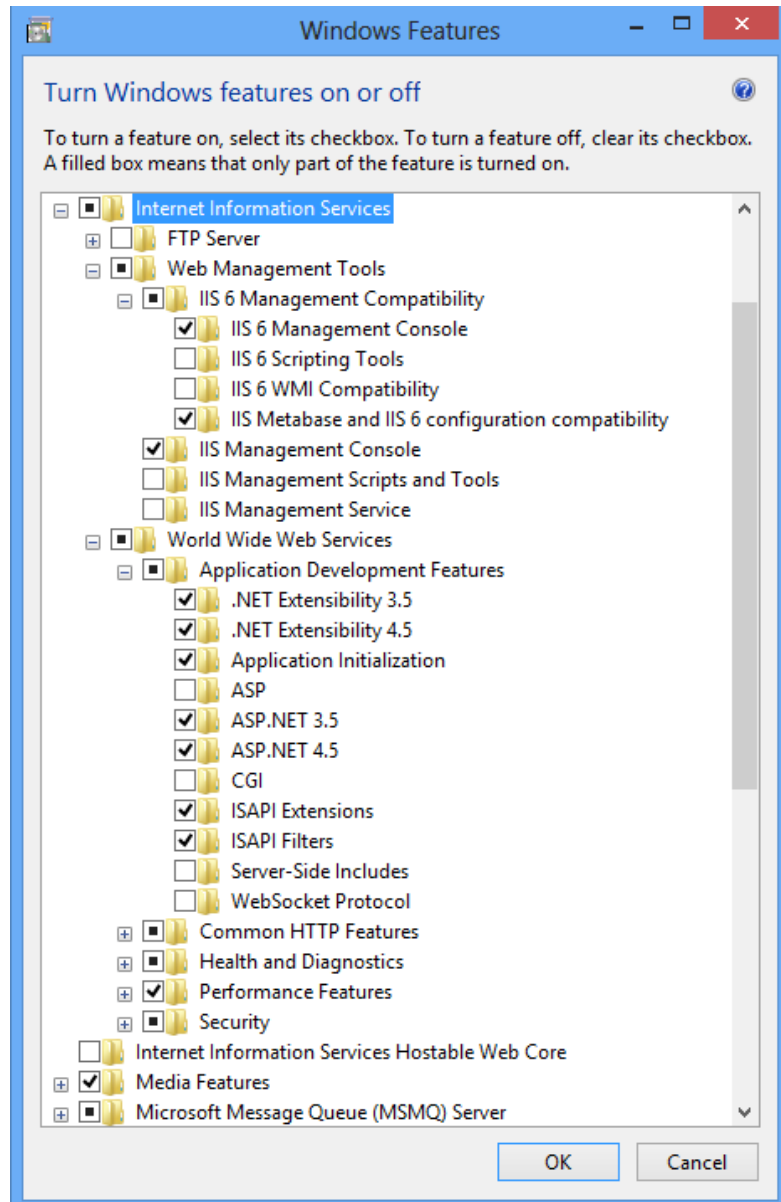
- Message Queuing Server (Active Directory Integration Not Required)

#### Windows Process Activation (All Options)

- Process Model
- .NET Environment 3.5
- Configuration APIs

## Windows 8

Operation System	Windows 8
Memory	Minimum 1GB; more required depending on job complexity and dataset size.
.NET Version	.NET 3.5 Framework (See below).
Windows Features	Internet Information Services



### Common HTTP Features

- Static Content
- Default Document
- HTTP Errors

#### Health & Diagnostics

- No Requirements

#### Performance

- Static Content Compression

#### Security

- No Requirements

#### Application Development

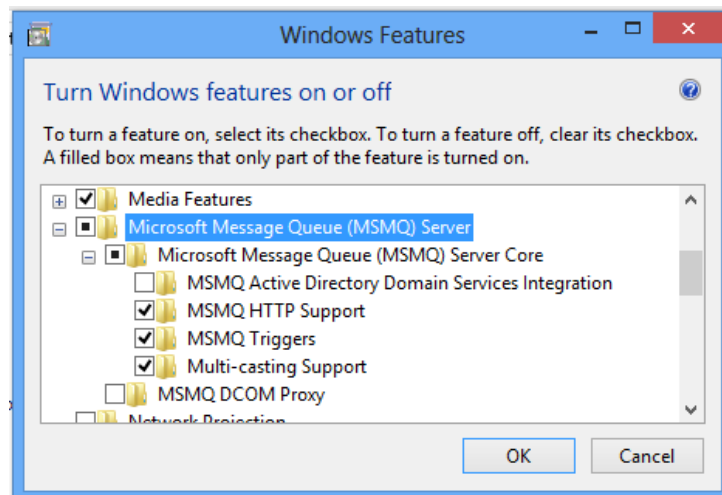
- .Net Extensibility 3.5
- ASP.Net 3.5
- ISAPI Extension
- ISAPI Filters

#### Management Tools

- IIS Management Console
- IIS 6 Management Compatibility (All)

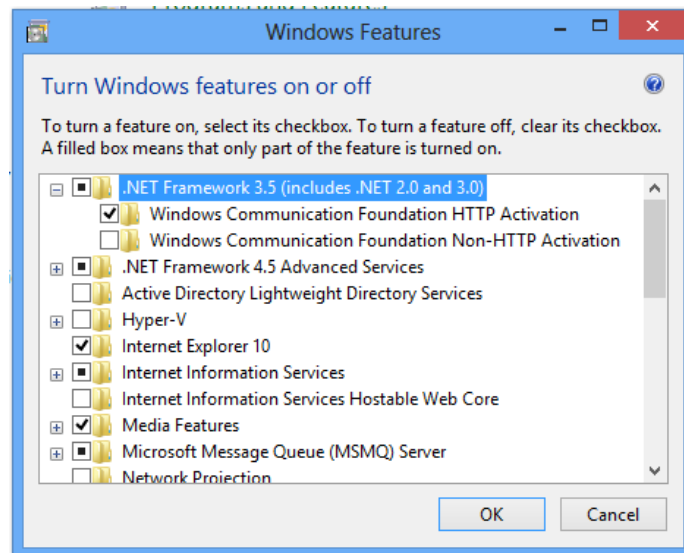
#### Message Queuing

- Message Queuing Server (Active Directory Integration Not Required)



#### .NET Framework 3.5

- Windows Communication Foundation HTTP Activation



## Server Virtualisation

IMan has been tested and is fully supported on VMWare. If an issue is shown to be a problem with VMWare, the customer may be referred to VMWare support.

Other virtualisation products such as Microsoft's Virtual PC or Virtual Server, or Citrix' XEN Server are assumed to work, but have not been explicitly tested.

For these other virtualisation products Realisable will provide support for issues that either are known to occur on the native OS, or can be demonstrated not to be as a result of running on virtualisation software. If a problem is a known IMan issue, Realisable Support will recommend the appropriate solution on the native OS. If that solution does not work in the virtualised environment, the customer will be referred to the support of the virtualisation software. If the problem is determined not to be a known IMan issue, we will refer the customer to the support of the virtualisation software. When the customer can demonstrate that the issue occurs when running on the native OS, Realisable will resume support under its standard support terms.



## IMAN CLIENT REQUIREMENTS

---

All access to IMan is performed from a web browser. For designing jobs, we recommend a relatively fast workstation located on the same local network as the server.

The following web browsers are supported:

- Internet Explorer 8 or above (IE 9 or 10 are preferred due to performance and script handling enhancements)
- Mozilla Firefox 3.0 and above.
- Google Chrome

# INSTALLATION

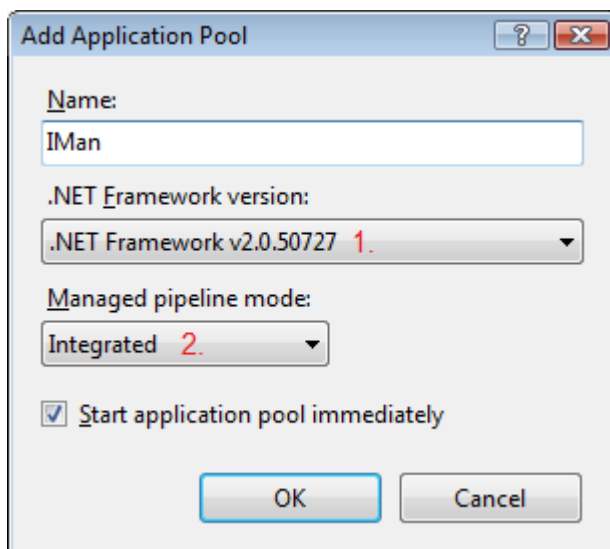
---

## CREATING IMAN IIS APPLICATION POOL

---

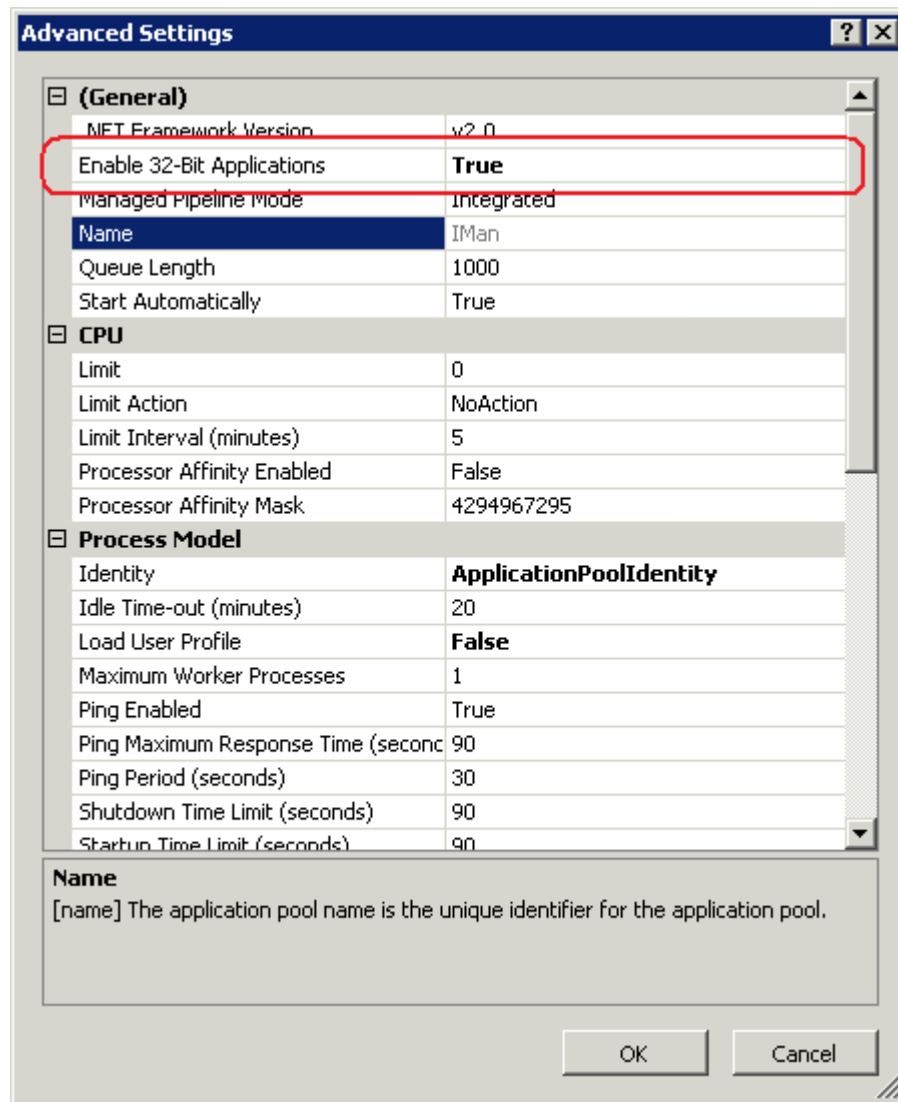
It is recommended that an application pool dedicated to IMan is created. This partitions IMan from other IIS based applications, preventing it from causing resource conflicts with those applications and vice versa.

1. Open the IIS Management Console. (Start, Control Panel, Administrative Tools, IIS Manager).
2. Expand the computer name on the left hand menu and click 'Application Pools'.
3. In the right hand panel, right click and 'Add Application Pool'.
4. Enter details as displayed below; ensuring that '.Net Framework v2.0.50727' & 'Integrated' is select.



Press Ok when complete.

5. If the server is 64-Bit, right click the Application Pool and go Advanced Settings.
6. Set Enable 32-Bit Applications to True, then OK.



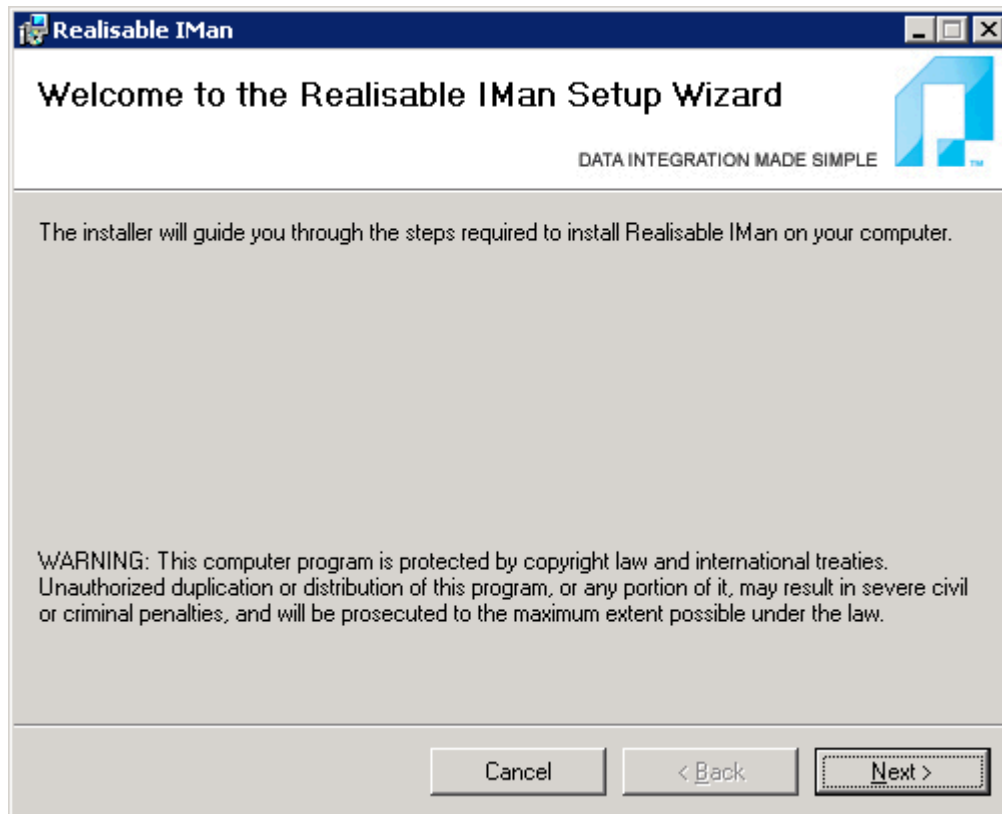
## INSTALLING REALISABLE IMAN

---

1. Start the IMan installer by running the IManInstaller.msi file.

If User Access Control (UAC) is enabled on the workstation or server you may receive the following error. To remedy, run the accompanying setup.exe file (bundled with the installer) as the Administrator.

2. When prompted, press Next to start the installation.

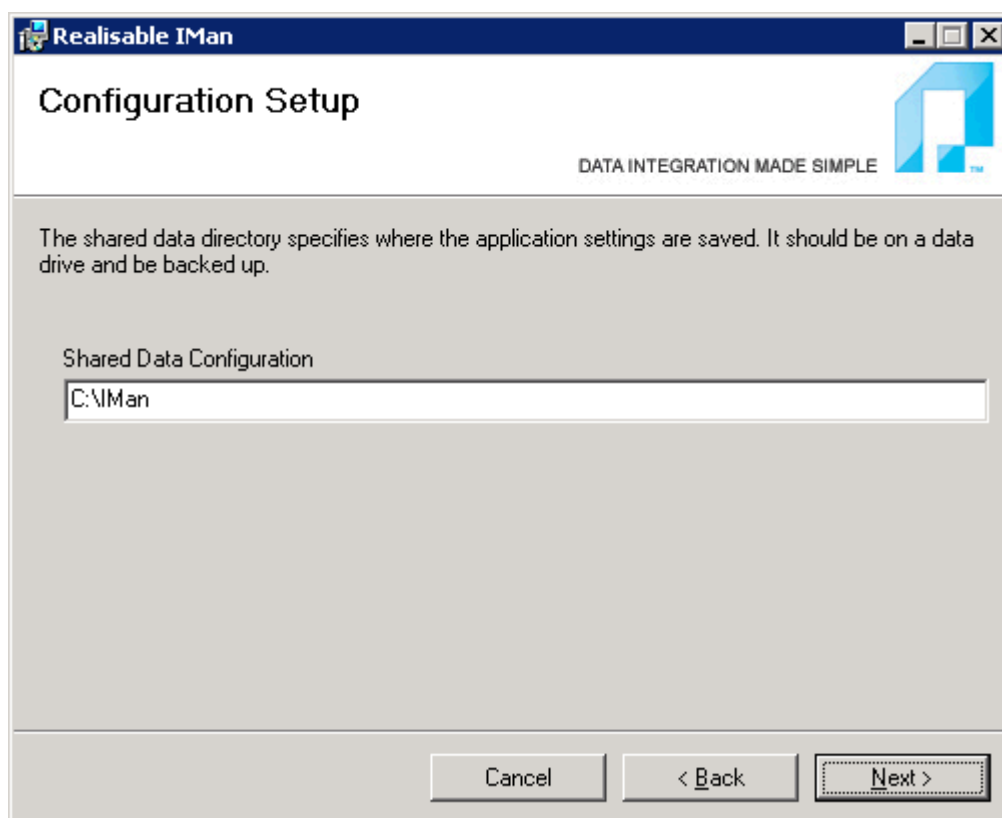


3. Accept or reject the license agreement.

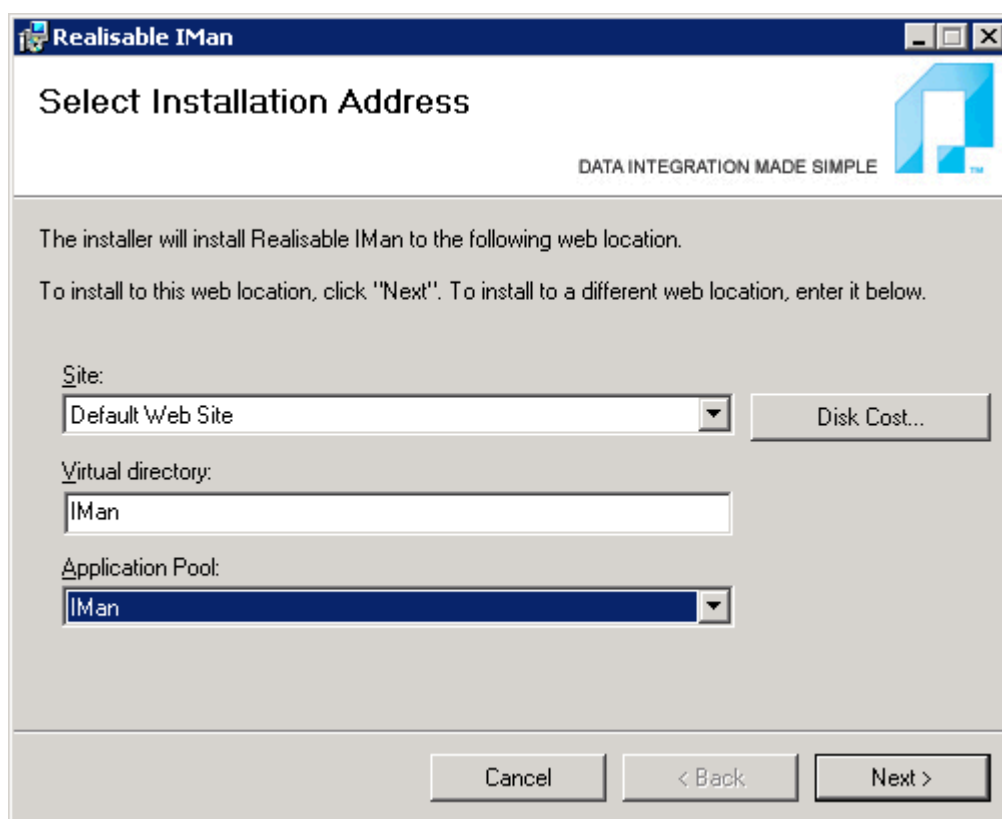


4. Enter a location to store the IMan configuration files. The directory should be path that is local to the machine. It is recommended that the path be located as a subdirectory to the root of a drive and be named IMan, as shown below.

This folder contains the job configuration files and requires backing up.

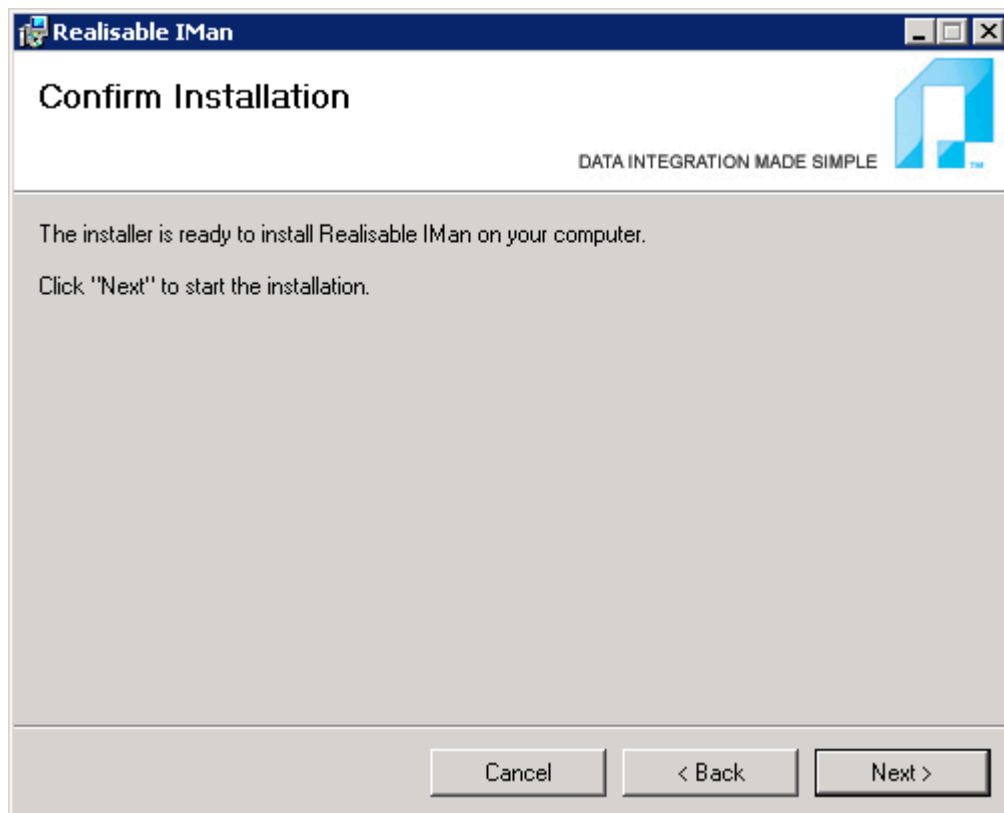


5. Select the Website, the Virtual directory and the application pool the web interface portion of IMan will use.

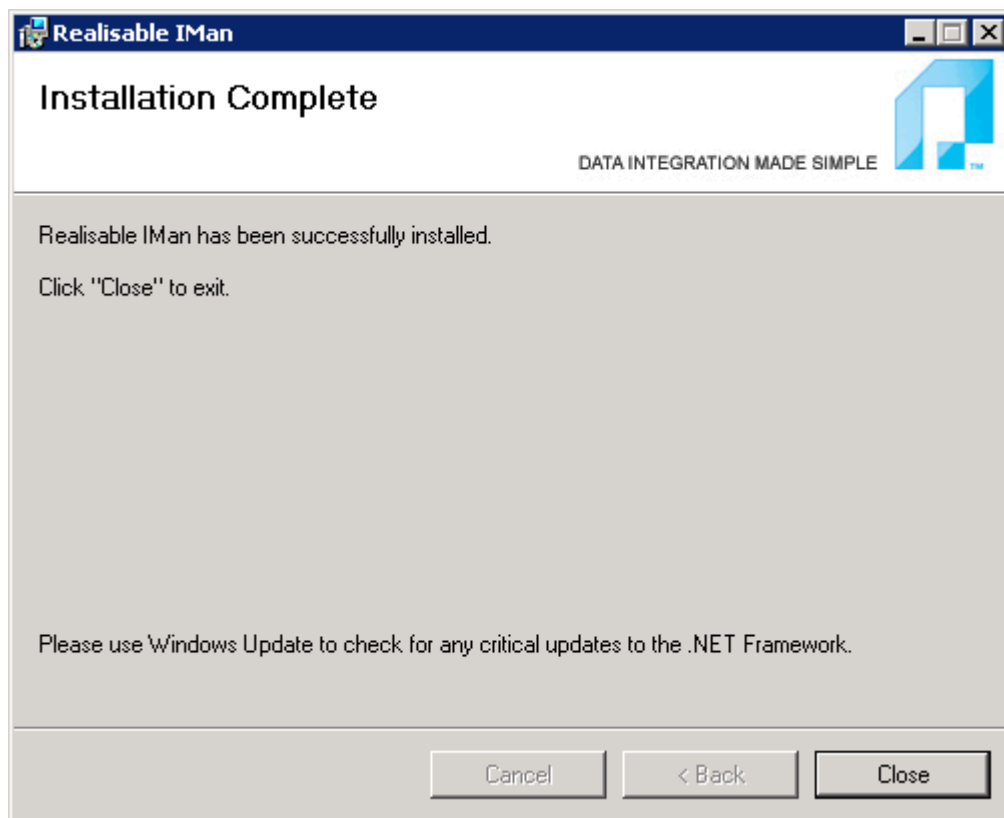


Site	This is the website defined in IIS to install to.
Virtual Path	The virtual path is the first part of the URL after the hostname of the server. In the screenshot above to the URL to open IMan will always be in the form:  <a href="http://&lt;hostname&gt;/IMan/.....">http://&lt;hostname&gt;/IMan/.....</a>
Application Pool	The application pool is a set of processes that run in IIS under the process. We recommend that IMan is allocated its own application pool.

6. Press Next to start the installation.



7. Press close to complete the installation.



8. After the installation is complete, there are several post installation steps required before IMan can be used. Please see the next section (Administering IMan) for details.



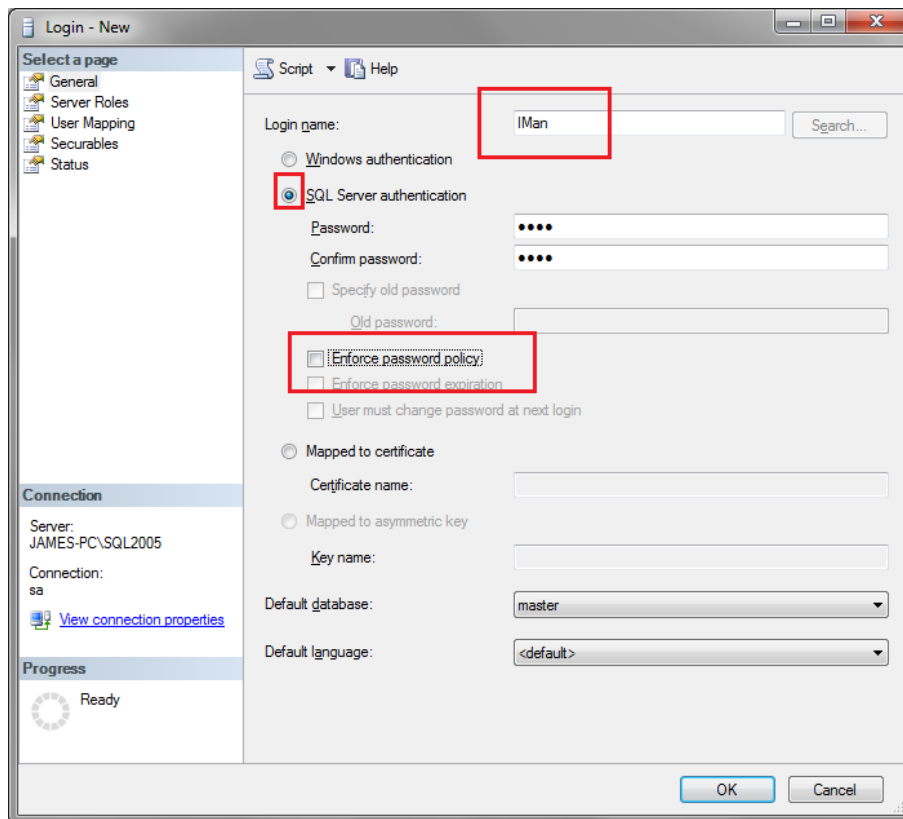
## IMAN DATABASE SETUP

In this section, you will create the IMan database.

1. If you do not have a SQL database, download and install SQL Express, otherwise use an existing SQL instance.

<http://www.microsoft.com/sqlserver/en/us/editions/express.aspx>

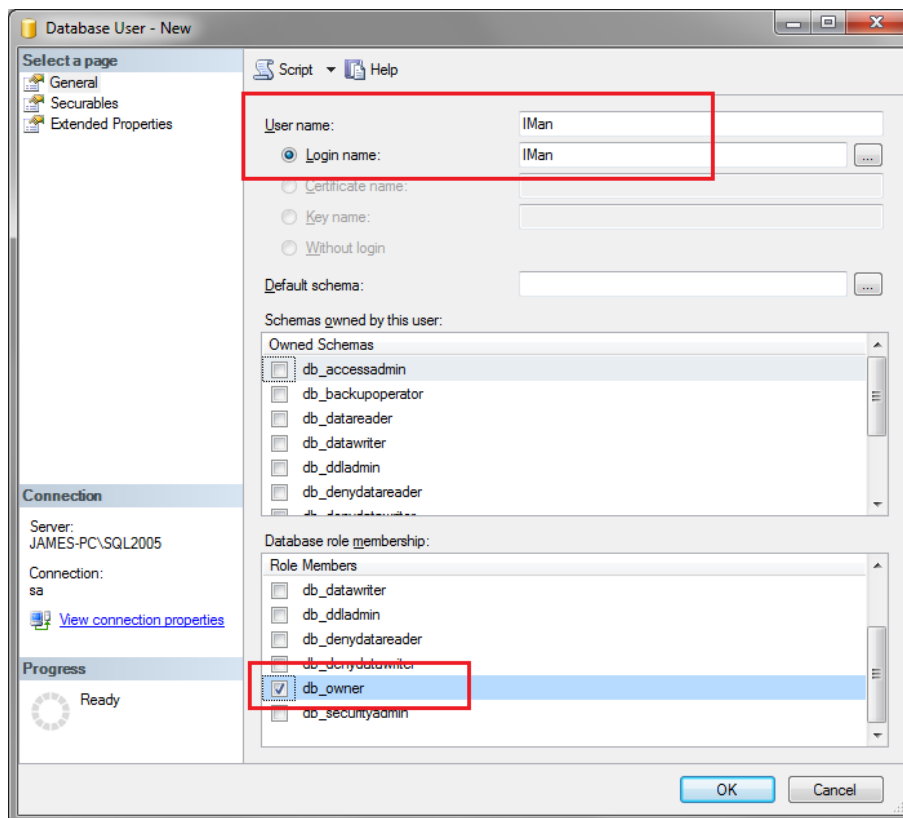
2. Create a database in SQL Server, call it IMan.
3. Create a SQL User, and ensure you disable 'Enforce password policy', then OK to close. IMan does support Windows authentication, but it is more difficult to setup.



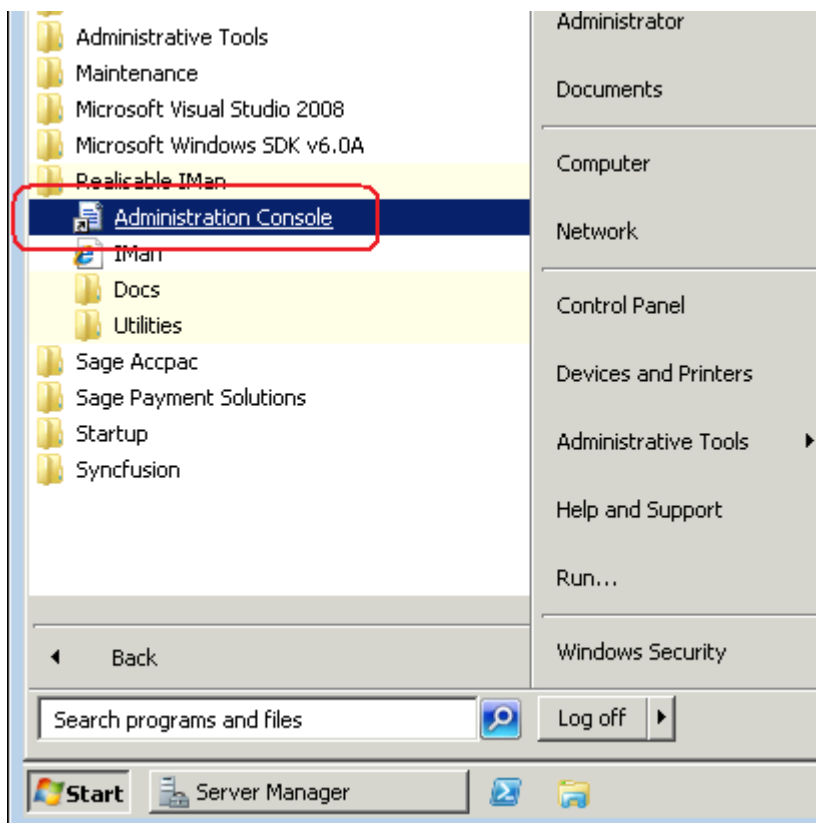
4. Assign the IMan user db\_owner rights to the IMan database.

Right click the Security against the IMan database and go New User.

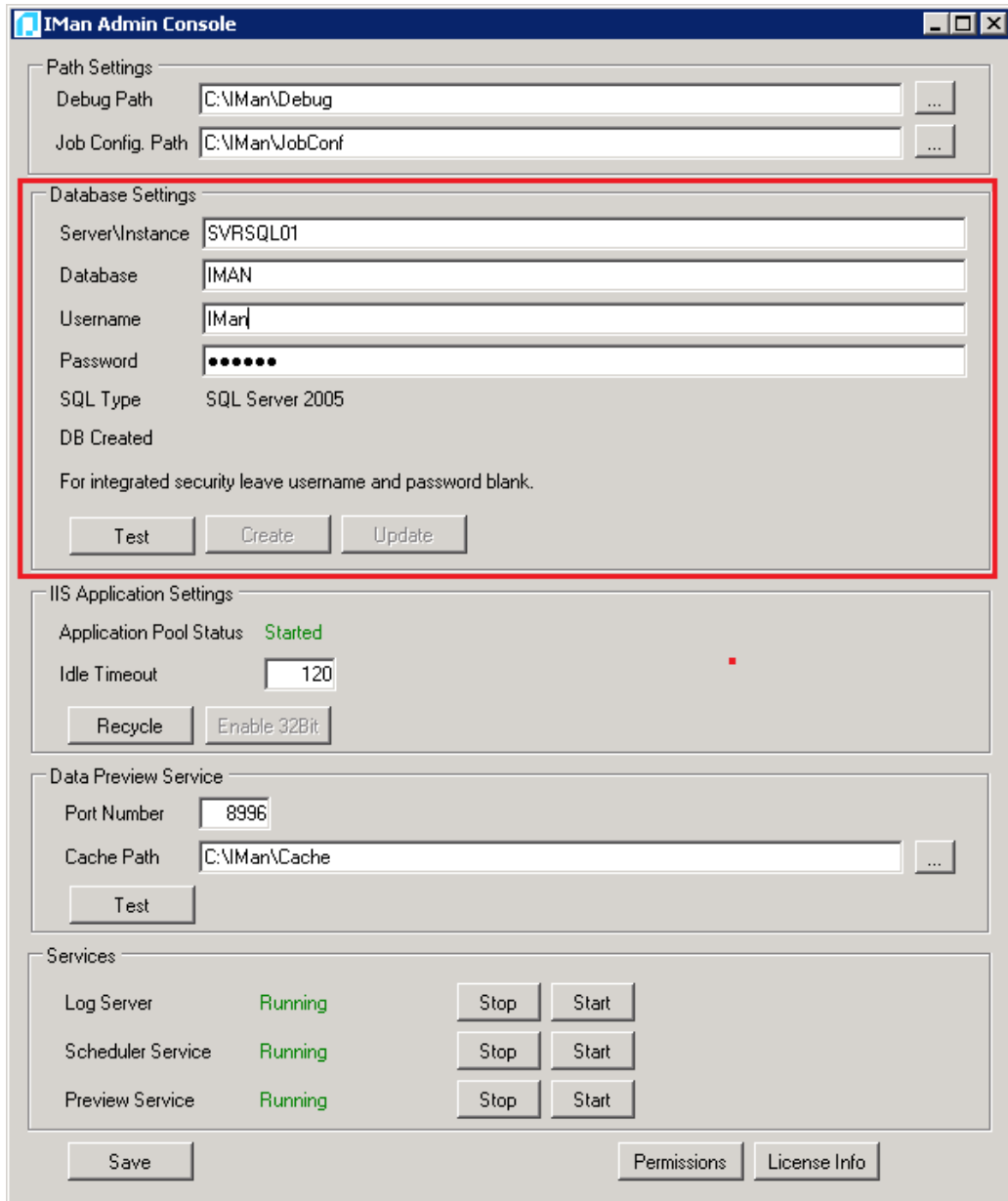
Assign the IMan SQL User db\_owner rights as shown below.



5. Open the IMan Administration Console by going Start, Programs, Realisable IMan, Administration Console.



- Set the Server/Instance, Database, Username and Password fields as per the database created in step 1 of this section.



The screenshot shows the IMan Admin Console window. The 'Database Settings' section is highlighted with a red border. It contains the following fields and controls:

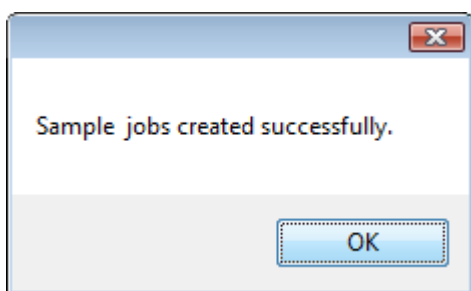
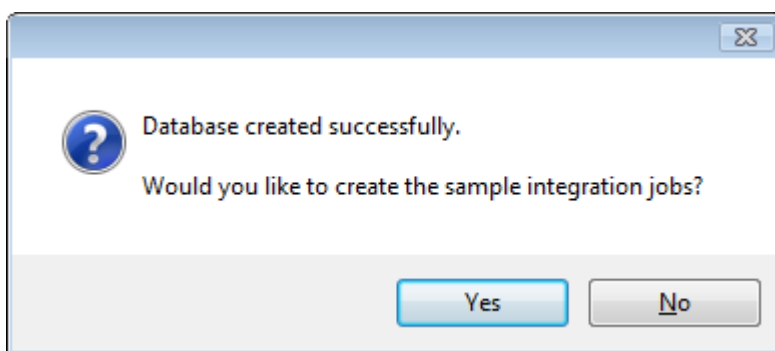
- Path Settings:
  - Debug Path: C:\IMan\Debug
  - Job Config. Path: C:\IMan\JobConf
- Database Settings (highlighted):
  - Server\Instance: SVRSQL01
  - Database: IMAN
  - Username: IMan
  - Password: (masked with dots)
  - SQL Type: SQL Server 2005
  - DB Created: (checkbox)
  - For integrated security leave username and password blank.
  - Buttons: Test, Create, Update
- IIS Application Settings:
  - Application Pool Status: Started
  - Idle Timeout: 120
  - Buttons: Recycle, Enable 32Bit
- Data Preview Service:
  - Port Number: 8996
  - Cache Path: C:\IMan\Cache
  - Button: Test
- Services:
  - Log Server: Running (Stop, Start buttons)
  - Scheduler Service: Running (Stop, Start buttons)
  - Preview Service: Running (Stop, Start buttons)
- Bottom Buttons: Save, Permissions, License Info

- Press the Test to check the credentials are correct. A message box will display the results from the connection attempt.



- If the previous step was successful, the Create will be enabled; press to create the database.

If the database was created successfully, press 'Yes' to create the sample jobs.



*THE FOLLOWING SECTIONS OF THIS GUIDE ASSSUME THE SAMPLE JOBS HAVE BEEN CREATED.*

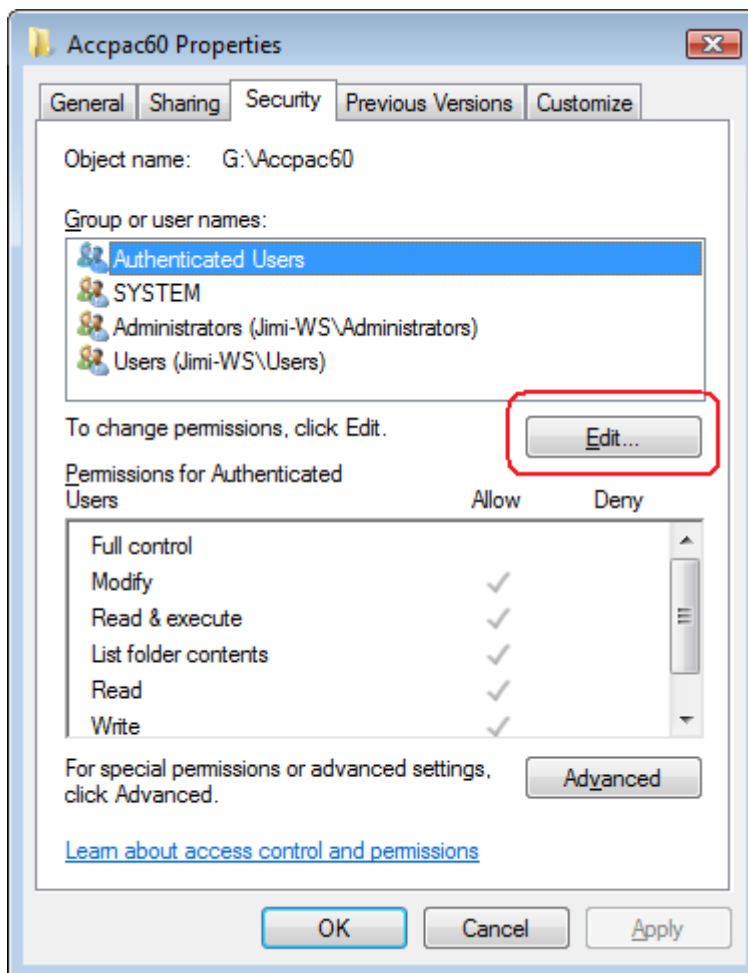
9. Press Save to save all the settings.

## SAGE300 (ACCPAC) - SETTING SECURITY

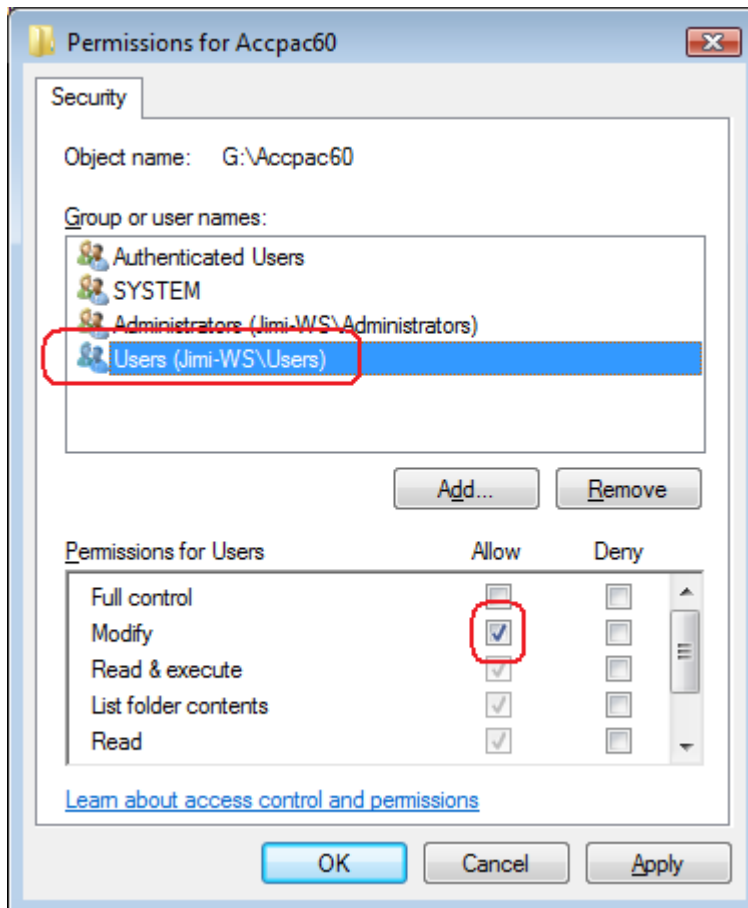
If integrating with Sage300 it is necessary to set the file security on the Sage300 shared data directory to allow the IMan programs access.

This section describes the 'easy' way to set security and assumes the Sage300 shared data is located on the same workstation or server as IMan. If not, a full discussion and how to setup security is found in the Sage300 IMan User Guide.

1. Open the properties for the Sage300 shared data directory, click the security tab, and press Edit.

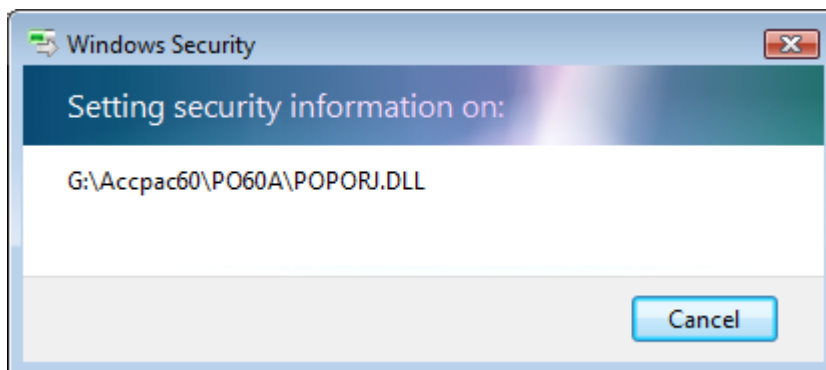


2. In the 'Group or User names' box, check for the Users group. If the 'Users' group is not already listed, click Add to add the 'Users' group.



After adding/selecting the group, ensure the group has the 'Modify' permission enabled.

3. Press Apply to apply the settings.



4. When complete close the Properties box.

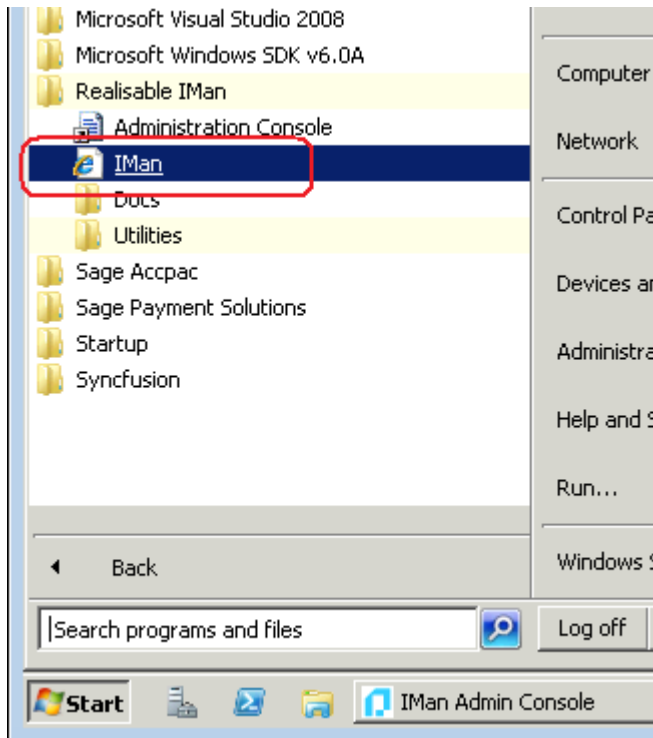
## SETTING UP IMAN

---

In this section you will setup the few areas needed to get IMan running.

### Log into IMan

1. Go Start, Programs, Realisable IMan, IMan.



IMan will take about 15-20 seconds the first it is opened.

2. The default credentials are:

User – Admin

Pass – imanAdmin

A screenshot of the IMan login interface. At the top, it says 'WELCOME TO REALISABLE IMAN' next to the Realisable logo. Below this is a login form with two input fields: 'Login' with the text 'james' and 'Password' with masked characters '□□□□'. A 'Log In' button is positioned below the password field.

### Setting Email Settings

Each time an integration is run, an email containing the results is sent via email. In order for the audit emails to be sent you require details of an SMTP server through which

outbound email can be relayed. Typically this is through either an internal mailserver such as Exchange or an external server provided by your internet service provider (ISP).

1. Click on Setup tab at the top of the screen.



2. On the left hand menu bar click Email Servers, then double-click the 'DEFAULT' row in the grid.

**VBScript & Lookups** ^

- Lookup Table Maint
- LookupTables
- Lookups
- Counters
- Common Functions

**Setup Items** ^

- Email Servers **1.**
- Email Groups
- FTP Servers
- System Connectors

### EMAIL SERVERS

	ID	Description	Username	Address	SSL	TLS
*						
▶	DEFAULT	Default Mail Server	<b>2. Double Click</b>		False	False

◀ ▶ ✎ ✕ ↺

3. Enter the details for an SMTP server and press the Green tick save the record.

### EMAIL SERVERS

ID:

Description:

Email Server:

Port:   
Leave as 0 to use the default port for the respective Email Server type.

User Name:

Password:

SSL: ☐

TLS: ☐

☒ ☐



- Click the Email Groups option on the left hand menu bar, and double click the 'EMAILGRP' row.

### EMAIL GROUPS

	ID	Description	Server ID	From	To Address	CC Address	BCC Address
*							
▶	EMAILGRP	Default Email Group	DEFAULT	Double Click			



- Enter an email address into the 'From' and 'To' address fields. When setting up IMan initially, set both these fields to your own email address.

### EMAIL GROUPS

Group ID:	<input type="text" value="DEFAULT"/>
Description:	<input type="text" value="Default Email Group"/>
Mail Server:	<input type="text" value="Realisable Software"/>
From Address:	<input type="text" value="johnsmith@realisable.co.uk"/>
Email To:	<input type="text" value="johnsmith@realisable.co.uk"/>
Email CC:	<input type="text"/>
Email BCC:	<input type="text"/>



## Setting System Connectors

System connectors define the various settings required to connect and authenticate to applications such as Sage300, Sage200, SageCRM, etc. It is necessary to review the default connectors, and make changes where necessary.

1. Click the System Connectors option of the left hand side of the Setup menu.

**VBScript & Lookups** ^

- Lookup Table Maint
- Lookup Tables
- Lookups
- Counters
- Common Functions

**Setup Items** ^

- Database Connections
- Email Servers
- Email Groups
- System Connectors** 1.
- Connector Licensing

### SYSTEM CONNECTORS

ID	Description	Connector Type	Username	Organisation ID
*				
▶ SAMS300INC	Sample Sage300 SAMINC Connector	Sage300 ERP (Accpac)	ADMIN	ASMI61
SAMS300LTD	Sample Sage300 SAMLTD Connector	Sage300 ERP (Accpac)	ADMIN	ASML61
SAMSAGE200	Sample Sage200 Connector	Sage 200	admin	Sage200DemoData
SAMSAGECRM	Sample SageCRM Connector	Sage CRM	admin	

2. Double Click

Navigation icons: << >> ✎ ✕ ↺

2. Double click on the relevant row in the grid to edit its settings.
3. For each of the connectors alter the Username, Password and Company Id fields as required.

### SYSTEM CONNECTORS

System ID:

System Type:  ▼

Description:

Connection String:

Username:

Password:

Confirm Password:

Company Id:

Navigation icons: ✓ ✕

For Sage300 connectors, the Company Id field is the Database ID as defined in the Database Setup.

For Sage200 connectors, the Company Id field is the Company Name (not the Database Id) as defined in the System Administration console.

For SageCRM connectors, alter the Connection String, username and password.

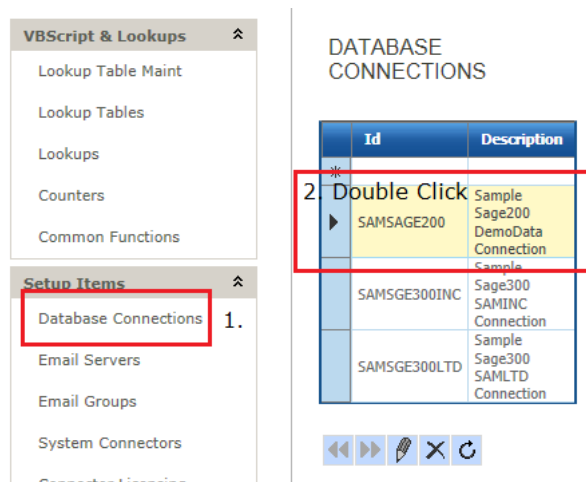
## Setting the Database Connections

In order to use some of the sample integrations it necessary to setup the database connection string so that IMan can connect to either the Sage200 or Sage300 databases.

Connection strings provide a means for applications to know how to connect to a particular database. A typical connection string will specify items such as the database server, database id, the username to connect to the database and its corresponding password. Connection strings can be daunting, but when broken down to their constituent parts are relatively easy to understand.

Please note, IMan integrates to all Sage applications via its API, these connection strings are used for data extraction and in two of the sample integrations to flag that a transaction has been exported/processed by toggling an optional field value.

1. Click the Database Connections option of the left hand side of the Setup menu.



2. Double click the relevant database connection.
3. Alter the connection string as required. IMan supports either ODBC/OleDb style connection strings (*Native Connection strings are not supported*).

### Sql2005 Sample Connection String

```
Provider=SQLNCLI;Server=myServerAddress;Database=myDataBase;Uid=myUsername;Pwd=myPassword;
```

### Sql2008 Sample Connection String

```
Provider=SQLNCLI10;Server=myServerAddress;Database=myDataBase;Uid=myUsername;Pwd=myPassword;
```

### Sql2012 Sample Connection String

```
Provider=SQLNCLI11;Server=myServerAddress;Database=myDataBase;Uid=myU  
sername; Pwd=myPassword;
```

In each of the samples replace:

- myServerAddress – With the server the relevant database is located.
- Database – The actual database id of the database.
- Username – The SQL user to connect to the database.
- Pwd – The password of the SQL User.

For Sage200, you may need to create a SQL user and assign it to the Sage200 database.

A good resource for connection strings is:

[www.connectionstrings.com](http://www.connectionstrings.com)

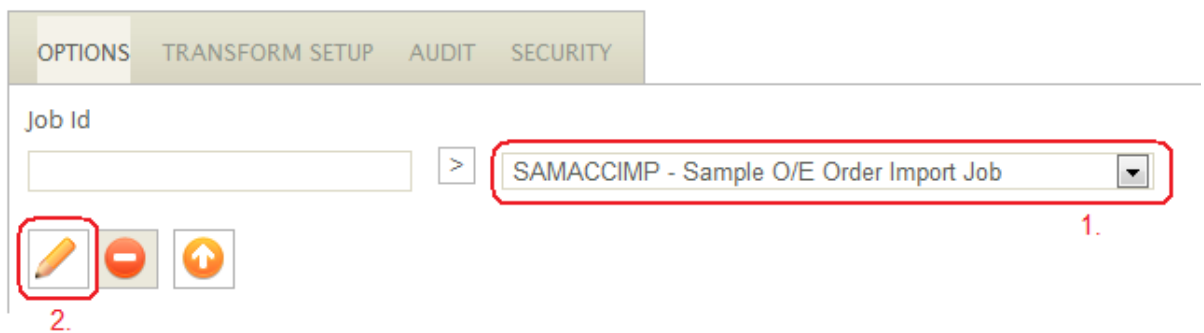
## USING THE INTEGRATION DESIGNER

There are several pre-designed integrations which are provided to help you understand to how create IMan integrations and highlight some functionality within the product.

This section describes in short each of the sample integrations and how to work with the designer. Please see the User Guide for more detailed description or use the training guides for a step by step detailed walkthrough.

### Opening the Designer

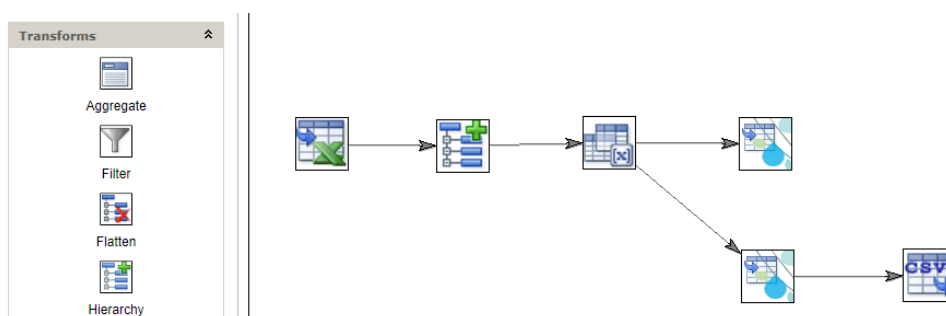
1. Click on the 'design' tab at the top of the screen.
2. From the 'Integration Id' drop down select one of the sample integrations, then press the Edit button to load the job.



3. Click on the Transform Setup tab to show the design the design palette.

On the left hand are the available 'transforms/readers/writers/tasks/connectors' which can be dragged onto the right hand palette.

The right hand side defines the sequence of nodes within the integration.



4. Under each of the nodes is a screen which defines various settings pertaining to the node or transforms type.

The first node in an integration is typically a reader or 'datasource'. Double click on the node to open its setup screen.

5. Each transform setup screen is divided into two sections:

- Left hand pane – Defines the transform settings of which there are three separate pages (File Layout/Options, Field Mapping, Audit), accessed by the tab options at the top of the screen.
- Right hand pane – The preview area allows you to view on the fly, the results of the transform.

Press the refresh button to generate the preview dataset on the right. (This may take about 10 seconds or so, on the first refresh).

Transform Id  
Read  
Transform Ids cannot contain the characters (&lt;~>).

Data Source  
File

+ File Options

Worksheet Id  
0  
The worksheet id to read the data from. Enter a number for the worksheet index or the id referring to the worksheet.

Header Rows  
1  
The number of header rows denoting field names.

+ Hierarchical Dataset Options

Refresh Save Close

Preview  
Orders: 7 Items | Generation Status: Partial Result Set

OrderId	CustomerId	CustomerName	CustomerAddress1	CustomerAddress2	CustomerAddress3
WEB0234	1400	Coastal Electric Company	586 Tanner Road		Toronto
WEB0234	1400	Coastal Electric Company	586 Tanner Road		Toronto
WEB0234	1400	Coastal Electric Company	586 Tanner Road		Toronto
WEB0237	1650	Realisable Software Ltd	91 Brick Lane		LONDON
WEB0237	1650	Realisable Software Ltd	91 Brick Lane		LONDON
WEB0238	7300	The Royal Cavendish Co.	8940 Mason Circle		LONDON
WEB0238	7300	The Royal Cavendish Co.	8940 Mason Circle		LONDON

6. To understand the actions of each transform open each node and press Refresh, moving to the Field Mapping tab.

## SAMPLE INTEGRATIONS WALKTHROUGH

---

This section walks through each of the steps in the sample integration jobs. These walkthroughs are a high-level guide to help you understand IMan and some of its capabilities.

These sample integrations are a first step to using and understanding IMan, following these you should use the relevant training guide for Sage300 or Sage200 which will guide you through the setup and configuration of several integrations.

Further resources can be found in the IMan User Guide and the application specific guides for Sage200, Sage300 and SageCRM.

The samples provided cover Sage300, Sage200 and SageCRM.

### SAMACCIMP (SAMPLE SAGE300 O/E ORDER IMPORT) & SAMS200IMP (SAMPLE SAGE200 SOP ORDER IMPORT)

---

The sales order integrations for both Sage200 and Sage300 both import data from an Excel spreadsheet, apply transformation to the data before creating/updating sales ledger customers and creating sales orders in the relevant application.

These integrations are a typical of an order import from a webcommerce site/application. In a real scenario the datasource would be Xml, however to simplify the integration we have substituted this for an Excel datasource.

For both Sage200 (UK) and Sage300, these integrations will create a corresponding payment. For Sage300 this will be a Prepayment (and A/R Receipt); for Sage200 it will be a Payment against Order.

#### 'Read' Transform




The read transform extracts data from the Excel spreadsheet.

#### 'Hierarchy' Transform

To import the data into Sage200/Sage300 it must be in a form or structure that corresponds to the type of data that is being imported. For Sales Orders, a header detail/structure is required.

The hierarchy transform allows you to take a dataset and apply a header/detail structure to it. The transform allows you to repeat this structure where there are a number of deeply nested children, but in this example a structure of 1 level deep is being created.

To view the hierarchical transform is applied press the Refresh button on the Field Mapping tab, and click '+' symbol to expand the record.

Orders: 3 Items     Generation Status: Partial Result Set					
	OrderId	CustomerId	CustomerName	CustomerAddress1	CustomerAddress2
	WEB0234	BIR004	Cute and Cosey Kitsch	23 Queen Victoria Rd	
OrderDetail: 3 Items					
	OrderId	OrderComments	LineNo	ItemNo	Description
	WEB0234	Please ship complete.	1	ACS/BLENDER	Professional Blender 5 S
	WEB0234	Please ship complete.	2	ACS/MEAT-SLICER	Commercial Meat Slicer
	WEB0234	Please ship complete.	3	ACS/TOASTER	Polished Steel 4 Slice To
	WEB0237	LON011	Realisable Software Ltd	91 Brick Lane	
	WEB0238	LON013	The Royal Cavendish Co.	8940 Mason Circle	

To view how the hierarchical transform is setup click Edit button (pencil).

Transaction Id to Hierarchise

Orders

Select the top most transaction id that is to be made into a hierarchy.

New Transaction Id

> Orders

Parent Id

Select the corresponding parent transaction id.



The selected fields are those that make-up the resulting 'Order'/header transaction.

The Key column is used to specify the field(s) used to define the relationship between each of the transaction types. In this example the 'Order Id' field uniquely identifies each 'Order'/header transaction.



	Input Fld. Name	Type	Import	New Name	Key
►	OrderId	Text	<input checked="" type="checkbox"/>	OrderId	1
	CustomerId	Text	<input checked="" type="checkbox"/>	CustomerId	0
	CustomerName	Text	<input checked="" type="checkbox"/>	CustomerName	0
	CustomerAddress1	Text	<input checked="" type="checkbox"/>	CustomerAddress1	0
	CustomerAddress2	Text	<input checked="" type="checkbox"/>	CustomerAddress2	0
	CustomerAddress3	Text	<input checked="" type="checkbox"/>	CustomerAddress3	0
	CustomerCountry	Text	<input checked="" type="checkbox"/>	CustomerCountry	0
	CustomerZip	Text	<input checked="" type="checkbox"/>	CustomerZip	0
	ShipAddress1	Text	<input checked="" type="checkbox"/>	ShipAddress1	0
	ShipAddress2	Text	<input checked="" type="checkbox"/>	ShipAddress2	0
	ShipAddress3	Text	<input checked="" type="checkbox"/>	ShipAddress3	0
	ShipCountry	Text	<input checked="" type="checkbox"/>	ShipCountry	0
	ShipZip	Text	<input checked="" type="checkbox"/>	ShipZip	0
	ShipDate	Date/Time	<input checked="" type="checkbox"/>	ShipDate	0
	OrderComments	Text	<input type="checkbox"/>	OrderComments	0



To return to the previous screen press either the green tick or the red cross.

To view the Order detail setup change the drop-down from Orders to OrderDetail and press Edit.

New Transaction Id

OrderDetail ▼

Parent Id

Orders ▼

Select the corresponding parent transaction id.

	Input Fld. Name	Type	Import	New Name	Key
▶	OrderTotal	Decimal	<input type="checkbox"/>	OrderTotal	0
	PaymentAmount	Decimal	<input type="checkbox"/>	PaymentAmount	0
	SalesPerson	Text	<input type="checkbox"/>	SalesPerson	0
	LineNo	Integer	<input checked="" type="checkbox"/>	LineNo	2
	ItemNo	Text	<input checked="" type="checkbox"/>	ItemNo	0
	Description	Text	<input checked="" type="checkbox"/>	Description	0
	Qty	Decimal	<input checked="" type="checkbox"/>	Qty	0
	Price	Decimal	<input checked="" type="checkbox"/>	Price	0
	ItemDisc	Decimal	<input checked="" type="checkbox"/>	ItemDisc	0



The first page of fields has only the 'Order Id' field is selected, with Key column set to 1. The second page has the detail fields selected, the 'LineNo' field uniquely identifies the detail line so it has the Key value set to 2.

### Map Transform

The Map transform allows you to apply expression/formulas to each field. The expression can be either a static value, or a formula allowing you to manipulate the value of an existing or new field.

Because of its flexibility, almost every job has a Map transform. We recommend that even if it isn't required, a Map transform is inserted into a job, as it will makes future changes and maintenance easier.

VBScript is used as the underlying language for formulae. IMan augments VBScript by allowing fields from the dataset to be referenced, prefixing the field name with a percentage '%' sign, as shown below.

In the example below the function uses the FuzzyCountryNameLookup function, normalising the CountryName field to make it consistent when it is imported. A full function reference is available in Appendix A of the in the User Guide.

Current Field Name

CustomerCountry

New Field Name

CustomerCountry

Field Type

Text

Log Key

0

Enable Script Evaluation

☒

Field Value/Script Formula

FuzzyCountryNameLookup(%CustomerCountry, "ENG", "NAME", True)

Check

☒ No Errors Found.

To view the results of Map transform press the Refresh button on the Field Mapping tab. On refresh there will be several new fields added to the dataset.

## Customer & Order Imports (Connector Transforms)

The connector transforms define how the dataset fields are mapped into the application.

Each of the connector transforms i.e. Sage300, Sage200, SageCRM, etc all use a common screen for setup.

The Options tab defines the 'System' we are importing into, the type of data that is being imported and how updates are made.

Transform Id

Transform Ids cannot contain the characters (&<>?"'!).

Priority

Select Sage Accpac System

Sage Accpac Import Type

Update Operation

The Field Mapping tab defines how the transactions and fields are mapped to the 'Import Type' selected on the Options tab.

To map the fields, press the Edit button, and select the field from the field dropdown.

Transaction Id

Orders

Sage Accpac Transaction Type

Customers

	Field Name	Type	Log Key	Field
	OrderId	Text	0	
▶	CustomerId	Text	0	Customer Number
	CustomerName	Text	0	Customer Name
	CustomerAddress1	Text	0	Group Code
	CustomerAddress2	Text	0	Short Name
	CustomerCity	Text	0	Status
	CustomerState	Text	0	On Hold
	CustomerCountry	Text	0	Start Date
	CustomerZip	Text	0	Territory Code
	ShipAddress1	Text	0	Address Line 1
	ShipAddress2	Text	0	Address Line 2
	ShipCity	Text	0	Address Line 3
	ShipState	Text	0	Address Line 4
	ShipCountry	Text	0	City
	ShipZip	Text	0	State/Prov.
				Zip/Postal Code
				Country
				Phone Number
				Fax Number
				E-mail
				Web Site
				Contact Name
				Contact's Phone
				Contact's Fax
				Contact's E-mail

For the 'Order' connector the Transaction Id dropdown is matched against the Transaction type of the Import Type. In the screenshot below the 'OrderDetails' transaction id is matched against the 'Order Details' transaction type of the Sales Order Import type.

Transaction Id

OrderDetails

Sage Accpac Transaction Type

Order Details

Transaction Id  
is matched to  
Transaction Type

Pressing Refresh will import the transactions into Accpac/Sage200. When the import is complete, scroll the preview area to the right and you will find the Order Number field populated with Sage200's/Sage300's auto-generated value.

## Preview

JNTRY	SHPZIP	EXPDATE	COMMENT	RECPAMOUNT	SALESPER1	ORDNUMBER	BA
	67183-1085	12/08/2010 00:00:00	Please ship complete.	377.9	SP	ORD000000000211	CCI
igdom	E1 6QL	21/08/2010 00:00:00		701.24	BB	ORD000000000212	CCI
igdom	NW1 9HA	15/08/2010 00:00:00	Deliver direct to office.	97.53	WEB	ORD000000000213	CCI

### Status Transform (CSV Write Transform)

IMan has facility to write/export to a number of data formats such as CSV/Text, Xml, Excel & ODBC/OleDb databases.

This transform exports the results of the order import to a CSV file. This type of operation is often required, so the 'source' application can be updated with a flag to indicating the transaction has been processed or an identifier such as an auto-generated Order number or entry reference.

Transform Id

Transform Ids cannot contain the characters (&lt;?\*"!).



File Options

Field Delimiter

Line Delimiter

Quote String Fields



Write Header Rows



Generate File Per Transaction

Leave blank to generate a file for the entire dataset.

Press Refresh to generate the file. The file will be created in the File Path field as specified in the File Options.



## File Options

### File Path

G:\Personal\Software\IntMan\Src\OutputData

The path for where the files are located.

### File Name

"OrderStatus" & Format(Date, "yyyymmdd") & ".csv"

The file name may be either a static value or contain wildcards (\*, ?).

## SAMACCEXTRCT (SAMPLE ITEM & PRICE EXTRACT) & SAMS200EXTRT (SAMPLE SAGE200 ITEM & PRICE EXTRACT)

---

These two integrations export Item and Pricing data from their respective databases to an Xml file. This style of integration is representative of a master data extract required for an e-commerce store.

To export data from Sage300/Sage200 a database reader will extract data directly from the database, since IMan doesn't support native data extraction from Sage300, Sage200, etc.

### Database Reader

The connection to the database must first be defined in order for the extractions to work.


*When first opening the database reader it may take 10 seconds to open as IMan is searching for the server in the pre-configured connection string.*

1. Load the job, by selecting it from the Job Id drop down, pressing the edit button and clicking on the 'Transform Setup' tab.
2. Open the 'reader' node, expand the SQL Database Options.

Transform Id

Read

Transform Ids cannot contain the characters (&<>?\*!").


 SQL Database Options

Database Connection/Connection String

Sample Sage300 SAMLTD Connection

SQL Statement

```
select I.ITEMNO, I.[DESC], I.FMTITEMNO, I.CATEGORY, I.COMMENT1  
as 'Short Desc', I.COMMENT2 as 'Long Desc', D.UNITPRICE as 'Base Price',  
I.STOCKUNIT  
from ICITEM I left outer join ICPRIC P on (I.ITEMNO = P.ITEMNO and  
P.PRICELIST = 'RTL' and P.CURRENCY = 'USD' and P.PRICEBY = 1) left outer  
join ICPRICP D on (P.ITEMNO = D.ITEMNO and P.PRICELIST = D.PRICELIST
```

 Heirarchical Dataset Options

3. The Database Connection/Connection String field should not require changing as this is already pointing to the shared Database Connection setup in the section preceding the sample integrations.



4. Click Refresh to check the connection, if successful the results will be shown in the Preview area otherwise an error is displayed. If an error is displayed, it is probably because the connection string is ill-configured, in which return to Database Connections within Setup and make the necessary changes.

### Map Transform

In both transforms the Map transform adds a new field for the run date & time.

See Order Import jobs for further detail.

### Write Transform (Xml Writer)

The write transform creates the export in Xml format. To specify the various nodes within the Xml file, each transaction and field is assigned a path and/or attribute.

Field Name  
Short Desc

Export Field  
☒

Is Relative XPath  
☒

XPath  
Attributes/ShortDescription

Attribute

Namespaces  
▼

Enclose Value in CDATA  
☒

Press Refresh to generate the file and double click to open the file in Internet Explorer. The annotated screenshot below illustrates how the various paths in the resultant file.

```

<?xml version="1.0" encoding="utf-8" ?>
- <Items ExportDateTime="20110811T14:16:58.000Z">
+ <Item>
+ <Item>
+ <Item>
- <Item>
  <ItemNo>A11030</ItemNo>
  <Description>Fluorescent Desk Lamp</Description>
  <FormattedItemNumber>A1-103/0</FormattedItemNumber>
  <UnitOfMeasure>Ea.</UnitOfMeasure>
- <Price>
  <Base>59.99</Base>
  </Price>
- <Attributes Category="A1">
  - <ShortDescription>
    <![CDATA[ Comes with stand and mounting bracket for wall or shelf. ]]>
    </ShortDescription>
  - <LongDescription>
    <![CDATA[ Uses a 13W mini fluorescent bulb (included). ]]>
    </LongDescription>
  </Attributes>
</Item>
- <Item>
  <ItemNo>A11050</ItemNo>
  <Description>13W Mini Fluorescent Bulb</Description>
  <FormattedItemNumber>A1-105/0</FormattedItemNumber>
  <UnitOfMeasure>Ea.</UnitOfMeasure>

```

Initial Path

Transaction Path

Fields

## SAMACCARINV (SAMPLE SAGE300 A/R INVOICE IMPORT)

---

This imports A/R Invoices from an Excel file into Sage300 as an A/R Invoice batch. The invoices created are a mixture of item and summary invoices, where the determination of the invoice type is based on the data in the file.

The integration follows a fairly standard flow of:

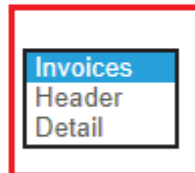
- Read – Pulls data from the Excel file.
- Hierarchy – Transforms flat dataset into a hierarchical one consisting of three levels: invoice batch (Invoices), invoice header (Header) & invoice details (Detail).

### Transaction Id to Hierarchise

Invoices ▾

Select the top most transaction id that is to be made into a hierarchy.

### New Transaction Id



### Parent Id

▾

Select the corresponding parent transaction id.

- Map – Adds some fields (Description, Batch Total, Batch Count & Batch Number) into the Invoice Batch (Invoices) to be able to specify and capture those details.

The InvoiceType field in the invoice header (Header) uses a formula to determine if an invoice is a summary or item invoice.

- Connector – Maps the data into Sage300.

## SAMACCARRCP (SAMPLE SAGE300 A/R RECEIPT IMPORT)

This imports A/R Receipts from a CSV file into Sage300 as an A/R Receipt batch. The receipts are a mixture of applied receipts and prepayments, where the determination of the receipt type is based on the data in the file.

The integration the flow of:

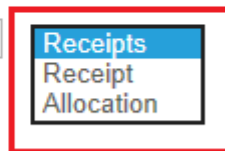
- Read – Pulls data from the CSV file.
- Hierarchy – Transforms flat dataset into a hierarchical one consisting of three levels: receipt batch (Receipts), receipt header (Receipt) & receipt details (Allocation).

### Transaction Id to Hierarchise

Receipts ▾

Select the top most transaction id that is to be made into a hierarchy.

### New Transaction Id



### Parent Id

Select the corresponding parent transaction id.

- Map – Adds some fields (Bank, Description, Batch Total, Batch Count & Batch Number) into the Receipt Batch (Receipts) to be able to specify and capture those details.

The ReceiptType field in the receipt header (Receipt) transaction uses a formula based on the value of Memo field to determine if a receipt is a prepayment or applied receipt.

The ReceiptAmount field in the receipt header (Receipt) calculates the total of the receipt using the Sum function. The total is then used to set the Receipt Total field in Sage300.

- Connector – Maps the data into Sage300.

## SAMACCPTIME (SAMPLE SAGE300 CANADIAN PAYROLL IMPORT)

This integration illustrates how to import a very simple timesheet from Excel into Canadian Payroll.

The Excel sheet has columns to record hours for each day of the week i.e. Monday, Tuesday, etc. The integration uses a combination of the Hierarchy, Map & Aggregate transforms so that each 'Day' column is transformed to its own 'Day' record so that it can be imported as a 'Timecard Detail'.

The integration the flow of:

### Read

Extracts the data from the spreadsheet.

### Hierarchy

Creates a two level hierarchy of Timecard header and the timecard details, where the details are still in a day per column format.

### Preview

Timecard: 1 Items   ( Showing 1 to 1 )   Generation Status: Partial Result Set									
	ID	Employee	PeriodEnd	imanpkey0					
▶	📅	1	300200	09/05/2013 00:00:00	0				
TimecardDetail: 1 Items									
	ID	Employee	PeriodEnd	Monday	Tuesday	Wednesday	Thursday	Friday	imanpkey1
	1	300200	09/05/2013 00:00:00	2	3	4	2.5	4	0

### Map

The Timecard field in the Timecard transaction uses an IMan Counter to generate an incrementing timecard id. Without this the Excel sheet would to specify a unique timecard for every transaction.

```
GetCounterSequence("S300CPTIME")
```

This function calls the S300CPTIME Counter defined in the setup area of the IMan (created as part of the samples). For each Timecard this function generates a new sequential id and as such the Counter function can be used to specify document numbers (as our this example), customer numbers (for Sage300) and any other unique id.

**VBScript & Lookups**  
 Lookup Table Maint  
 Lookup Tables  
 Lookups  
 Counters  
 Common Functions  
**Setup Items**  
 Database Connections  
 Email Servers  
 Email Groups  
 System Connectors  
 Connector Licensing

**COUNTER ADMIN**  
  
 Counter ID:   
 Description:   
 Counter Prefix:   
 Counter Suffix:   
 Evaluate Key: ☐  
 Formula:   
 Step:   
 Starting Number:   
 Total Counter Length:   
☐ ☐

The Timecard detail adds four fields to the detail transaction which will be used to translate the column per day values to a record per day.

#### Current Transaction Id

Select the transaction id to edit.

VPJQKEXV: 12 Items ( Showing 1 to 12 )				
	Current Name	New Name	Type	Evaluate
*				
	ID	ID	Text	False
	Employee	Employee	Text	False
	Monday	Monday	Decimal	False
	Tuesday	Tuesday	Decimal	False
	Wednesday	Wednesday	Decimal	False
	Thursday	Thursday	Decimal	False
	Friday	Friday	Decimal	False
▶	PeriodEnd	PeriodEnd	Text	False
		Hours	Decimal	False
		EarningType	Text	False
		Rate	Decimal	False
		Date	Text	False

The EarningType and Rate fields have the static values of SALARY and 10 assigned them respectively.

#### Aggregate Transform

The aggregate transform is where the columns are translated to their own records so they can be imported into Sage300 as shown with the Hours & Date field below.

## Preview

Timecard: 1 Items   ( Showing 1 to 1 )   Generation Status: Complete											
ID	Employee	PeriodEnd					Timecard		imanpkey0		
1	300200	09/05/2013 00:00:00					C00002		0		
TimecardDetail: 5 Items											
ID	Employee	Monday	Tuesday	Wednesday	Thursday	Friday	Hours	EarningType	Rate	PeriodEnd	Date
1	300200	2	2	4	2.5	4	2	SALARY	10	09/05/2013 00:00:00	05/05/2013
1	300200	2	3	4	2.5	4	3	SALARY	10	09/05/2013 00:00:00	06/05/2013
1	300200	2	3	4	2.5	4	4	SALARY	10	09/05/2013 00:00:00	07/05/2013
1	300200	2	3	4	2.5	4	2.5	SALARY	10	09/05/2013 00:00:00	08/05/2013
1	300200	2	3	4	2.5	4	4	SALARY	10	09/05/2013 00:00:00	09/05/2013

OPTIONS	<b>FIELD MAPPING</b>	1. AUDIT
---------	----------------------	----------

Current Transaction Id  
TimecardDetail ▼  
Select the transaction id to edit.

2.

Delete Child Records  
☒
  
When enabled will delete the all the children records to be replaced with the calc records.

Calc Records  
Monday  
Tuesday  
Wednesday  
Thursday  
Friday

3.

Click on the Field Mapping tab & change the Current Transaction Id to TimecardDetail. Attached to the TimeCardDetail are five 'Calc Records', one for each day.

Select Monday and press the Edit button directly below.

To set the Hour field of this record to Monday's value the Hours field simply references the Monday column.

Field Name  
Hours

FieldType  
Decimal

Allow Value Accumulation  
☐

Enable Script Evaluation  
☒

Field Value/Script Formula  
%Monday

Check No Errors Found.

Detailed description: This is a configuration window for a field named 'Hours'. The field type is set to 'Decimal'. The 'Allow Value Accumulation' checkbox is unchecked, while 'Enable Script Evaluation' is checked. The 'Field Value/Script Formula' text area contains the formula '%Monday', which is highlighted with a red rectangular box. At the bottom, there is a 'Check' button with a circular arrow icon, followed by a green checkmark icon and the text 'No Errors Found.'

A similar formula is used to calculate the Date of the transaction. The Date of the transaction is simply the %PeriodDate minus four days.

Field Name  
Date

FieldType  
Text

Allow Value Accumulation  
☐

Enable Script Evaluation  
☒

Field Value/Script Formula  
CDate(%PeriodEnd) - 4

Check No Errors Found.

Detailed description: This is a configuration window for a field named 'Date'. The field type is set to 'Text'. The 'Allow Value Accumulation' checkbox is unchecked, while 'Enable Script Evaluation' is checked. The 'Field Value/Script Formula' text area contains the formula 'CDate(%PeriodEnd) - 4'. At the bottom, there is a 'Check' button with a circular arrow icon, followed by a green checkmark icon and the text 'No Errors Found.'

Press the Cancel button to return back to the Field Mapping tab.

Select Tuesday from the 'Calc Records' dropdown and press Edit.

The Hours field now references Tuesday and the Date field, instead of subtracting 4 from the PeriodEnd field subtracts 3.

Repeat for Wednesday, Thursday & Friday.

### Sage300 Connector

Maps the data into Sage300.



## SAMACCPJCTIME (SAMPLE SAGE300 PJC TIMECARD IMPORT)

---

This imports PJC Timesheets an Excel file into Sage300 as PJC Timesheets.

The integration the flow of:

- Read – Pulls data from the Excel file.
- Hierarchy – Transforms flat dataset into a hierarchical one consisting of the timecard header and the timecard totals.

Map – Adds a single field, Timecard so as to capture the Sage300 Timecard number.

- Connector – Maps the data into Sage300.

## SAMACCINTCO (SAMPLE SAGE300 INTERCOMPANY PROCESSING AUTOMATION)

---

Two integrations SAMACCINTEXP & SAMINTCOIMP demonstrate how IMan can be used to automate intercompany purchase and sales order processing. The

### Database Connection Setup

To extract intercompany purchase orders from the SAMLTD company and to extract the shipments from the SAMINC database the two integration query these Sage300 database directly.

The databases are also updated directly using the Sage300 database writers to toggle the export optional field (see below) to the transactions processed status.

**IMPORTANT!** Updating the Sage300 database directly is only permissible in certain scenarios where there no substantive impact on data integrity.

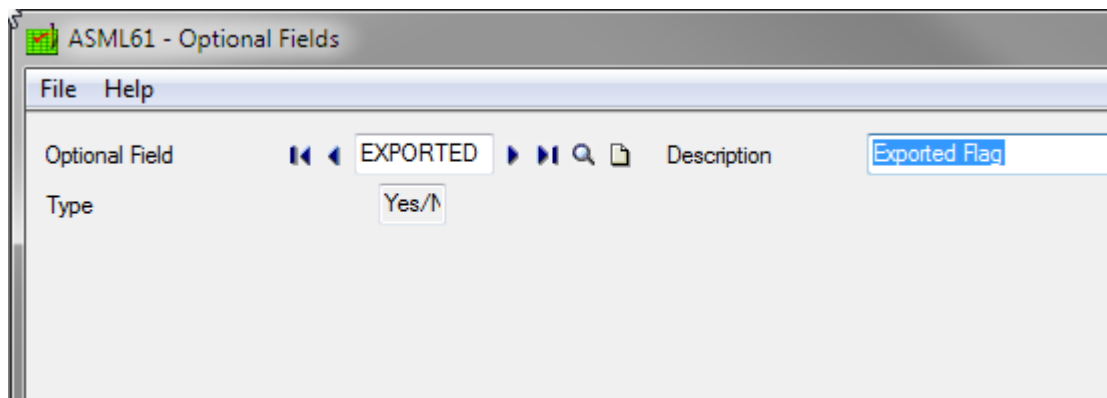
To ensure the Read & Write operations succeed the Database Connections for both SAMINC and SAMLTD must have been setup (see page 32).

### Sage300 Setup

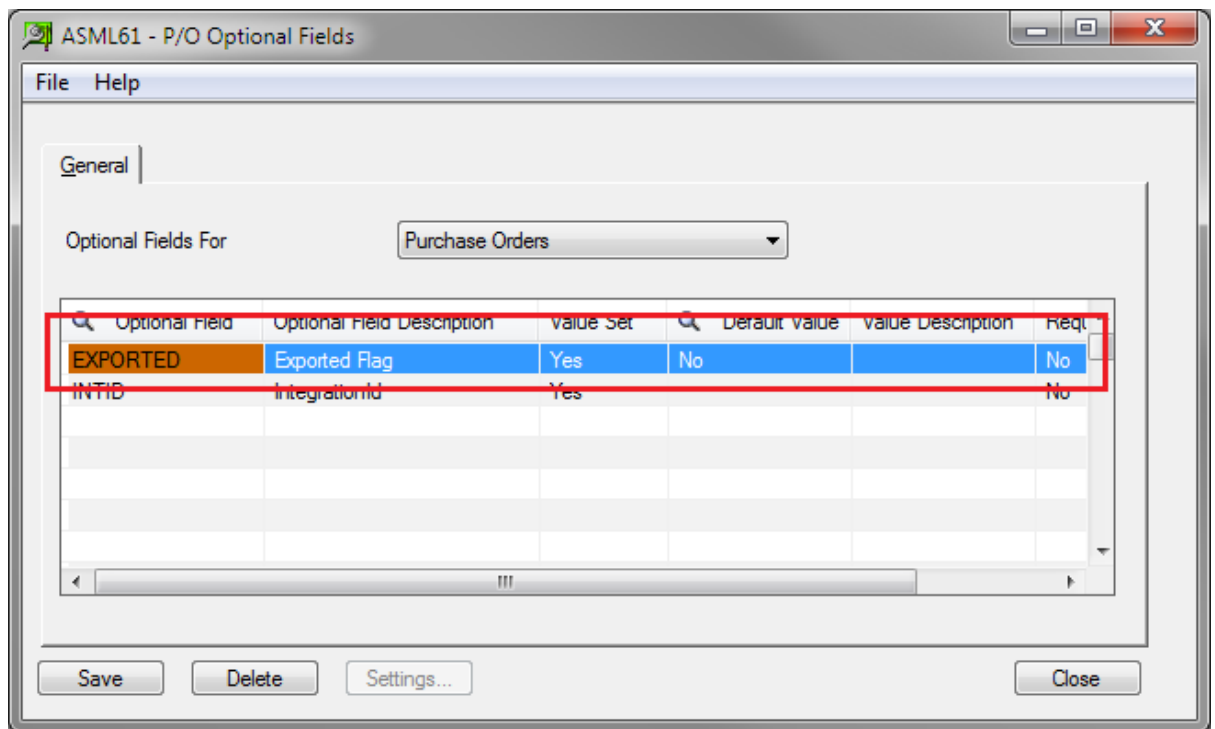
In order to use these two integrations you must have both SAMINC & SAMLTD setup on your Sage300 installation.

#### SAMLTD Setup

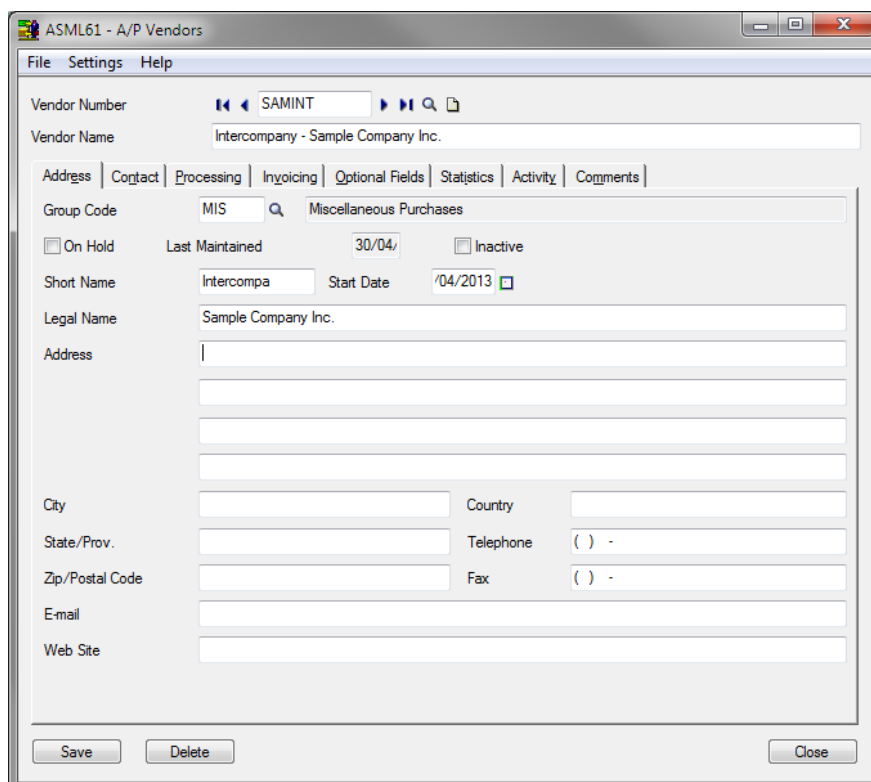
1. Create a Yes/No Optional field titled EXPORTED.



2. Assign the optional field to the P/O Purchase Orders as per the following:
  - Value Set – Yes
  - Default Value – No
  - Required – No
  - Auto Insert – Yes

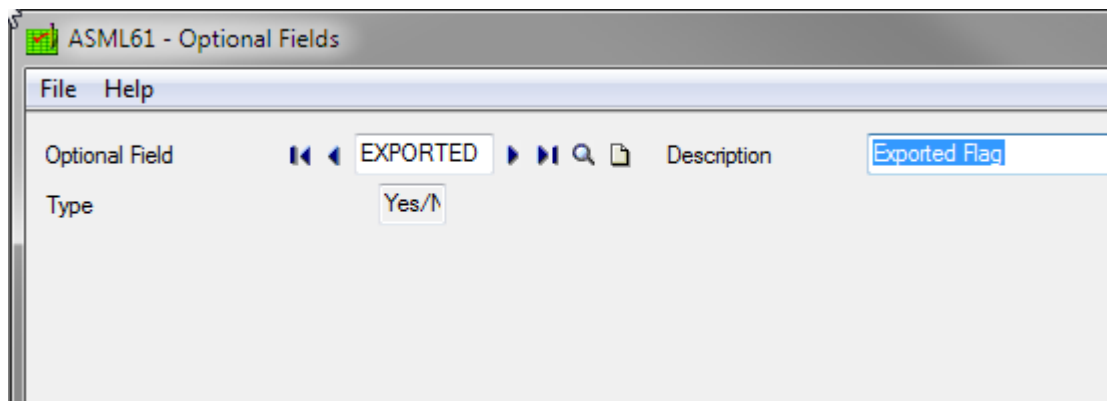


3. Create an A/P Vendor with the ID 'SAMINT' as shown below. This vendor will be used for any intercompany purchases. The query used to extract purchase orders will filter based on this Vendor.



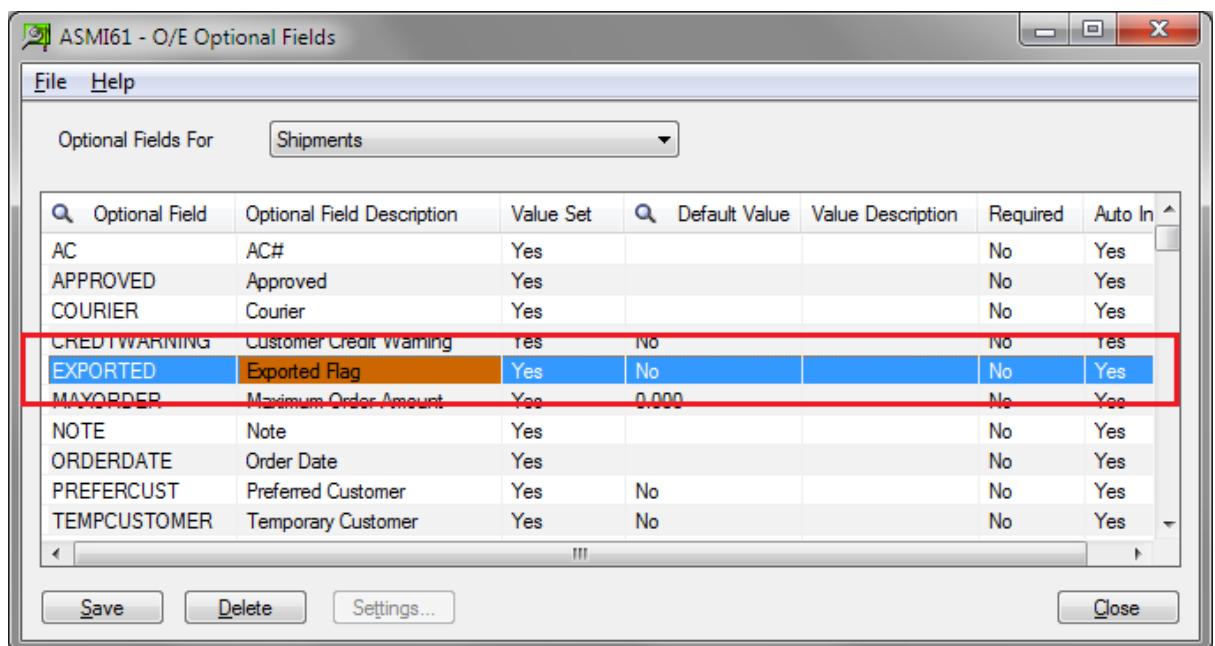
## SAMINC Setup

1. Create a Yes/No Optional field titled EXPORTED.

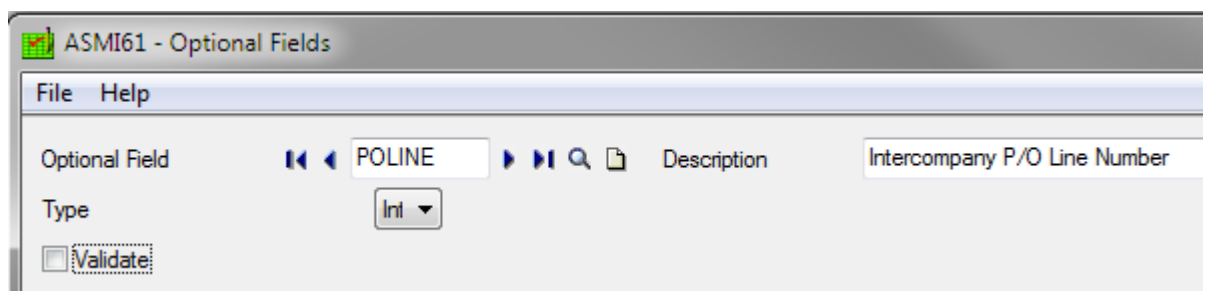


2. Assign the optional field to the O/E Shipments as per the following:

- Value Set – Yes
- Default Value – No
- Required – No
- Auto Insert – Yes

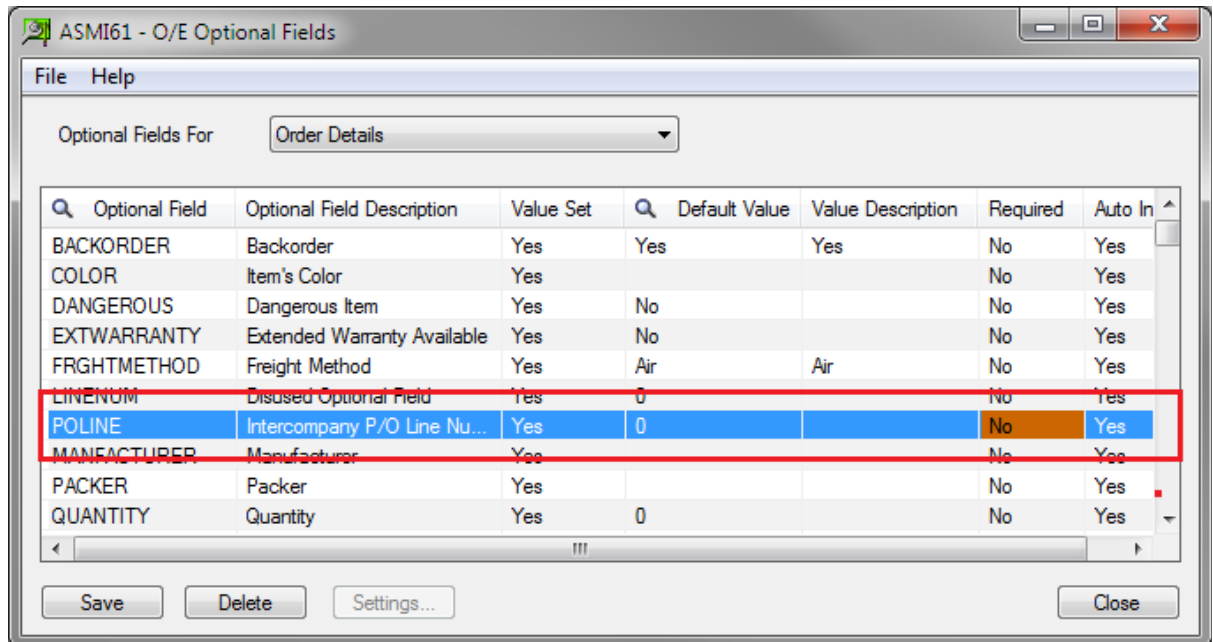


3. Create an integer optional field titled POLINE.



4. Assign the POLINE optional field to O/E Order Details as per the following:

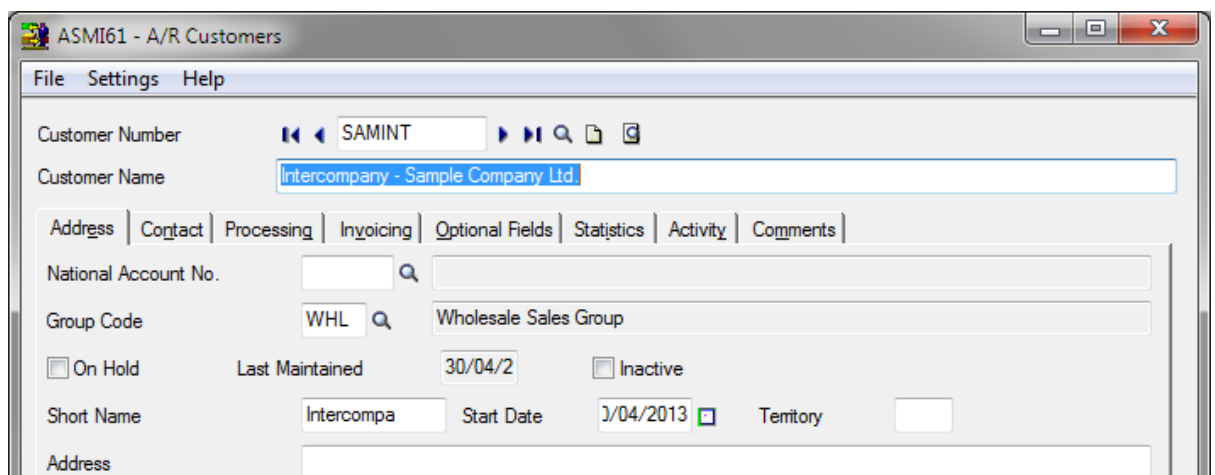
- Value Set – Yes
- Default Value – 0
- Required – No
- Auto Insert – Yes



5. Assign the same optional field to O/E Shipment Details as per the following:

- Value Set – Yes
- Default Value – 0
- Required – No
- Auto Insert – Yes

6. Create an A/R Customer in SAMINC with the ID SAMINT. Similarly to the intercompany vendor created in SAMLTD, this customer will be used for intercompany sales.



## SAMACCINTEXP - SAMLTD P/O to SAMINC O/E

This integration extracts the purchase orders from SAMLTD to create a corresponding O/E Order in SAMINC. The final step flags the exported orders to prevent them from being re-exported.



## SAMLTD - Create an Intercompany Purchase Order

Create an intercompany purchase order using the intercompany vendor (SAMINT) in SAMLTD.

Completed	Item Number	Item Description	Location	Drop-Ship	Quantity Ordered	Unit of Measure
No	A1-103/0	Item Num.	1	No	2	Ea.
No	A1-401/0	Desk Calendar Pad	1	No	4	Ea.

## Read Transform

This extracts purchase orders from SAMLTD using a SQL Query. The query extracts only the very basic fields required for the integration: Purchase Order Sequence Number, Purchase Order Number, Purchase Order Date, Purchase Order Line Sequence (below), Item Number & Quantity Ordered.

The Purchase Order Line Sequence is used to record which line an item belongs to. It will be stored with the line on the O/E Order and will be used in the return transaction to uniquely identify the item to receipt.

Orders: 2 Items   ( Showing 1 to 2 )   Generation Status: Partial Result Set						
	PORHSEQ	PONUMBER	DATE	PORLSEQ	ITEMNO	OQORDERED
▶	47282	PO000000483	20130507	47221	A1-103/0	2
	47282	PO000000483	20130507	47223	A1-401/0	4

The where clause in the query limits the result set to only intercompany purchase orders where the EXPORTED optional field is False.

#### SQL Statement

```
select H.PORHSEQ, H.PONUMBER, H.DATE, L.PORLSEQ, L.ITEMNO,
L.OQORDERED
from POPORH1 H inner join POPORL L on H.PORHSEQ = L.PORHSEQ
inner join POPORHO O on H.PORHSEQ = O.PORHSEQ and O.OPTFIELD
= 'EXPORTED'
where LTRIM(O.VALUE) = '0' and H.VDCODE = 'SAMINT'
```

#### Hierarchy Transform

Transforms the flat dataset to a hierarchical dataset consisting of a header and details.

Orders: 1 Items   ( Showing 1 to 1 )   Generation Status: Complete					
	PORHSEQ	PONUMBER	DATE	imanpkey0	
▶	☰	47282	PO000000483	20130507	0
OrderDetails: 2 Items					
	PONUMBER	PORLSEQ	ITEMNO	OQORDERED	imanpkey1
	PO000000483	47221	A1-103/0	2	0
	PO000000483	47223	A1-401/0	4	1

#### Map Transform

The Map transforms adds two fields to the dataset and modifies the value of another so the order can be imported into SAMINC.

Orders: 1 Items   ( Showing 1 to 1 )   Generation Status: Complete						
	PONUMBER	DATE	CUSTOMER	ORDNUMBER	PORHSEQ	imanpkey0
▶	PO000000483	07/05/2013	SAMINT		47282	0
OrderDetails: 2 Items						
	PONUMBER	PORLSEQ	ITEMNO	OQORDERED	imanpkey1	
	PO000000483	47221	A1-103/0	2	0	
	PO000000483	47223	A1-401/0	4	1	

ORDNUMBER – An empty field to capture the generated order number used on the audit report.

CUSTOMER – Set to a static value 'SAMINT'.

DATE – Uses the NumberToDate function to convert the numeric Date value (yyyymmdd format) to a date format which can be imported into Sage300 (dd/mm/yyyy or mm/dd/yyyy depending on your region).

### Sage300 Connector

The Sage300 Connector defines the mapping of the data into the SAMINC O/E Order.

Transform Id

Transform Ids cannot contain the characters (&<>?\*!).

Select Sage300 ERP (Accpac) System

▼

Sage300 ERP (Accpac) Import Type

▼

Update Operation

▼

The Field Mapping we map each of the fields in our dataset.



OPTIONS	FIELD MAPPING	AUDIT
---------	---------------	-------

Transaction Id

Orders ▼

Sage300 ERP (Accpac) Transaction Type

Orders ▼

	Field Name	Type	Log Key	Sage300 ERP (Accpac) Field
▶	PONUMBER	Text	1	Purchase Order Number
	DATE	Decimal	0	Order Date
	CUSTOMER	Text	0	Customer Number
	ORDNUMBER	Text	0	Order Number
	PORHSEQ	Decimal	20	

The Order Detail is mapped accordingly where the PO Line Sequence is mapped to the POLINE optional field. As stated previously this value will be used in the return transaction to identify the line in the purchase order to receipt.

Transaction Id

OrderDetails ▼

Sage300 ERP (Accpac) Transaction Type

Order Details ▼

	Field Name	Type	Log Key	Sage300 ERP (Accpac) Field
	PONUMBER	Text	1	
	PORLSEQ	Decimal	0	Intercompany P/O Line Number
	ITEMNO	Text	2	Item
▶	OQORDERED	Decimal	0	Quantity Ordered

The resulting preview will look something like.

OE0520: 1 Items   ( Showing 1 to 1 )   Generation Status: Complete						
	PONUMBER	ORDDATE	CUSTOMER	ORDNUMBER	PORHSEQ	imanpkey0
▶	PO000000483	07/05/2013	SAMINT	ORD000000000080	47282	0

OE0500: 2 Items					
	PONUMBER	POLINE	ITEM	QTYORDERED	imanpkey1
	PO000000483	47221	A1-103/0	2	0
	PO000000483	47223	A1-401/0	4	1

The O/E Order in SAMINC.

ASMI61 - O/E Order Entry

File Settings Help

Order No. ORD000000000080 No. of Shipments 0 Last Shipment No.

Customer No. SAMINT Intercompany - Sample Company Ltd. Last Invoice No.

Order Customer Taxes Optional Fields Sales Split Totals

Template Code ACTIV PO No. PO000000483 Status: Posted Entered By: ADMIN

Order Date 5/2013 On Hold

Order Type Active From Multiple Quotes Job Related Project Invoicing Retainage

Ship-To Location Location 1 Central warehouse - Seattle

Deliver By 5/2013 Exp. Ship Date 7/05/2013 Calc. Tax

Ship Via CCT Cross-Country Trucking Lines Tracking No.

Description Reference

Lin...	Type	Item No./ Misc. Charge	Kit/BOM	Description	Price List	Location	Deliv
1	Item	A1-103/0		Fluorescent Des...	USA	1	07/0
2	Item	A1-401/0		Desk Calendar ...	USA	1	07/0

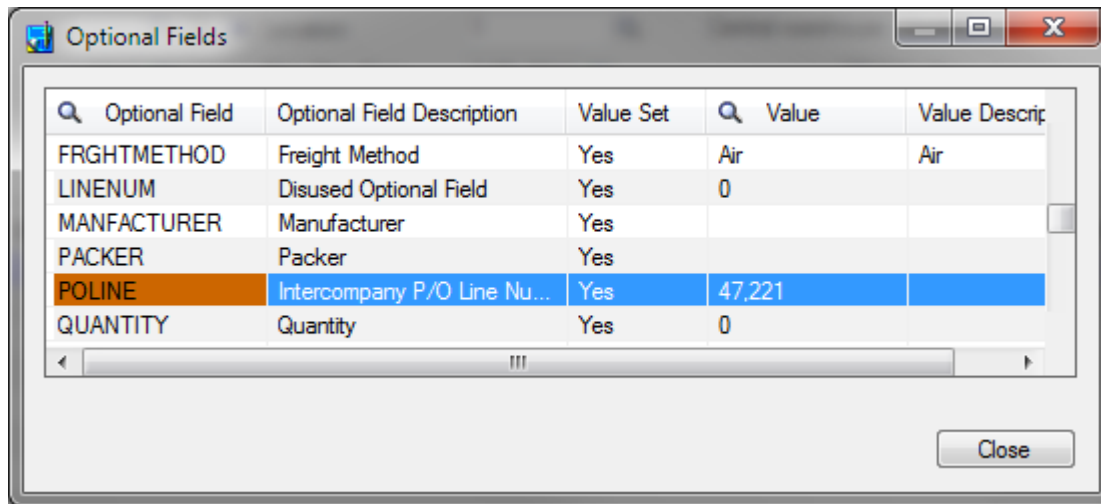
	Qty. on Hand	Qty. on Sales Order	Qty. on Purchase Order	Qty. Committed	Qty. Available
Location 1 (Ea.)	116	40	310	0	116
All Locations (Ea.)	636	71	486	0	636

Item/Tax... Components... Ship All

Order Subtotal 183.94

Post Create PO Delete History... Prepayment... Close

The POLINE optional field.



Optional Field	Optional Field Description	Value Set	Value	Value Description
FRGHTMETHOD	Freight Method	Yes	Air	Air
LINENUM	Disused Optional Field	Yes	0	
MANUFACTURER	Manufacturer	Yes		
PACKER	Packer	Yes		
<b>POLINE</b>	<b>Intercompany P/O Line Nu...</b>	<b>Yes</b>	<b>47,221</b>	
QUANTITY	Quantity	Yes	0	

### Filter Transform

The next three steps flag the SAMLTD Purchase Order to indicate it has been processed and prevent it from being exported again.

This Filter step is optional and would depend on the logic desired, but in this example we want to prevent orders which cannot be imported (for whatever reason) from flagging the EXPORTED. I.e. The order will continue to be exported from SAMLTD until it can be imported into SAMINC or until the EXPORTED optional field on the Purchase Order is manually set to True.

#### Current Transaction Id

OE0520 ▼

Select the transaction id to edit.

#### Field Value/Script Formula

%ORDNUMBER <> ""

 Check  No Errors Found.

### 2<sup>nd</sup> Map Transform

In this transform a field (VALUE) is added to the dataset which will enable the EXPORTED optional field to be set to True.

Current Transaction Id

OE0520

Select the transaction id to edit.

UBGIUYVP: 6 Items ( Showing 1 to 6 )				
	Current Name	New Name	Type	Evaluate
*				
▶	PONUMBER	PONUMBER	Text	False
	ORDDATE	ORDDATE	Decimal	False
	CUSTOMER	CUSTOMER	Text	False
	ORDNUMBER	ORDNUMBER	Text	False
		VALUE	Text	False
	PORHSEQ	PORHSEQ	Decimal	False

## Preview

OE0520: 1 Items ( Showing 1 to 1 )   Generation Status: Complete					
	PONUMBER	ORDDATE	CUSTOMER	ORDNUMBER	VALUE
▶	PO000000483	07/05/2013	SAMINT	ORD0000000000080	1

1

## DB Writer

This step performs the update to the EXPORTED optional field.

The Options tab specifies the database to update and the type of SQL Operation to perform. Note, the operation is 'Update' i.e. the existing EXPORTED optional field will be set from False to True.

Transform Id

UpdateSage300

Transform Ids cannot contain the characters (&amp;&lt;&gt;?\*"!).



## SQL Database Options

Database Connection/Connection String

Sample Sage300 SAMLTD Connection

SQL Operation

Update

Commit Transaction

(none)

Set to (none) to commit the entire dataset as a single transaction.

The Field Mapping tab is where the Update operation is setup.

Map To Table – Updating the Purchase Order Header Optional Fields Table.

Where Clause – This property specifies the Where clause used in the SQL Update command. Here we limit the records being updated to a single record by specifying the OPTFIELD field and then parameterising clause with the Purchase Order Header Sequence field (PORHSEQ), forming the primary key on the POPORHO table.

Current Transaction Id

OE0520 ▼

Select the transaction id to edit.

Map To Table

POPORHO

The table to insert/update records on.

Where Clause

OPTFIELD = 'EXPORTED' and PORHSEQ = %PORHSEQ

The 'where' clause to specify the records to update.

	Field Name	Export	SQL Command
▶	PONUMBER	False	
	ORDDATE	False	
	CUSTOMER	False	
	ORDNUMBER	False	
	VALUE	True	
	PORHSEQ	False	

After the update is run, the EXPORTED optional field on the Purchase Order should not be set to True.

ASML61 - P/O Purchase Order Entry

File Settings Help

PO Number PO000000483

Vendor Number SAMINT Intercompany - Sample Company Inc.

Last Receipt No.: Source: Er

Order Taxes Optional Fields Totals

Optional Field	Optional Field Description	Value Set	Value	Value Description
EXPORTED	Exported Flag	Yes	Yes	
INTID	IntegrationId	Yes		

## SAMACCINTIMP - SAMINC O/E Shipment to SAMLTD P/O Receipt

This second integration performs the return transaction by creating a P/O Receipt in SAMLTD whenever a shipment is raised against an intercompany order within SAMINC. The sequence for the integration is identical the one previous where the instead of querying SAMLTD for Intercompany Orders, the query is performed against SAMINC to intercompany shipments, and where Orders are created in SAMINC, instead Receipts are created in SAMLTD.

## Create Shipment for Intercompany Order in SAMINC

Create a shipment in SAMINC for the intercompany order created previously. To test that the quantities are correctly updated in the receipt, don't ship all, set them to be less than originally ordered.

## Read Transform

The Read transform extracts the intercompany shipments.

Shipments: 2 Items   ( Showing 1 to 2 )   Generation Status: Complete							
	SHIUNIQ	ORDNUMBER	SHINUMBER	PONUMBER	ITEM	QTYSHIPPED	POLINENUM
▶	3265	ORD00000000000080	SH0000000000000000066	PO000000483	A1-103/0	1	47221
	3265	ORD00000000000080	SH0000000000000000066	PO000000483	A1-401/0	3	47223

## Hierarchy Transform

Transforms a flat dataset into a hierarchical one.

Shipments: 1 Items   ( Showing 1 to 1 )   Generation Status: Partial Result Set						
	SHIUNIQ	ORDNUMBER	SHINUMBER	PONUMBER	imanpkey0	
▶	☰	3265	ORD0000000000080	SH0000000000000000066	PO0000000483	0
ShipLines: 2 Items						
	SHIUNIQ	PONUMBER	ITEM	QTYSHIPPED	POLINENUM	imanpkey1
	3265	PO0000000483	A1-103/0	1	47221	0
	3265	PO0000000483	A1-401/0	3	47223	1

## Map Transform

Add a single empty field to the dataset to allow the auto-generated receipt number to be captured and then reported on the Audit Report.

Current Transaction Id

Shipments ▼

Select the transaction id to edit.

OYTXQDV: 5 Items ( Showing 1 to 5 )				
	Current Name	New Name	Type	Evaluate
*				
▶	SHIUNIQ	SHIUNIQ	Decimal	False
	ORDNUMBER	ORDNUMBER	Text	False
	SHINUMBER	SHINUMBER	Text	False
	PONUMBER	PONUMBER	Text	False
		RECEIPTNBR	Text	False

Preview

Shipments: 1 Items   ( Showing 1 to 1 )   Generation Status: Complete					
	SHIUNIQ	ORDNUMBER	SHINUMBER	PONUMBER	RECEIPTNBR
▶	3265	ORD0000000000080	SH0000000000000000066	PO0000000483	

1

## Sage300 Connector

In this step we map the dataset onto the SAMLTD P/O Receipt transaction.

Transform Id

Sage300PORceipt

Transform Ids cannot contain the characters (&<>?\*)).

Select Sage300 ERP (Accpac) System

Sample Sage300 SAMLTD Connector ▼

Sage300 ERP (Accpac) Import Type

P/O Receipt ▼

Update Operation

Insert ▼

On the Field Mapping the P/O Receipt Header is mapped. In order to identify which purchase order to create a receipt for the PONUMBER field from O/E Shipment is set to the Purchase Order Number field of the receipt which is identical to the way a receipt is created in Sage300.

Transaction Id

Shipments ▼

Sage300 ERP (Accpac) Transaction Type

Receipts ▼

	Field Name	Type	Log Key	Sage300 ERP (Accpac) Field
▶	SHIUNIQ	Decimal	20	
	ORDNUMBER	Text	0	Description
	SHINUMBER	Text	0	Reference
	PONUMBER	Text	1	Purchase Order Number
	RECEIPTNBR	Text	0	Receipt Number

The Details are then mapped to the Receipt Lines.

To be able to identify and match which line to update within the receipt the POLINENUM field (the POLINE optional field) is mapped to the Purchase Order Line Sequence field.

To set the receipted quantity the Shipment Quantity field (QTYSHIPPED) is mapped to the Quantity Received field.

Transaction Id

ShipLines ▼

Sage300 ERP (Accpac) Transaction Type

Receipt Lines ▼

	Field Name	Type	Log Key	Sage300 ERP (Accpac) Field
▶	SHIUNIQ	Decimal	0	
	ITEM	Text	2	
	QTYSHIPPED	Decimal	0	Quantity Received
	POLINENUM	Text	0	Purchase Order Line Sequence
	PONUMBER	Text	1	

If the import is successful the RCPNUMBER should be populated.

PO0700: 1 Items   ( Showing 1 to 1 )   Generation Status: Complete					
	SHIUNIQ	DESSCRIPTIO	REFERENCE	PONUMBER	RCPNUMBER
▶	3265	ORD0000000000080	SH0000000000000000066	PO0000000483	RCP00000276



The corresponding Receipt is SAMLTD should have the Description and Reference populated with the O/E Order and O/E Shipment numbers from SAMINC respectively and the Quantities for the two lines should be populated with the quantities as per the shipment.

ASML61 - P/O Receipt Entry

File Settings Help

Receipt Number: RCP00000276

Vendor Number: SAMINT Intercompany - Sample Company Inc.

Receipt Taxes Additional Costs Optional Fields Totals

PO Number: PO00000483 From Multiple POs Job Related Retainage No. of Receipts: 1

Receipt Date: 5/2013 Posting Date: 5/2013 2013 - 05 Last Receipt No.: RCP00000276

Template: Bill-To Location: 1 Ship-To Location: 1

FOB Point: Ship-Via:

Terms Code: DUET Due by Invoice Date

Vendor Acct. Set: OTHE Accounts Payable, Other

Description: ORD000000000080 Reference: SH0000000000000000066

Lin...	Completes PO	Item Number	Item Description	Location	Drop-Ship	Quantity Received	Unit of
1	No	A1-103/0	Item Num.	1	No	1	Ea.
2	No	A1-401/0	Desk Calendar Pad	1	No	3	Ea.

Item/Tax... Calc. Taxes Receive All

Receipt Subtotal: 20.11 CAD

Post Invoice... History... Close

### Filter, 2<sup>nd</sup> Map & DB Write

The next three steps write back to the SAMINC O/E Shipment transaction to flag the EXPORTED optional field from False to True, to prevent it from being re-exported.

The logic in these steps is identical to the 1<sup>st</sup> integration and will not be explained in detail.

## SAMS200INVIMP (SAMPLE SAGE200 SALES INVOICE IMPORT)

This integration imports sale invoices from a CSV file into Sage200.

### CSV Read Transform




Extracts data from the CSV file.

Invoices: 4 Items   ( Showing 1 to 4 )   Generation Status: Complete									
	Customer	DocumentNo	LineNo	Description	NetAmount	TaxCode	VATAmount	TransactionDate	
▶	ABB001	IN00987	1	Monthly Fees	100	1	15	01/05/2013	C
	ABB001	IN00987	2	Excess Usage	20	1	5	01/05/2013	C
	MOL001	IN00988	1	Monthly Fees - May	145	2	0	01/06/2013	C
	NAN001	CN00091	1	Incorrect Charge	20	1	5	01/05/2013	C

### Hierarchy Transform

Transform the flat dataset into a hierarchical one which can be accepted into Sage200.

Invoices: 3 Items | ( Showing 1 to 3 ) | Generation Status: Partial Result Set

Customer	DocumentNo	TransactionDate	imanpkey0				
 ABB001	IN00987	01/05/2013	0				
InvoiceDetails: 2 Items							
Customer	DocumentNo	LineNo	Description	NetAmount	TaxCode	VATAmount	imanpkey1
ABB001	IN00987	1	Monthly Fees	100	1	15	0
ABB001	IN00987	2	Excess Usage	20	1	5	1
 MOL001	IN00988	01/06/2013	1				
InvoiceDetails: 1 Items							
Customer	DocumentNo	LineNo	Description	NetAmount	TaxCode	VATAmount	imanpkey1
MOL001	IN00988	1	Monthly Fees - May	145	2	0	2
 NAN001	CN00091	01/05/2013	2				
InvoiceDetails: 1 Items							
Customer	DocumentNo	LineNo	Description	NetAmount	TaxCode	VATAmount	imanpkey1
NAN001	CN00091	1	Incorrect Charge	20	1	5	3

### Map Transform

The Map transform prepares the dataset so it can be imported into Sage200.

Invoices: 3 Items | ( Showing 1 to 3 ) | Generation Status: Complete

Customer	InstrumentType	DocumentNo	TransactionDate	URN	CheckReference	imanpkey0
ABB001	INVOICE	IN00987	01/05/2013		False	0

InvoiceDetails: 2 Items

Customer	DocumentNo	LineNo	Description	NetAmount	TaxCode	VATAmount	NominalSpec	imanpkey1
ABB001	IN00987	1	Monthly Fees	100	1	15	31100-SCO-ADM	0
ABB001	IN00987	2	Excess Usage	20	1	5	31100-SCO-ADM	1

+	MOL001	INVOICE	IN00988	01/06/2013		False	1
+	NAN001	CREDITNOTE	CN00091	01/05/2013		False	2

**InstrumentType** – Uses a simple formula based on the first two characters of the DocumentNo field to determine if an invoice or credit note is to be raised.

**URN** – An empty field to capture the URN generated by Sage on import and then to report on the Audit Report.

**NominalSpec** – Set to a static value '31100-SCO-ADM'

## Sage200 Connector

The Sage200 connectors provides the mapping of the dataset into Sage200.

### Transform Id

Transform Ids cannot contain the characters (&<>?\*"!).

### Select Sage 200 System

### Sage 200 Import Type

### Update Operation

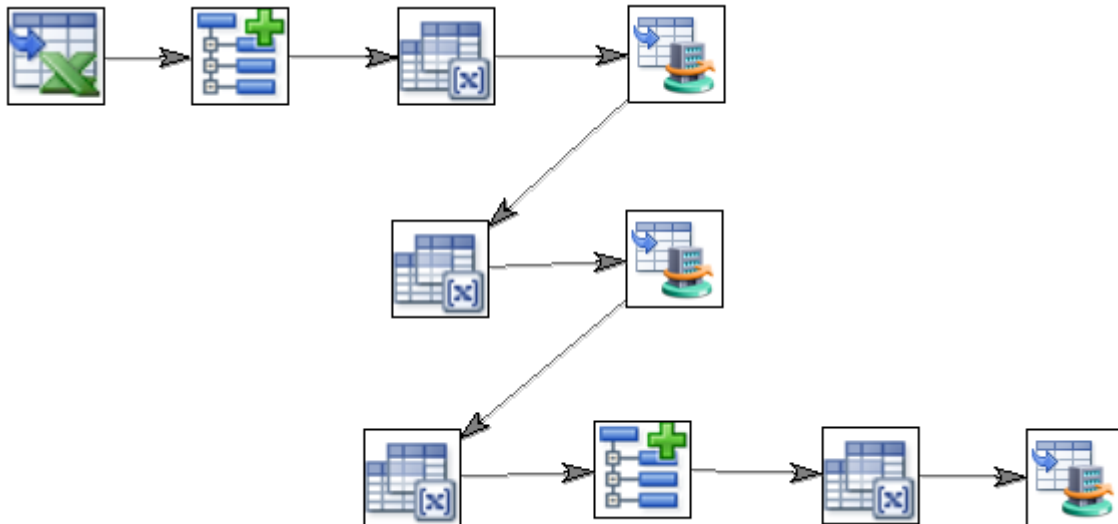
If the import is successful the URN field will be populated with the auto-generated Unique Reference Number.

## Preview

SLInvoice: 3 Items   ( Showing 1 to 3 )   Generation Status: Complete					
	Customer	InstrumentType	InstrumentNo	InstrumentDate	URN
▶	⊕ ABB001	INVOICE	DMOIN00987	01/05/2013	27329
	⊕ MOL001	INVOICE	DMOIN00988	01/06/2013	27330
	⊕ NAN001	CREDITNOTE	DMOCN00091	01/05/2013	27331

## SAMS200CONSPRNT (SAMPLE SAGE200 SOP INVOICE & CONSOLIDATED BILLING)

This integration demonstrates how IMan can be used to not only generate SOP Orders but to also create SOP Despatches and to Print & Post SOP Invoices.



### Read Transform

The Read transform extracts data from an Excel sheet. It is worth noting the data is extremely simple consisting of the bare minimum fields.

Record: 6 Items   ( Showing 1 to 6 )   Generation Status: Complete				
	ID	CUSTOMER	DESC	AMT
▶	1	ABB001	Line 1	10
	2	ABB001	Line 1	20
	3	KNO001	Line 1	30
	4	ABB001	Line 1	40
	5	LON007	Line 2	50
	6	KNO001	Line 3	60

### Hierarchy Transform

Transform the flat dataset into a hierarchical one consisting of a header and details which can be accepted into Sage200.

Record: 6 Items   ( Showing 1 to 6 )   Generation Status: Partial Result Set				
	ID	CUSTOMER		imanpkey0
▶	1	ABB001		0
Detail: 1 Items				
	ID	DESC	AMT	imanpkey1
	1	Line 1	10	0
+	2	ABB001		1

## Map Transform

The Map Transform adds a number of fields to the Dataset in readiness for it to be imported into Sage:

DocumentNo – Added to the header record to allow the Sage generated order number to be captured and reported.

LineType – Set to 1 to specify that Free Text lines will be generated.

LineId – Set to an empty value allowing the Sage LineId to be captured and used in the subsequent Despatch step.

Record: 6 Items   ( Showing 1 to 6 )   Generation Status: Complete					
	ID	CUSTOMER	DocumentNo	imanpkey0	
▶	1	ABB001			
Detail: 1 Items					
	ID	DESC	AMT	LineType	LineId
	1	Line 1	10	1	

## Sage200 SOP Order Connector

In this step we simply map the fields onto Sage to create the SOP Sales Order. If the orders can be successfully imported both the DocumentNo and LineId fields will be populated.

SOPOrder: 6 Items   ( Showing 1 to 6 )   Generation Status: Complete					
Customer		DocumentNo	imanpkey0		
▶	ABB001	0000005415	0		
SOPOrderLine: 1 Items					
	ItemDescription	UnitSellingPrice	SOPLineType	SOPOrderReturnLine	imanpkey1
	Line 1	10	1	1674493	0

## PreDespatchMap Transform

In this step an additional field is added to the Dataset to allow the despatch document to be created in Sage.

The DoDespatches field is added and the field's value is set to True. This field will be set to the 'Do Despatches' field in the next step, and indicates that a Despatch is to be created as opposed to a Return.

SOPOrder: 6 Items | ( Showing 1 to 6 ) | Generation Status: Partial Result Set

	Customer	DocumentNo	DoDespatches
▶	⊕ ABB001	0000005415	True
	⊕ ABB001	0000005416	True
	⊕ KNO001	0000005417	True
	⊕ ABB001	0000005418	True
	⊕ LON007	0000005419	True
	⊕ KNO001	0000005420	True

## Sage200 SOP Despatch Connector

This steps creates a despatch document from the Order. To create a Despatch against the order the the DocumentNo field is mapped to the Order Return No as shown below.

Transaction Id

SOPOrder ▼

Sage 200 Transaction Type

SOP Despatch/Receipt ▼

	Field Name	Type	Log Key	Sage 200 Field
▶	Customer	Text	0	Customer
	DocumentNo	Text	0	Order Return No
	DoDespatches	Boolean	0	Do Despatches

If Refresh is pressed and the Status goes to complete without displaying any errors the despatch will have succeeded.

**SOP - Amend Order**

Order Details | Delivery & Invoicing | Payment with Order

**Account selection**

☒ Customer account ☐ Cash account

A/C ref: ABB001

Short name: Abbey

Postcode: EH12 1WE

Name: Abbey Retail Ltd

By default supply from: WAREHOUSE

**Order detail**

Order no: 0000005418

Order status: Live

Document date: 03/06/2013

Date requested: 03/06/2013

Date promised: 03/06/2013

Customer order no:

☐ Show values ☒ Show quantities

Item	Description	Quantity	Allocated	Despatched	Invoiced	POP Order	Complete
Line 1		1.00000	n/a	1.00000	0.00000	No	No

**Exchange rate**

Rate: 1.000000 Currency: Pounds Sterling

**Subtotals**

Goods: £ 40.00

Charges: £ 0.00

Ord disc: £ 0.00

**Totals**

Net: £ 40.00

Tax: £ 6.97

Gross: £ 46.97

## PreInvoiceMap

This step adds to the IMan dataset the fields necessary to print & post the SOP Invoice.

- PostInvoice – Set to True will post the invoice to Sales Ledger.
- InvoiceNumber – Will allow the auto-generated Invoice Number to reported on the Audit report.
- ExportFile – This field generates file path for the exported invoice.
- UsesConsolidatedBilling – Performs a lookup against the Sage200 database to determine if the customer uses consolidated billing or not.
- ConsolidationID – When the invoices are generated we will respect the Use Consolidated Billing setting of the Customer. This field uses a conditional formula to either return the Customer or the Order number depending on the Consolidation value. It will then be used in a subsequent step to 'pivot' the data in a hierarchy transform.

In the screenshot below the Customer KON001 has been set to Use Consolidated Billing.



Adjustment: 6 Items   (Showing 1 to 6)   Generation Status: Complete						
OrderReturnNo	DoDespatches	PostInvoice	InvoiceNumber	UsesConsolidatedBilling	ConsolidationID	ExportFile
00005415	True	True		False	0000005415	C:\IMan\OutputData\0000005415.pdf
00005416	True	True		False	0000005416	C:\IMan\OutputData\0000005416.pdf
00005417	True	True		True	KNO001	C:\IMan\OutputData\KNO001.pdf
00005418	True	True		False	0000005418	C:\IMan\OutputData\0000005418.pdf
00005419	True	True		False	0000005419	C:\IMan\OutputData\0000005419.pdf
00005420	True	True		True	KNO001	C:\IMan\OutputData\KNO001.pdf

## ConsolidateInvoice

To group the invoices by their consolidation setting we use a Hierarchy transform to pivot on the ConsolidationID field.

Within the Hierarchy Setup the ConsolidationID field is used to provide the pivot by setting it to be the single key field in the hierarchy definition.

	Input Fld. Name	Type	Import	New Name	Key
▶	Customer	Text	<input checked="" type="checkbox"/>	Customer	0
	OrderReturnNo	Text	<input checked="" type="checkbox"/>	OrderReturnNo	0
	DoDespatches	Boolean	<input checked="" type="checkbox"/>	DoDespatches	0
	PostInvoice	Boolean	<input checked="" type="checkbox"/>	PostInvoice	0
	InvoiceNumber	Text	<input checked="" type="checkbox"/>	InvoiceNumber	0
	UsesConsolidatedBilling	Text	<input checked="" type="checkbox"/>	UsesConsolidatedBilling	0
	ConsolidationID	Text	<input checked="" type="checkbox"/>	ConsolidationID	1
	ExportFile	Text	<input checked="" type="checkbox"/>	ExportFile	0

When refreshed the dataset will now group any orders by their ConsolidationID.

SOPDespatchReceiptAdjustment: 5 Items   (Showing 1 to 5)   Generation Status: Complete							
	Customer	OrderReturnNo	DoDespatches	PostInvoice	InvoiceNumber	UsesConsolidatedBilling	ConsolidationID
⊞	ABB001	0000005415	True	True		False	00000
▶	ABB001	0000005416	True	True		False	00000
Orders: 1 Items							
	Customer	OrderReturnNo	ConsolidationID	imanpkey1			
⊞	ABB001	0000005416	0000005416	1			
⊞	KNO001	0000005417	True	True		True	KNO001
Orders: 2 Items							
	Customer	OrderReturnNo	ConsolidationID	imanpkey1			
⊞	KNO001	0000005417	KNO001	2			
⊞	KNO001	0000005420	KNO001	3			
⊞	ABB001	0000005418	True	True		False	00000
⊞	LON007	0000005419	True	True		False	00000

## ConsolidateInvMap

In order to produce a consolidated invoice, we need to construct a comma-delimited field for each of the orders that you wish to generate a consolidated invoice for.

To construct the comma-delimited string, we simply use the Concatenate function.

Current Field Name

New Field Name

Field Type

Text ▼


Log Key


Enable Script Evaluation

☒

Field Value/Script Formula

Concatenate("OrderReturnNo", "Orders", ",")


Check


No Errors Found.

When refreshed Customer KON001 now lists the two orders.

SOPDespatchReceiptAdjustment: 5 Items   ( Showing 1 to 5 )   Generation Status: Complete						
	Customer	OrderReturnNo	DoDespatches	PostInvoice	InvoiceNumber	ExportFile
▶	⊕ ABB001	0000005415	True	True		C:\IMan\OutputData\0000005415.pdf
	⊕ ABB001	0000005416	True	True		C:\IMan\OutputData\0000005416.pdf
	⊕ KON001	0000005417,0000005420	True	True		C:\IMan\OutputData\KNO001.pdf
	⊕ ABB001	0000005418	True	True		C:\IMan\OutputData\0000005418.pdf
	⊕ LON007	0000005419	True	True		C:\IMan\OutputData\0000005419.pdf

## Sage200 Invoice Print Post

In this step we map the fields into Sage to generate the Invoice.

**IMPORTANT!** Prior to running this IMan must be setup to allow invoices to be printed. Consult the IMan Sage200 user guide for details.

Transaction Id

SOPDespatchReceiptAdjustment ▼

Sage 200 Transaction Type






SOP Print/Post Invoice ▼

	Field Name	Type	Log Key	Sage 200 Field
▶	Customer	Text	0	
	OrderReturnNo	Text	0	Document No
	DoDespatches	Boolean	0	
	PostInvoice	Boolean	0	Post Invoice
	InvoiceNumber	Text	0	Invoice Credit No
	ExportFile	Text	0	Invoice Export File
	DocumentType	Integer	0	Document Type
	ConsolidationID	Text	0	
	UsesConsolidatedBilling	Text	0	

Pressing Refresh will print & post the invoice. If successful, the InvoiceCreditNo field will be populated.

SOPPostInvoice: 5 Items   (Showing 1 to 5)   Generation Status: Complete					
	DocumentNo	PostInvoice	InvoiceCreditNo	InvoiceExportFile	DocumentType
▶	0000005421	True	0000005133	C:\IMan\OutputData\0000005421.pdf	0
	0000005422	True	0000005134	C:\IMan\OutputData\0000005422.pdf	0
	0000005423,0000005426	True	0000005135	C:\IMan\OutputData\KNO001.pdf	0
	0000005424	True	0000005136	C:\IMan\OutputData\0000005424.pdf	0
	0000005425	True	0000005137	C:\IMan\OutputData\0000005425.pdf	0

The invoices will have been exported to file.

Name	Date modified	Type	Size
 0000005421.pdf	03/06/2013 15:10	Adobe Acrobat D...	8 KB
 0000005422.pdf	03/06/2013 15:10	Adobe Acrobat D...	8 KB
 0000005424.pdf	03/06/2013 15:10	Adobe Acrobat D...	8 KB
 0000005425.pdf	03/06/2013 15:10	Adobe Acrobat D...	8 KB
 KNO001.pdf	03/06/2013 15:10	Adobe Acrobat D...	5 KB

## SAMSAGECRMIM (SAMPLE SAGE CRM OPPORTUNITY QUOTE IMPORT)

The SageCRM import integration creates & updates Opportunities and Quotes in SageCRM. It takes data from an Excel spreadsheet, applies a multi-level hierarchical transformation, some expression based transformation, and finally imports the data into SageCRM.

*It is important that SageCRM is correctly setup as per the IMan Setup and CRM Setup sections in the CRM User Guide document.*

### 'Read' Transform

The read transform extracts data from the Excel spreadsheet.

### 'Hierarchy' Transform

To import the data into SageCRM it must be in a form or structure that corresponds to the type of data that is being imported. The data being imported consists of an Opportunity, with a Quote and QuoteItems.

The hierarchy transform will apply a 3 three level hierarchical structure to the data, one level for each entity.

To view the hierarchical transform press the Refresh button on the Field Mapping tab, then click '+' to expand the record.

Opportunity: 3 Items     Generation Status: Complete									
Company		QuoteDate	Description				imanpkey0		
	Cable Inc Coast	12/08/2010 00:00:00	Quote to Jim following his inquiry				0		
Quote: 1 Items									
Company		QuoteId	QuoteDate	Description				imanpkey1	
	Cable Inc Coast	WEB0234	12/08/2010 00:00:00	Quote to Jim following his inquiry				0	
QuoteLines: 3 Items									
	Company	QuoteId	LineNo	LineType	DetailDescription	Qty	Price	imanpkey2	
	Cable Inc Coast	WEB0234	1	f	Mini Lights	2	7.5	0	
	Cable Inc Coast	WEB0234	2	f	Desk Note Book (Black)	3	18	1	
	Cable Inc Coast	WEB0234	3	f	Calculator	6	54	2	
	AB McAlpine	21/08/2010 00:00:00	Warehousing Stock Inquiry				1		
	Alexander Motors	15/08/2010 00:00:00	Reference #77237				2		

To view how the hierarchical transform is setup click Edit button (pencil).

## Transaction Id to Hierarchise

Select the top most transaction id that is to be made into a hierarchy.

## New Transaction Id

&gt;

## Parent Id

Select the corresponding parent transaction id.



The selected fields are those that make-up the resulting Opportunity transaction.

The Key column is used to specify the field(s) used to define the relationship between each of the transaction types. In this example the 'Company' field is being used to uniquely identify each 'Opportunity' transaction.

	Input Fld. Name	Type	Import	New Name	Key
▶	Company	Text	<input checked="" type="checkbox"/>	Company	1
	QuoteId	Text	<input type="checkbox"/>	Quoteld	0
	QuoteDate	Text	<input checked="" type="checkbox"/>	QuoteDate	0
	Description	Text	<input checked="" type="checkbox"/>	Description	0
	LineNo	Integer	<input type="checkbox"/>	LineNo	0
	LineType	Text	<input type="checkbox"/>	LineType	0
	DetailDescription	Text	<input type="checkbox"/>	DetailDescription	0
	Qty	Decimal	<input type="checkbox"/>	Qty	0
	Price	Decimal	<input type="checkbox"/>	Price	0

To return to the previous screen press either the Green Tick or the Red Cross below the grid.

To view the Quote record, change the drop-down from Opportunity to Quote and press Edit.

New Transaction Id



Quote ▼

Parent Id

Opportunity ▼

Select the corresponding parent transaction id.

	Input Fld. Name	Type	Import	New Name	Key
▶	Company	Text	<input checked="" type="checkbox"/>	Company	1
	QuoteId	Text	<input checked="" type="checkbox"/>	QuoteId	2
	QuoteDate	Text	<input checked="" type="checkbox"/>	QuoteDate	0
	Description	Text	<input checked="" type="checkbox"/>	Description	0
	LineNo	Integer	<input type="checkbox"/>	LineNo	0
	LineType	Text	<input type="checkbox"/>	LineType	0
	DetailDescription	Text	<input type="checkbox"/>	DetailDescription	0
	Qty	Decimal	<input type="checkbox"/>	Qty	0
	Price	Decimal	<input type="checkbox"/>	Price	0

To uniquely identify the Quote record the Company and QuoteId fields are used.

Return to the previous screen by pressing the Green tick or Red Cross.

New Transaction Id



QuoteLines ▼

Parent Id

Quote ▼

Select the corresponding parent transaction id.

Change the drop-down from Quote to QuoteLines; you will see the Parent Id is set to Quote. Now press the Edit button to view the setup.

	Input Fld. Name	Type	Import	New Name	Key
►	Company	Text	<input checked="" type="checkbox"/>	Company	1
	QuoteId	Text	<input checked="" type="checkbox"/>	Quoteld	2
	QuoteDate	Text	<input type="checkbox"/>	QuoteDate	0
	Description	Text	<input type="checkbox"/>	Description	0
	LineNo	Integer	<input checked="" type="checkbox"/>	LineNo	3
	LineType	Text	<input checked="" type="checkbox"/>	LineType	0
	DetailDescription	Text	<input checked="" type="checkbox"/>	DetailDescription	0
	Qty	Decimal	<input checked="" type="checkbox"/>	Qty	0
	Price	Decimal	<input checked="" type="checkbox"/>	Price	0

To uniquely identify QuoteLines the previous two key fields are used (Company & QuoteId) in addition to the LineNo field.

### 'Map' Transform

The Map transform allows expressions and formulas to transform the field values.

This transform performs two lookups against CRM to obtain the value of some key fields.

#### CompanyId Field

To associate the imported opportunity with a company, we require the ID of the company record corresponding to its name. To achieve this, the SageCRMLookup function is used. Double click the CompanyId field to show the formula.

KWIOLMBL: 9 Items				
	Current Name	New Name	Type	Evaluate
*				
	Company	Company	Text	False
	QuoteDate	QuoteDate	Text	False
	Description	Description	Text	False
		CompanyId	Integer	True
		OpportunityDetails	Text	False
		AssignedTo	Integer	False
		CloseBy	Date/Time	True
		OpportunityId	Integer	True
		ImportSuccess	Text	False

The formula below queries the Company table to return the company ID field using the Company field in the dataset.

A more detailed explanation of the lookup formula(s) can be found in both the SageCRM Guide as well as the User Guide.

Current Field Name

New Field Name

CompanyId

Field Type

Integer

Log Key

0


Enable Script Evaluation

☒

Field Value/Script Formula

SageCRMLookup("SAGECRM", "Company", "comp\_companyid", Array  
("comp\_name"), Array(" & %Company & "), true)

Check

 No Errors Found.

Once complete press the Green Tick or Red Cross.

The following table describes the other fields with expressions.

Field	Notes
OpportunityDetails	A static text value set to 'Web Site Inquiry'
AssignedTo	A static integer value set to a specific user. Instead of using a static value, a lookup could be used to translate the name of the user to its underlying ID.
CloseBy	A formula to calculate the CloseBy date. The formula adds 21 days to the current date.
OpportunityId	<p>A lookup to obtain the ID of an opportunity within SageCRM. The lookup queries the Opportunity table using the Description field. If no match, the function does not return a value and the field remains empty.</p> <p>When importing into SageCRM, if the ID field of the entity is empty or set to '0' a new record will be created. When the field is populated with a valid ID the corresponding record is updated.</p> <p>This lookup should return no result on the first instance the</p>



	entire import is run; the second and subsequent times will the lookup will return a value and consequently an update will be performed to the existing record.
ImportSuccess	An empty field that will be used to store a textual value to denote if the record has been successfully imported or not.

## 'SageCRM' Connector Transform

The connector transform define how the dataset fields are mapped into the SageCRM.

The connector transform for the SageCRM import will map fields in all three of the dataset records to entities within SageCRM.

When opening the SageCRM connector, it may take up to 20 seconds to complete whilst a connection is made to SageCRM.

The 'Options' tab defines some keys settings to the import.

The SageCRM System selects which SageCRM instance to connect to; the settings are found in the System Connectors setup in the Setup area of IMan.

SageCRM Import Type selects the top-level entity that is being imported. This is set to Opportunity as this is the top-level entity in the dataset.

Update Operation defines how records are inserted/updated in SageCRM.

OPTIONS      FIELD MAPPING

Transform Id

SageCRM

Transform Ids cannot contain the characters (<>?\*)

Select Sage CRM System

Sage CRM 70 ▼

Sage CRM Import Type

Opportunity ▼

Update Operation

Insert/Update ▼

The Field Mapping tab defines how the entities and fields are mapped to the 'Import Type' selected on the Options tab.

Transaction Id

Opportunity

Sage CRM Transaction Type

Opportunity

Transaction Id is matched to Transaction Type (or SageCRM Entity)

	Field Name	Type	Log Key	Sage CRM Field
►	Company	Text	0	
	QuoteDate	Text	0	Created Date
	Description	Text	0	Description
	CompanyId	Integer	0	Company
	OpportunityDetails	Text	0	Details
	AssignedTo	Integer	0	Assigned To
	CloseBy	Date/Time	0	Close By
	OpportunityId	Integer	0	Opportunity
	ImportSuccess	Text	0	CRM Import Success

To map the fields, press the Edit button, and select the field from the field dropdown.

	Field Name	Type	Log Key	Sage CRM Field
	Company	Text	0	
	QuoteDate	Text	0	Created Date
	Description	Text	0	Description
►	CompanyId	Integer	0	Company
	OpportunityDetails	Text	0	Company
	AssignedTo	Integer	0	Created By
	CloseBy	Date/Time	0	Created Date
	OpportunityId	Integer	0	CRM Import Success
	ImportSuccess	Text	0	Customer Ref
				Decision Timeframe
				Deleted
				Description
				Details
				External Id
				Forecast
				Forecast Currency
				Forecast Scenario



Once complete press the Green Tick or Red Cross at the bottom of the grid.

Now change the 'Transaction Id' drop down at the top of the page from Opportunity to Quotes. Because the Quote transaction is mapped to the Quote entity, the SageCRM Transaction Type dropdown changes accordingly.

Transaction Id

Sage CRM Transaction Type


The connector uses metadata within SageCRM to extract related entities.

Type	Log Key	Sage CRM Field
Text	0	
Text	0	
Text	0	Created Date

Change the Transaction Id to QuoteLines; the Transaction Type is now set to Quote Items.

Pressing Refresh will initiate the import the transactions into SageCRM.

opportunity: 3 Items   Generation Status: Partial Result Set				
	oppo_createddate	oppo_description	oppo_primarycompanyid	oppo_no
▶	12/08/2010 00:00:00	Quote to Jim following his inquiry	1207	Web Site Inquiry
▶	21/08/2010 00:00:00	Warehousing Stock Inquiry	50	Web Site Inquiry
▶	15/08/2010 00:00:00	Reference #77237	71	Web Site Inquiry

1

At the top of the preview grid, check the Generation Status; initially this will be Partial Result Set. Press refresh until it changes to complete; this may take upwards for 20 seconds as SageCRM may need to startup.

When the import is complete, scroll the preview area to the right and you will find the OpportunityId field populated the auto-generated ID value. The import success field will also be populated with either Success or Failure.

oppo_assigneduserid	oppo_targetclose	oppo_opportunityid	SAGE200IMPSUCCESS
12	01/01/2012	368	Sucessfully Imported
12	01/01/2012	466	Sucessfully Imported
12	01/01/2012	467	Sucessfully Imported

If successful the Opportunity records with their quotes should all be imported.



## SAMSAGECRMEX (SAMPLE SAGECRM EXPORT)

---

This integration extracts Opportunities and Quotes from the Sage. This job demonstrates the SageCRM Reader transform which is uses powered by SData.

### 'Read' Transform

The SageCRM Reader transform provides the ability to extract data from SageCRM with the hierarchical structure between the data intact.

The Top Level Entity drop sets the entity that will be at the top of the resultant dataset; in this job it is set to Opportunity.

The filter statement provides a means to limit the dataset. The filter statement is in a SQL like language and is essentially a where clause in a SQL statement.

Transform Id  
  
Transform Ids cannot contain the characters (&<>?\*").

Select System


Top Level Entity

Result Filter Statement

The field mapping tab allows fields to be deselected from the dataset.

Press Refresh to generate the dataset. The result is a hierarchical dataset; click the '+' icons at the left hand side of the grid to view the child records.


opportunity: 1 Items || Generation Status: Complete

oppo_timestamp	oppo_notifytime	oppo_totalquotes	oppo_cust
 2010-11-29T12:24:00+00:00		\$ 3,515.00	

orders: 0 Items

imanpkey1

quotes: 1 Items

quot_workflowid	quot_timestamp	quot_consignee	quot_payterms	quot_rollup	quot_de
	2007-02-16T12:04:23+00:00			Y	Quote: 10 workshop

communication: 0 Items

comm_communicationid	comm_percentcomplete	comm_channelid	comm_status
----------------------	----------------------	----------------	-------------

notes: 0 Items

note_noteid	note_updateddate	note_deleted	note_channelid	note_createddate
-------------	------------------	--------------	----------------	------------------

library: 0 Items

imanpkey2

quoteitems: 3 Items

quit_promote	quit_erplacement	quit_listprice_cid	quit_listprice	quit_deleted	q
		\$	\$ 1,000.00		
		\$	\$ 400.00		
		\$	\$ 2,300.00		

### 'Map' Transform

In this job the map transform adds a new field 'RecType' to each record. On each record the field sets a different integer value.

### 'Write' Transform

The write transform exports the dataset to an Excel file.

## SCHEDULING AND RUNNING INTEGRATIONS

Integrations may be run in full or scheduled (to run in the background) from the Schedule area which is accessed by clicking Schedule at the top of the screen.

DESIGN

SCHEDULING

SETUP

To run or schedule an integration:

1. Select the integration from the drop down and double click the empty row in the schedule grid.
2. Select the Frequency, to run the job, set the Frequency to 'Now' and leave the start and end times as defaults.
3. Press the Green tick.

### JOB SETUP

Job Id

Sample Accpac O/E Order Import ▼ 1.

Schedule Description

Frequency

Now ▼ 2.

Start

11/08/2011 12:29 ▼

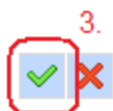
End

11/08/2021 12:29 ▼

Logon User

Only specify if you wish to schedule under the user name other than that of the scheduler service.

Password



4. After a few seconds the Audit Report will start to be displayed in the right-hand area.

The job when complete will send the same report via email to the email address specified in the Email Group setup.

## SUMMARY

### A/R Customer Account Import

3 Customer Processed. 0 Errors. 0 Inserted. 3 Updated.

### Order Entry Import

3 Orders Processed. 3 Orders Created. 0 Errors

Web Order WEB0234 - Accpac Order Number - ORD000000000214

Web Order WEB0237 - Accpac Order Number - ORD000000000215

Web Order WEB0238 - Accpac Order Number - ORD000000000216

Summary Section

Summaries are set in the Audit tab of each transform.

## DETAIL PROCESSING RESULTS

PROCESS DATE	PROCESS ID	RESULT CODE	SOURCE	MESSAGE
16/08/2011 12:18:58		0	TransformCoordinator	Starting job.
16/08/2011 12:19:00	Map	-1	WEB0237	Order Total [701.24] does not match calculated total [701.25]. Order will be put on hold.
16/08/2011 12:19:06	AccpacOrder	11022	WEB0237 - 1	0 - Information needed to set up the unit price does not exist for price list CANADA, item A1-320/0 in currency GBP. The price list record may not exist, or details have not been entered to price multiple units of measure.

Detail errors typically emitted by Applications i.e. Accpac, Sage200, etc.

## Audit Report

The audit report (shown above) is divided into two sections:

- a. Summary – The configurable summary section allows you to define the data you wish to appear on the report and in essence becomes the reconciliation report between your systems.

The Audit tab on each Transform setup screen is where the summaries are setup. The screenshot below shows the summary for Order Entry part of the summary above.



Action on Transform Error Reject Record ▼

Log Transform Start ☐

Log Transform End ☐

Log Warnings ☒

Report Group

RptGrp ▼




Summary Header

Order Entry Import **Heading shown on report.**

Audit Summary

%OE0520.PROCESSED Orders Processed. %OE0520.INSERTED Orders Create  
Web Order %OE0520.OrderId - Accpac Order Number - %OE0520.ORDNUMBER

**Summaries shown on report.**

- b. Detail Section – The detail section lists any errors or warnings. The errors/warnings are those emitted by the application during import.