



IMAN MAGENTO CONNECTOR USER GUIDE

Version 1.2.3

© 2013 EC Brands Corporation. All rights reserved.

EC Brands Corporation, EC Brands Corporation logos, and all EC Brands Corporation product and service names are registered trademarks or trademarks of EC Brands Corporation, or its affiliated entities. All other trademarks are the property of their respective owners.

EC Brands Corporation

Publisher

No part of this documentation may be copied, photocopied, reproduced, translated, or otherwise duplicated on any medium without written consent of EC Brands Corporation.

Use of the software programs described herein and this documentation is subject to the EC Brands Corporation License Agreement enclosed in the software package.

Contents

1) Overview	5
1.1) Typical Applications.....	5
Retail Web Sales	5
Business to Business Web Sales	6
1.2) Software	6
Realisable IMan.....	6
Magento Extensions	6
Magmi - Magento Mass Importer.....	6
2) Setup.....	6
2.1) Overview.....	6
Common Configurations.....	7
RAPID Web Sync™ Diagram.....	7
IMan	8
2.2) Magento Setup.....	8
Quick Guide	8
Installing the RAPID Web Sync™ Extension	8
Establishing Magento Access.....	9
Creating the API Role	9
Limiting API Access.....	11
Customers	12
Order-level Access	12
Catalog Products	13
Inventory Updates	14
Store IDs	14
Creating the API User	14
Assigning API Users to Roles.....	15
Installing Magmi	16
Installing the EC Internet Magmi Full Indexer plugin.....	17
Configuring Magmi	18
2.3) IMan Setup.....	19
Quick Guide	19
Required Files.....	19

Creating the System Connector Record.....	19
2.4) Lookup Tables	21
Overview	21
Setup	21
Viewing Results.....	23
3) Magento Connector Reference	26
3.1) Import Types Supported	26
3.2) Sales Orders - Update Order Status.....	26
3.3) Customer - Update / Create Customers.....	30
3.4) Item - Update / Create Products	34
General tab	34
Prices tab	35
Meta Information tab	35
Inventory tab	35
3.5) RAPID Web Sync	36
Inventory	36
Categories	36
Magento Extension.....	36
3.6) Shipment - Update / Create Shipments.....	37
3.7) Sage 300 ERP Examples.....	38
Pulling products from Sage 300 ERP	38
4) Magento Reader Reference	39
4.1) Export Types Supported.....	39
4.2) Orders - Download from Magento.....	39
Supported Types	39
Common Fields	39
4.3) Products - Download from Magento.....	40
Supported Types	40
Common Fields	40
4.4) Customers - Download from Magento.....	41
4.5) Invoices - Download from Magento	42
4.6) Shipments - Download from Magento	42

4.7)	Credit Memos - Download from Magento	43
4.8)	Order Download Settings.....	44
5)	Appendix 1 – Magento Version Notes (randy).....	46
6)	Appendix 2 – IMan Jobs.....	46
6.1)	Download Orders and Update Magento.....	46
6.2)	Update Orders	47
6.3)	Upload Inventory.....	47
6.4)	Adding & Updating Customers.....	47

1) Overview

The Magento Connector joins Magento web stores to Sage Software ERP and accounting systems. This integration allows automatic order processing for retail shopping sites, and it extends Magento support for Business to Business order taking.

The Magento Connector includes these capabilities:

- Add Magento Orders to Sage ERP
- Update Magento with inventory, availability, pricing and customer information
- Provide secure audit trail and useful exception reporting
- Support high-performance communication between Magento and Sage

The Magento Connector includes RAPID Web Sync™ technology to update Magento inventory data 100 times faster than standard integrations.

The Magento Connector uses XML-RPC data transfer protocol to communicate with the Magento API.

1.1) Typical Applications

The Magento Connector can perform basic operations like importing web orders into Sage ERP and accounting systems. The Magento connector has access to a wide range of transactions in the web store. Orders, Products, Customers, Invoices, Shipments, and Credit Notes can all be accessed and processed.

The Magento Connector supports uploading data into Magento. Records can be updated for products, orders, shipments, and customers with discrete API calls.

The Magento Connector also supports RAPID Web Sync™ that will perform mass add or update inventory items or customers across the Internet at speeds up to 2,000 records per minute. Products can be categorized in a product tree. And categories and sub categories can be created on the fly as items are added to Magento.

Retail Web Sales

For Retail Web Sales, the Magento Connector with RAPID Web Sync gives you the speed and flexibility to do imports of your entire catalog of products in mere moments. Now you can keep your webstore up to date at all times with no hassle. Moving from another ERP platform, or creating a new webstore from scratch? No problem! With the flexibility of IMan, you can pull from a number of existing ERP platforms and web stores, and quickly pass that data into your new Magento installation; customers included.

Business to Business Web Sales

For Business to Business Web Sales, the Magento Connector with RAPID Web Sync supports imports for tiered pricing, and deep integration with product inventory and multi-store support. Now you can import your entire business from Sage in little time at all, and keep your customer rules intact along the way.

1.2) Software

The Magento Connector relies on a number of software components in order to function to its full ability. Below you will find a list of software that you will need to familiarize yourself with.

Realisable IMan

The Magento Connector is built to work as part of Realisable Software's IMan integration manager. The Magento Connector readers and connectors appear in the developer palette within IMan. Learn more in Section 2.3.

Magento Extensions

The Magento Connector works with Magento Extensions that improve Magento's capability with Sage ERP systems, and that process bulk updates to Magento data.

Extensions are available to make input fields conform to the lengths required by Sage ERP, calculate Sage discount pricing, and to process mass updates in Magento.

- Sage Field Compatibility Extension – Limits address fields to fit into Sage ERP
- Sage B2B Pricing Extension – Calculates Sage 300 and Sage X3 pricing.
- RAPID Web Sync™ Extension – Processes mass updates for inventory and customers.

Magmi - Magento Mass Importer

Magmi is a field-proven mass import system that works independently of the Magento API. Magmi must be installed on the server supporting the Magento installation.

2)Setup

This section covers the installation and configuration of all extensions, applications, and modules needed to get your RAPID Web Sync up and running.

2.1) Overview

Setting up the Magento Connector is a moderately technical but straightforward task. If you've used IMan before initial setup of the Magento Connector can take as little as a half an hour. If you haven't, there are a number of resources available to you to help you get started.

In the below section, you'll find general setup guides for Magento, IMan, and the Magento Connector for IMan, as well as links to further IMan specific documentation

and use cases for the Magento Connector with examples. For those already well versed in IMan and Magento, initial setup is as follows:

1. Install the EC Internet Magento Connector into IMan.
2. Install the RAPID Web Sync extension into Magento.
3. Create the API Role and API User in Magneto
4. Create your first job in IMan to pull from your ERP Software and push into Magento via the Magento Connector

Common Configurations

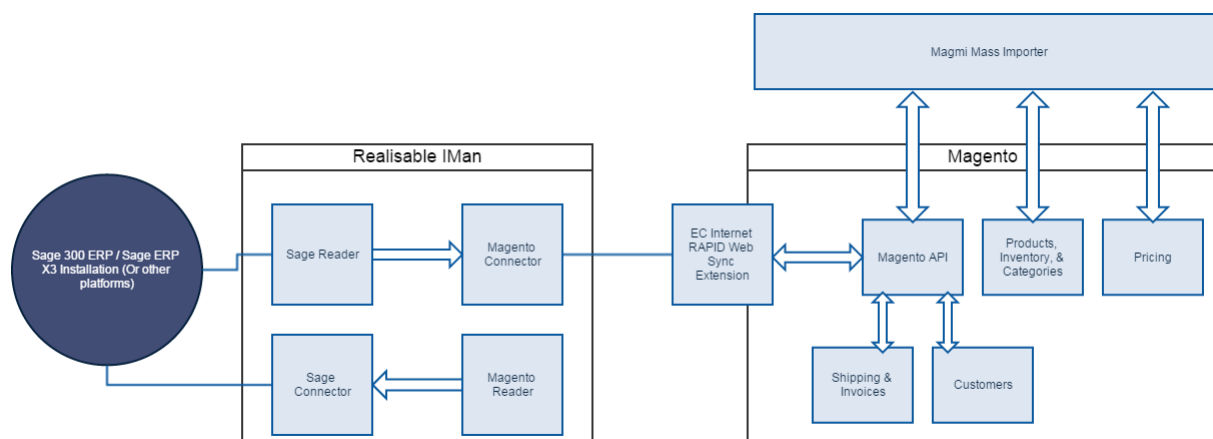
Below is a list of common configurations made with IMan and the Magento Connector with RAPID Web Sync.

- Local Sage, Remote Magento
 - o The standard “Business server” setup. IMan and Sage existing on a single server, with Magento on its own server. This allows you to keep your business records secure under a firewall, and maintain a public store at the same time.
- Local Sage, Local Magento
 - o All-local setup. IMan, Sage, and Magento all residing on the same server. These setups are exclusive to Windows hosts.
- Remote Sage, Remote Magento
 - o Allows for IMan to be installed on its own server. Sage and Magento can be on the same or different servers under this configuration, allowing for increased flexibility and security.

In all 3 above scenarios, only minor configuration changes in IMan are needed. A local connection uses the same resources and applications as a remote connection.

RAPID Web Sync™ Diagram

Below is a simplified diagram of how the entire RAPID Web Sync process works end-to-end.



IMan

Extensive IMan documentation and support is available at:

<http://www.realisable.co.uk/support/kb>

2.2) Magento Setup

Quick Guide

Setup in Magento mainly involves plugin installations and minor configuration. Initial setup is as follows:

1. Install the RAPID Web Sync Extension directly or from Magento Connect.
2. Create the API Role for IMan.
3. Create a new API User for IMan.
4. Assign your created Role to your new User.
5. Setup lookup tables for transaction counters, codes lists, translation tables, etc.

You may also want to install Magmi for faster product updates. A guide for this is available at the end of this setup section.

Once this is done you'll be able to initiate your first sync jobs directly from IMan. See the sections below for more a more in-depth installation guide.

Additional information on Magmi, including troubleshooting and general installation help can be found on the Magmi Wiki (<http://wiki.magmi.org/>).

Installing the RAPID Web Sync™ Extension

The RAPID Web Sync Magento Extensive gives you inside knowledge and information on your sync jobs right from the Magento Administration Area, and makes the necessary changes to the Magento API to allow seamless integration with IMan through the IMan Magento Connector.

To download this extension, visit the Magento Connect page at:

<http://www.magentoocommerce.com/magento-connect/rapid-web-sync.html>

If you received this extension outside of purchasing it at Magento Connect, it can be installed directly by copying the app folder from the extension archive directly into the root of your Magento installation (Typically the public_html folder).

Installing the Sage Field Lengths Extension

The Sage Field Limits Extension for Magento ensures that any data imported from your site is compatible with the SQL structure found in Sage 300 or Sage X3. Additionally, this plugin allows you to manually configure all of the field lengths to comply with SQL changes made in Sage, or other ERP platforms.

To download this extension, visit the Magento Connect page at:

<http://www.magentocommerce.com/magento-connect/field-compatibility-for-sage-300-erp-and-sage-erp-x3.html>

If you received this extension outside of purchasing it at Magento Connect, it can be installed directly by copying the app folder from the extension archive directly into the root of your Magento installation (Typically the public_html folder).

Establishing Magento Access

Creating API access in Magento is the first step in the setting up the Magento Connector.

API access will need to be configured before IMan can communicate with the Magento web store.

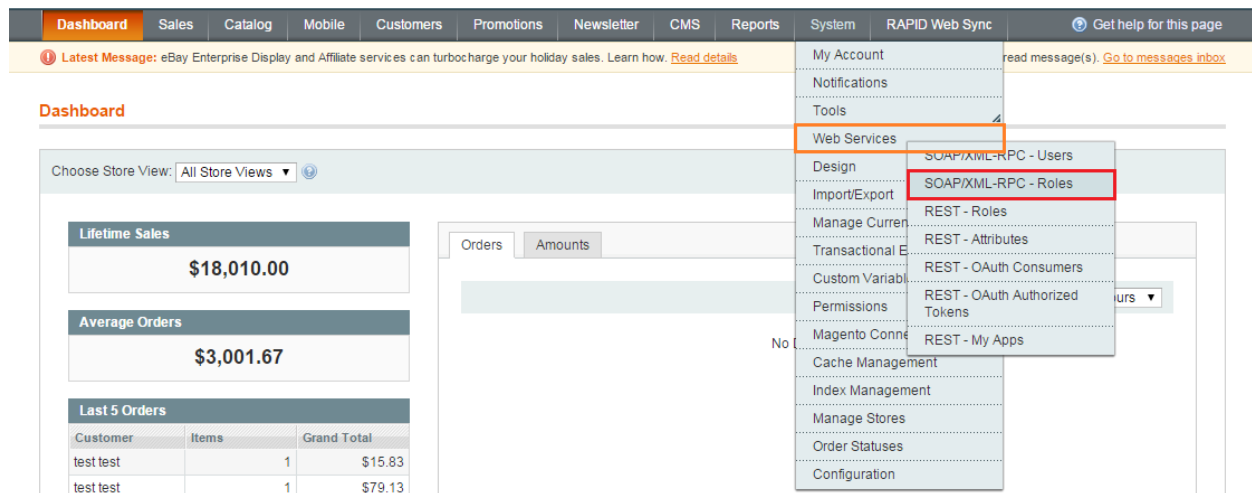
A Magento **Role** must be created by an admin-level user. This role will be assigned **Resources**, such as **Customers, Invoices, and Products** that will be read or updated by the Magento Connector. The Admin-level user will also create a Magento **API User** that is associated with the Role.

Creating the API Role

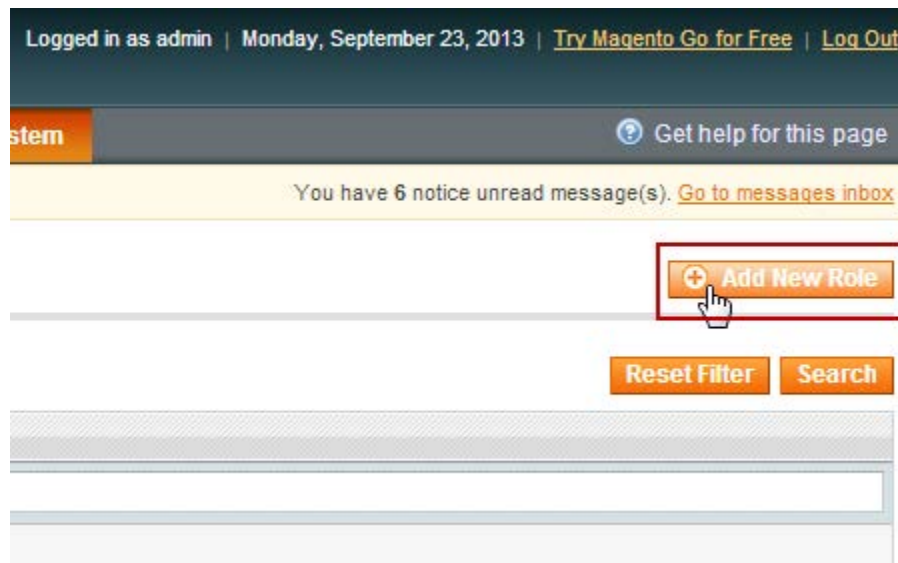
You will need access to the Admin section of Magento in order to set this up. Here are the screens you will use to create the role,

The image shows a screenshot of the Magento Admin Panel login interface. At the top, it says "Log in to Admin Panel". Below this, there are two input fields: "User Name:" with the text "admin" entered, and "Password:" with a masked password "*****". To the left of the password field is a link that says "Forgot your password?". To the right of the password field is an orange "Login" button. At the bottom of the form, there is a footer that reads "Magento is a trademark of Magento Inc. Copyright © 2013 Magento Inc."

Go to System → Web Services → SOAP/XML-RPC - Roles



If there is not already an API Role set up, you can create one by clicking **Add New Role**.



After clicking **Add New Role**, you will need to enter a "Role Name":

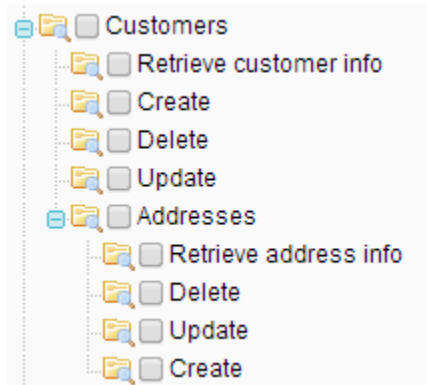
After setting a Role Name, you will have to assign Role Resources. You can access this screen by clicking **Role Resources** which is under **Role Information** on the left side of the screen:

Resource Access will be set to “All” for typical installations. You have the option of limiting API access to only essential data types.

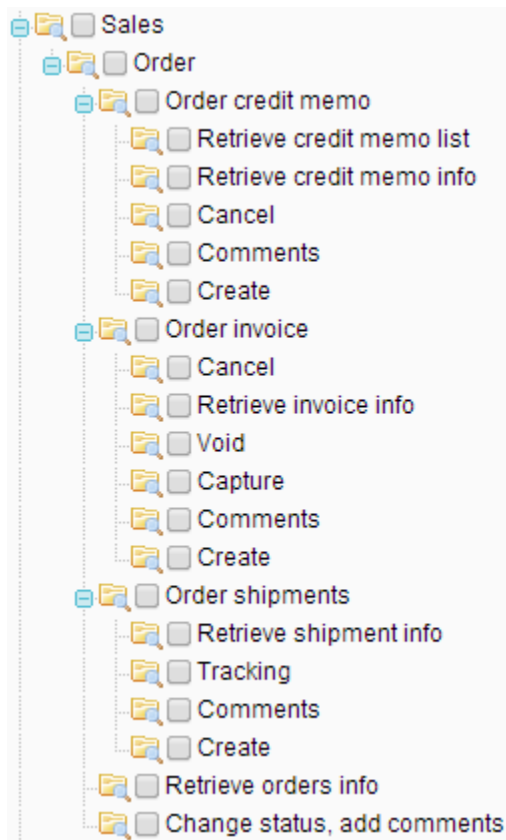
Limiting API Access

Magento supports limiting access by data type and operation. This is only required where data access is limited to certain organizations.

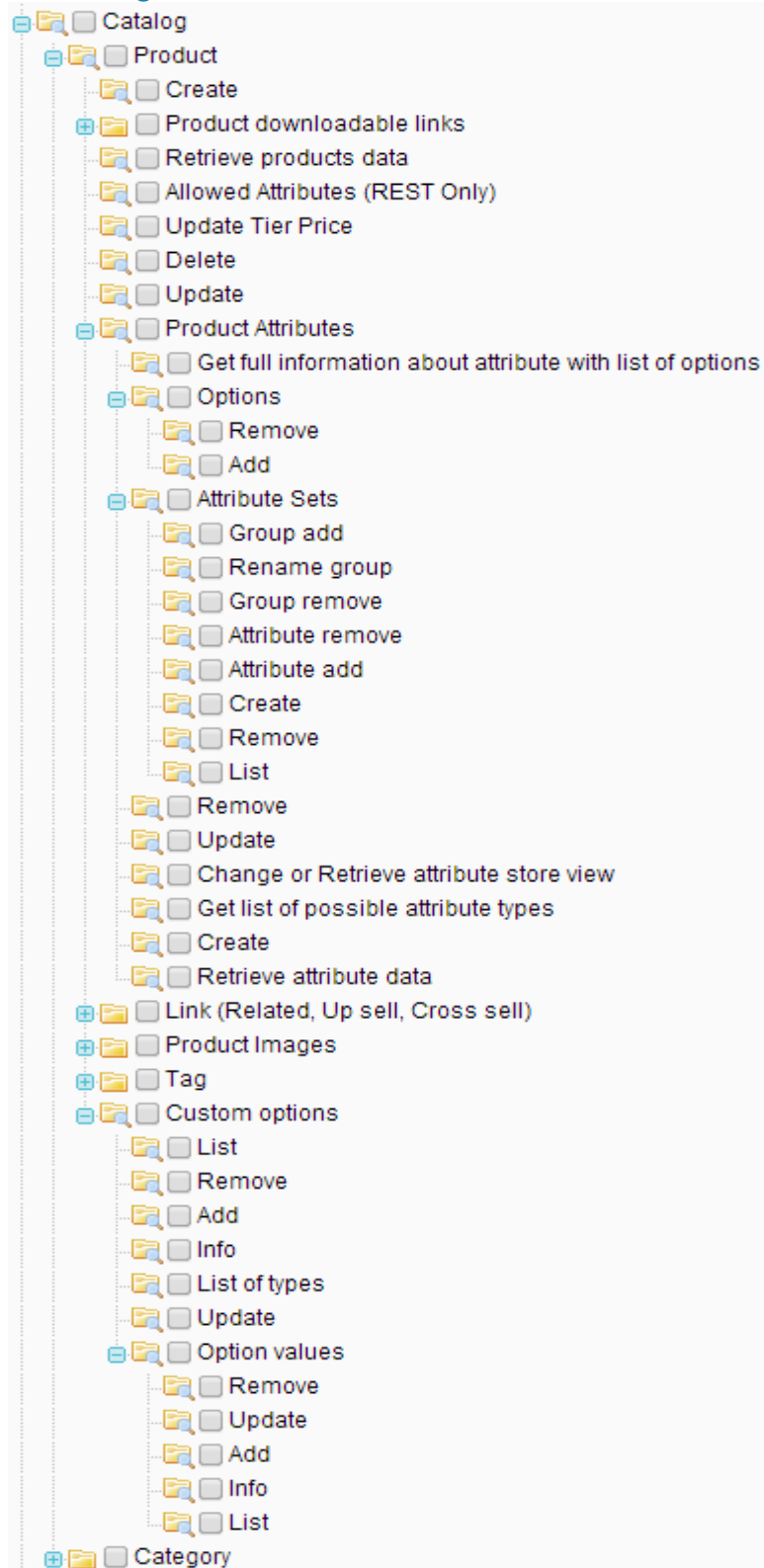
Customers



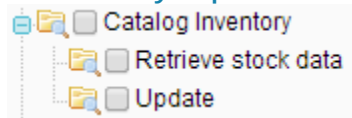
Order-level Access



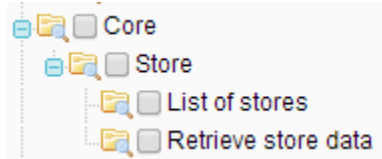
Catalog Products



Inventory Updates

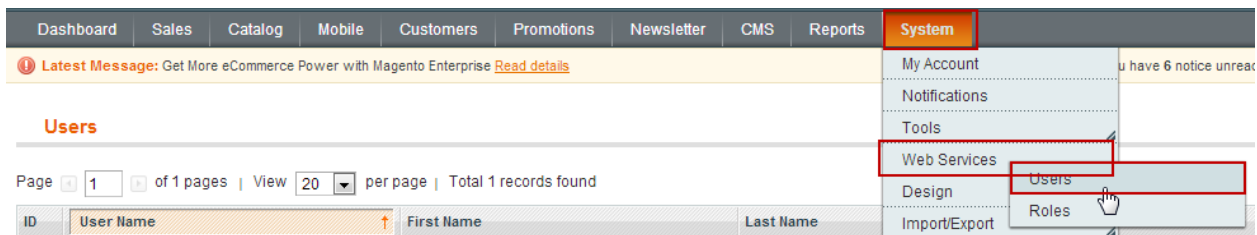


Store IDs

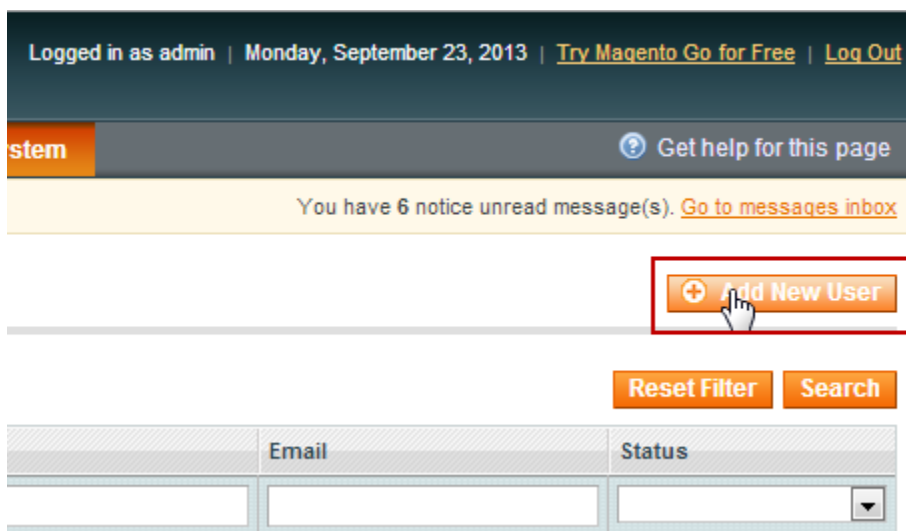


Creating the API User

Go to System → Web Services → Users



If there is not already an API User setup, you can create one by clicking **Add New User**.



On the New User page, you will have to fill out the following form for the user:

Account Information	
User Name *	
First Name *	
Last Name *	
Email *	
API Key *	
API Key Confirmation *	
This account is	Active v

Be sure to write down the User Name and API Key values.

You will need to enter these values in the connector setup in IMan.

Assigning API Users to Roles

You will now have to assign the newly created user to the previously created API Role.

This can be accomplished by clicking the radio button of the API Role you would like to assign to the user.

User Information	New User						
User Info User Role	Page 1 of 1 pages View 20 per page Total 1 records found <table border="1"> <thead> <tr> <th>Assigned</th> <th>Role Name</th> </tr> </thead> <tbody> <tr> <td>Any</td> <td></td> </tr> <tr> <td>☒</td> <td>API User</td> </tr> </tbody> </table>	Assigned	Role Name	Any		☒	API User
Assigned	Role Name						
Any							
☒	API User						

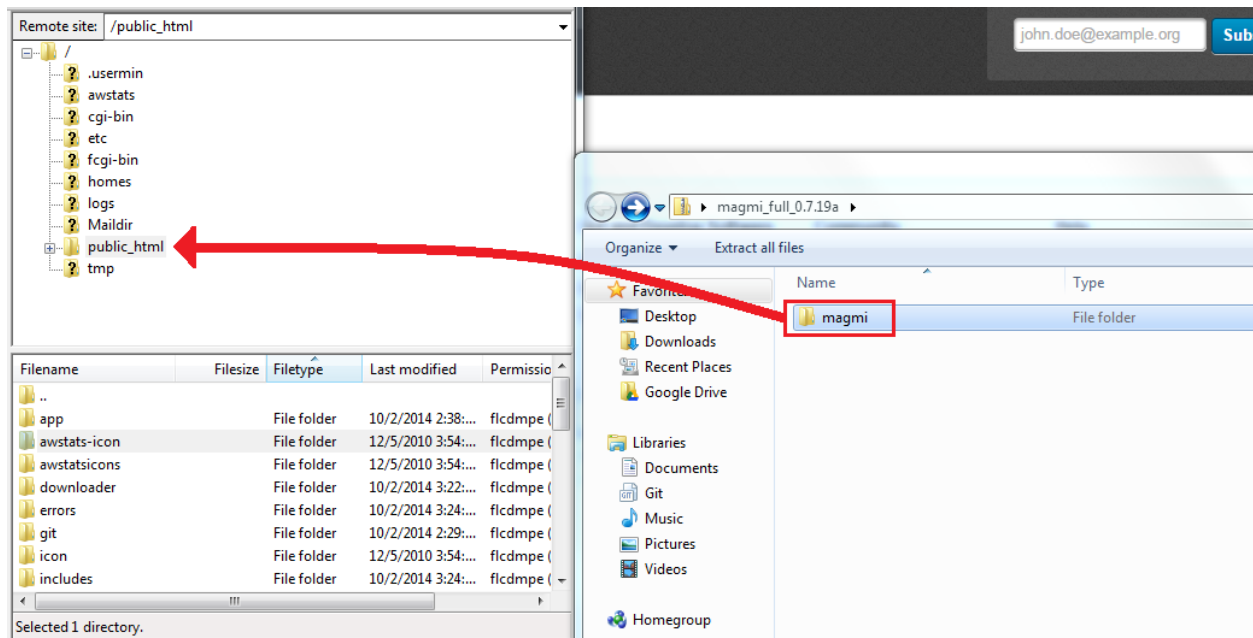
Installing Magmi

Magmi is one of the leading mass import tools for Magento, featuring lightning fast product imports, time-saving auto category generation features, multi-store imports, and safe defaults for products during imports. Magmi, along with the RAPID Web Sync, can save you countless hours.

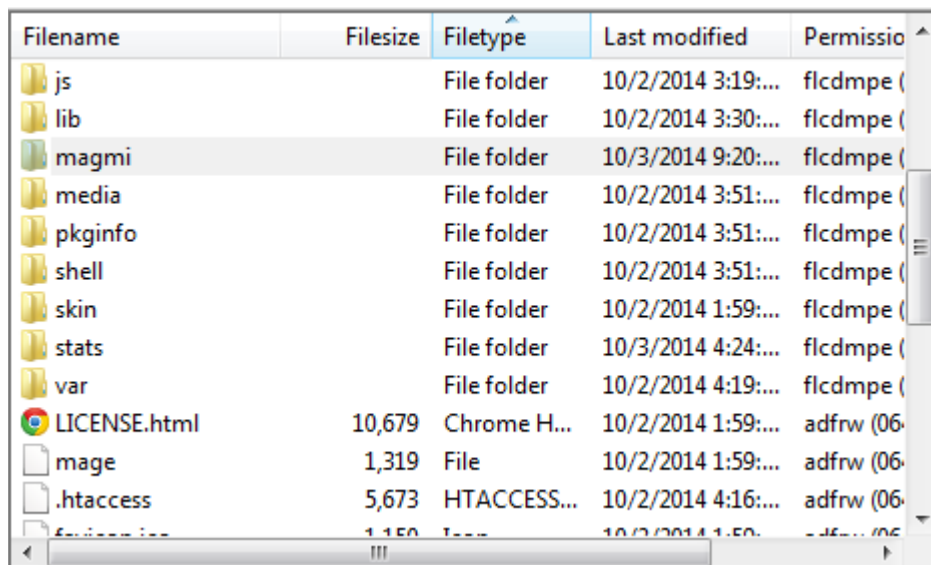
Installing Magmi is fairly straight forward and easy to do, and many hosting companies offer

To start, you'll need the full Magmi installations files. These are available to download at: <http://sourceforge.net/projects/magmi/>

These files will need to be deployed to your Magento installation either remotely via FTP or locally depending on your environment. Extract the Magmi archive and copy the contents of it to the root of your Magento installation.



When complete, your installation should look like this:



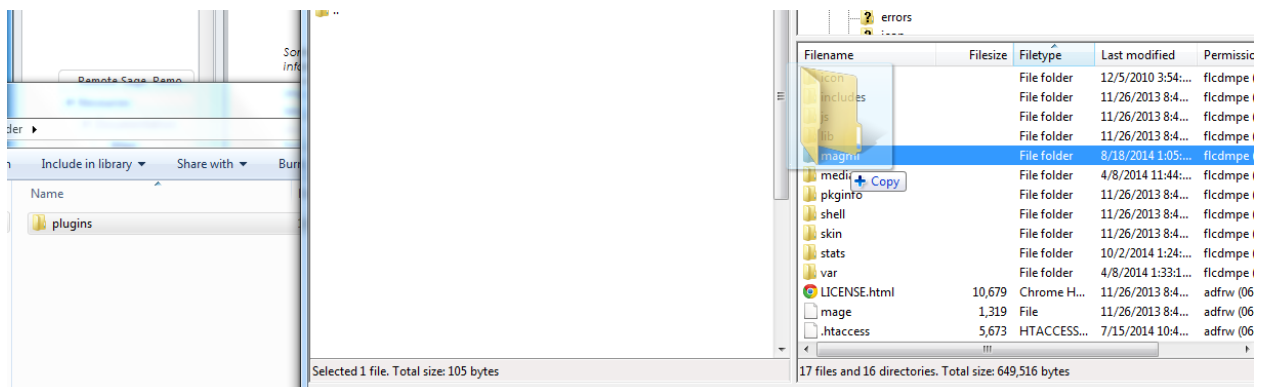
Filename	Filesize	Filetype	Last modified	Permissions
js		File folder	10/2/2014 3:19:...	flcdmpe (
lib		File folder	10/2/2014 3:30:...	flcdmpe (
magmi		File folder	10/3/2014 9:20:...	flcdmpe (
media		File folder	10/2/2014 3:51:...	flcdmpe (
pkginfo		File folder	10/2/2014 3:51:...	flcdmpe (
shell		File folder	10/2/2014 3:51:...	flcdmpe (
skin		File folder	10/2/2014 1:59:...	flcdmpe (
stats		File folder	10/3/2014 4:24:...	flcdmpe (
var		File folder	10/2/2014 4:19:...	flcdmpe (
LICENSE.html	10,679	Chrome H...	10/2/2014 1:59:...	adfrw (06
mage	1,319	File	10/2/2014 1:59:...	adfrw (06
.htaccess	5,673	HTACCESS...	10/2/2014 4:16:...	adfrw (06
...	...	...	...	...

Installing the EC Internet Magmi Full Indexer plugin

Some versions of Magento will not require this step. See [Appendix 1, Magento Versions](#) for more information.

Importing products with Magmi will often require reindexing of your product data in Magento in order for them to appear on the site. EC Internet offers an extension to automate this process. Upon completion of a RAPID Web Sync, Magmi will initiate a full reindex of your Magento site data automatically.

This plugin is packaged with the RAPID Web Sync extension for Magento. Extract the Magmi Full Indexer archive you received with your purchase of the RAPID Web Sync extension, and copy the **“plugins”** folder directly into your **magmi** folder on the server.



Your installation of the EC Internet Magmi Full Indexer plugin is now complete.

Configuring Magmi

You will now need to configure Magmi to access your Magento installation. This is done with the Magmi Web Interface, accessible via your_magento_installation/magmi/web/magmi.php.

Here you will be presented with a number of different settings panels. Under **Configure Global Parameters**, you will need to enter your Magento web server's SQL information.

By default, Magmi uses “**magento**” as the Database name, and the prefix is left as blank. Your situation may vary depending on how Magento was installed. Be sure to double check all parameters before saving or continuing.

For Magento installations later than 1.7.x, you will need to choose 1.7.x as your version in Magmi. Additional, your filesystem path for Magento should remain at the default of “**../..**”

Finally, enter your username and password for your SQL server. When done, your screen should resemble:

Configure Global Parameters		Saved: Fri Oct 3 09:45:14 2014	
Database		Magento	
Connectivity	Using host/port ▼	Version:	1.7.x ▼
Host:	localhost	Filesystem Path to magento directory:	../..
Port:	3306		
DB Name:	magento		
Username:	<<YOUR USERNAME>>		
Password:			
Table prefix:			
		Global	
		Reporting step in %:	0.5
		Multiselect value separator:	,
		Dir & File permissions	
		Directory permissions:	755
		File permissions:	644
Save global parameters			

Click **Save global parameters**. Next you'll need to enable a few of the plugins Magmi comes with, and the one we installed prior. This can be done in the **Configure Global Profile** section.

Check the following plugins to enable them:

- Magmi Optimizer v1.0.5
- ECIInternet Full Indexer v0.0.1 (*Installed in above section*)
- On the fly category creator/importer v0.2.4
- Tier price importer v0.0.9a

Click the **Save Profile** button at the bottom of the page. Your Magmi installation is now complete.

2.3) IMan Setup

This section covers general configuration and installation of the Magento Connector and Reader in IMan. For full IMan installation, please refer to the IMan user manual.

Quick Guide

IMan setup is relatively simple once the initial install process is completed. In the sections below we've outlined the steps needed to prepare an existing IMan installation for RAPID Web Sync. Please see the end of section 2.1 for full IMan installation information.

Required Files

Copy the below three files into the **Addins** folder of your IMan installation:

- ECInternet.Connectors.Magento.dll
- CookComputing.XmlRpcV2.dll
- Ez.Newsletter.MagentoApi.dll

Creating the System Connector Record

After adding the files to the **Addins** folder, you can create the **System Connector** to that particular Magento web store.

Navigate to **Setup** → **System Connectors**.

SYSTEM CONNECTORS

System ID:	Magento
System Type:	Magento Connector
Description:	Magento Webstore #1
Connection String:	http://www.samplemagentostore.com/index.php/api/xmlrpc
Username:	IMan
Password:	
Confirm Password:	
Company Id:	



Field	Notes
System ID	The internal Id of system connector.
System Type	Set to "Magento Connector"
Description	A description of the connection.
Connection String	[Magento Install] /index.php/api/xmlrpc May vary.
Username	Magento API User Name
Password & Confirm	Magento API Password (API Key)
Company	Not used.

2.4) Lookup Tables

Lookup Tables are what IMan uses to pull data back from other sources. In this section, we'll be covering the creation of Lookup Tables for your Magento installation in order to retrieve credit memos, invoices, and shipments for use in your ERP platform of choice.

Overview

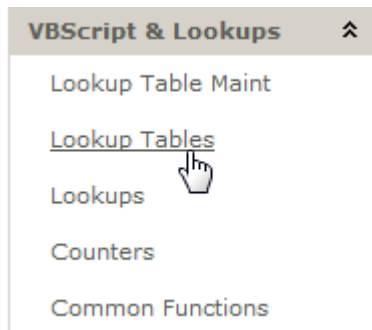
When processing orders from Magento in a typical integration, the user will pull the orders from Magento into their accounting package, and then update Magento with a new Order Status which effectively marks these orders as "processed."

There are a few downloadable "types" which do not have a mechanism such as "Order Status" to mark these records as processed. To overcome this, an internal counter is kept which will store the last record which was successfully downloaded from Magento.

Note: This table will only be updated if the user is running the job through the Scheduler page. Stepping through the job through the IMan Palette will not update this table.

Setup

Go to your IMan Setup screen. Under VBScript & Lookups, click the Lookup Tables link.



Create a new Lookup Table with an ID of "ECIMAG", and a meaningful Description.

There will be three fields which will need to be created: "**Last Number**", "**Last Retrieve Date Time**" and "**Message**".

Lookup ID:

Description:

	ID	Enabled	Formatted Name
▶	FLD01	True	Last Number
	FLD02	True	Last Retrieve Date Time
	FLD03	True	Message
	FLD04	False	
	FLD05	False	
	FLD06	False	
	FLD07	False	
	FLD08	False	
	FLD09	False	
	FLD10	False	
	FLD11	False	
	FLD12	False	

Last Number

Lookup ID:

Description:

Read Only ☒

Formatted Field Name

Field Type

Field Length

Last Retrieve Date Time

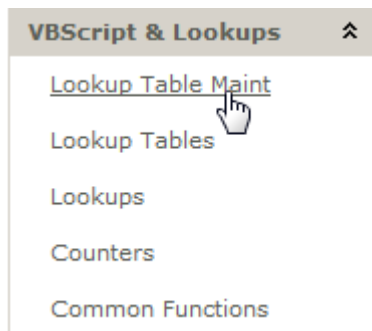
Lookup ID:	ECIMAG
Description:	EC Internet Magento Fields
Read Only	☒
Formatted Field Name	Last Retrieve Date Tim
Field Type	String Field
Field Length	30

Message

Lookup ID:	ECIMAG
Description:	EC Internet Magento Fields
Read Only	☒
Formatted Field Name	Message
Field Type	String Field
Field Length	100

Viewing Results

You can view the results of the Lookup Table by clicking the “Lookup Table Maint” link under “VBScript & Lookups” on the Setup page.



This is what the Lookup Table looks like for a multi-store Magento setup:

LOOKUP TABLE MAINTENANCE

EC Internet Magento Fields ▼

	Lookup Key	Lookup Result	Description
*			
	MAGE3-CRM-0		Credit Memo - Store 0
▶	MAGE3-CRM-1	Success	Credit Memo - Store 1
	MAGE3-CRM-2		Credit Memo - Store 2
	MAGE3-CRM-3		Credit Memo - Store 3
	MAGE3-INV-0		Invoice - Store 0
	MAGE3-INV-1	Success	Invoice - Store 1
	MAGE3-INV-2		Invoice - Store 2
	MAGE3-INV-3		Invoice - Store 3

◀▶1✎✕↺

The Lookup Key will be a unique string made up of the following:

“MAGE3” – The System Connector Id for the Magento installation.

“CRM” / “INV” / “SHIP” – The downloadable type:

“CRM” – Credit Memo

“INV” – Invoice

“SHIP” – Shipment

With the last part of the description matching the Store ID in a Multi-Store Magento install. Single-store Magento installations will always return 0.

The Lookup Result will have a value of “Success” if this System Connector / Downloadable Type / Store Id combination has been downloaded prior.

The Lookup Result will have a blank value if this System Connector / Downloadable Type / Store Id combination was tested for any new records and none were found. There are two scenarios where this would happen:

1. The user specifies a Store Id value when setting up a download which does not have any records of that particular Downloadable Type.

2. The user does not specify a Store Id value when setting up a download. In this case, the Magento Connector will query the Magento API for a list of all Stores in a particular Magento installation, and iterate through each of them, downloading the records.

In both of the above scenarios, a record will be added to the table. Clicking into the record will show the following data:

LOOKUP TABLE MAINTENANCE

EC Internet Magento Fields ▾	
Lookup Key:	MAGE3-CRM-0
Lookup Result:	
Description:	Credit Memo - Store 0
Last Number	1
Last Retrieve Date Time	9/10/2014 4:51:27 PM
Message	
 	

Clicking on a particular record – in this case “MAGE3-CRM-1” will show the following data:

LOOKUP TABLE MAINTENANCE

EC Internet Magento Fields ▼	
Lookup Key:	MAGE3-CRM-1
Lookup Result:	Success
Description:	Credit Memo - Store 1
Last Number	100000020
Last Retrieve Date Time	9/10/2014 4:51:59 PM
Message	Credit Memos downloaded.



Here is a breakdown of the data you see on this page:

Last Number	The increment_id value of the last successfully processed Credit Memo.
Last Retrieve Date	The date/time of the last time this job ran successfully.
Message	A simple message informing the user of the Credit Memos being downloaded.

3)Magento Connector Reference

This section covers a variety of things you may come across in Magento during your day to day use of the RAPID Web Sync.

3.1) Import Types Supported

Data can be updated in Magento without affecting the RAPID Web Sync Process.

3.2) Sales Orders - Update Order Status

Allows the user to update the “Status” of a particular order.

Also allows the user to optionally create an Order Comment which will contain an accounting system reference, or the tracking number for a particular order.

<u>Fields</u>	<u>Notes</u>
Increment_id	Required. Magento Order Number.
OrderNumberReference	Optional. If set, value will be added to Magento Order Comment.
NewMagentoStatus	Optional. If set, value will be used as new Magento Order Status.
AccpacTrackingNumber	Optional. If set, value will be added to Magento Order Comment.

Updates Update only.

Write-back Fields None.

Comment History breakdown for previously processed order:

Comments History

Add Order Comments

Status

Pending ▼

Comment

☐ Notify Customer by Email

☐ Visible on Frontend

Submit Comment

Aug 31, 2012 8:17:49 AM | Shipped

Customer Not Notified

Tracking Number: tracktrack

Aug 31, 2012 8:17:48 AM | Shipped

Customer Not Notified

New Order Status: Shipped for Order# ORD000000000296

Aug 31, 2012 8:14:24 AM | Downloaded

Customer Not Notified

Order has been downloaded as Order# ORD000000000296

Aug 30, 2012 1:48:21 PM | Pending

Customer Not Notified

Aug 30, 2012 - 1:48:21 PM | Pending

Order was created in Magento system.

Auto-assigned a Status of "Pending".

Aug 31, 2012 - 8:14:24 AM | Downloaded

Order was exported to accounting system.

Connector writes back to Magento an Order Status of "Downloaded" to indicate that the order has been downloaded to the accounting system.

Changing the order status will keep the order from being downloaded if IMan is looking for orders with status of “Pending”.

Aug 31, 2012 - 8:17:48 AM | Shipped

Order was fulfilled in the accounting system.

Connector writes back to Magento an Order Status of “Shipped” to indicate that the order has shipped.

Aug 31, 2012 - 8:17:49 AM | Shipped

When the order was fulfilled in the accounting system, a Tracking Number was assigned to the Sales Order with a value of “tracktrack”. This value will also be written to the Order Comment.

3.3) Customer - Update / Create Customers

Allows for the creation and updating of customers in a Magento webstore.

Addresses can be mapped in-line with the Customer Record, or as a child transaction of Customers. If there is a Customer Address which is in-line with the Customer Record, then it will have its **Is_Default_Billing** and **Is_Default_Shipping** flags set to true. Customer Addresses mapped in a child transaction will be added as additional Customer Addresses for the specified Customer, and has flags for setting default billing and shipping.

In-line Customer Address fields will be prefixed with "Address_"

ACCOUNT INFORMATION tab

<u>Fields</u>	<u>Notes</u>
Created_At	Example: "2014-08-13T10:04:44-07:00" This field is not usually updated.
Customer_Id	Required. If updating: used as lookup for Customer record. If inserting: written back to with newly created Id.
Default_Billing	Customer_Address_Id of default billing address.
Default_Shipping	Customer_Address_Id of default shipping address.
DOB	Format: "YYYY-MM-DD"
Email	Required. Used as customer login.
FirstName	Required. Customer first name.
Gender	See notes section below.
Group_Id	Customer Group Id. You must specify the Id of the Customer Group. You can find these values on the Customer Groups page which you can get to from the Customers dropdown from the admin back-end.
LastName	Required. Customer last name.
MiddleName	Customer middle name
Password	Plain text on creation. MD5 hash on update.
Prefix	Customer prefix (Dr., Ms., Mrs., Ms., etc.)
Store_Id	Store Id Customer is assigned to.
Suffix	Customer suffix (Jr., Sr., etc)
Taxvat	Tax VAT number for customer.
Website_Id	Required.

ADDRESSES tab

<u>Fields</u>	<u>Notes</u>
Customer_Address_Id	Required. Will be used as lookup if set, or written back to if CustomerAddress is created. This will be the unique identifier for the Customer Address record.
City	Required.
Company	Optional.
Country_Id	Required. "US" for United States, "CA" for Canada. Full list can be found by checking dropdown "<option>" values as described below in Notes section for Gender.
Fax	Optional.
FirstName	Required.
Is_Default_Billing	Default billing address flag.
Is_Default_Shipping	Default shipping address flag.
LastName	Required.
MiddleName	Optional.
PostCode	Required.
Prefix	Optional.
Region_Id	See notes section below.
Region	State/Province.
StreetLine1	Required.
StreetLine2	Optional.
Suffix	Optional.
Telephone	Required.

Notes

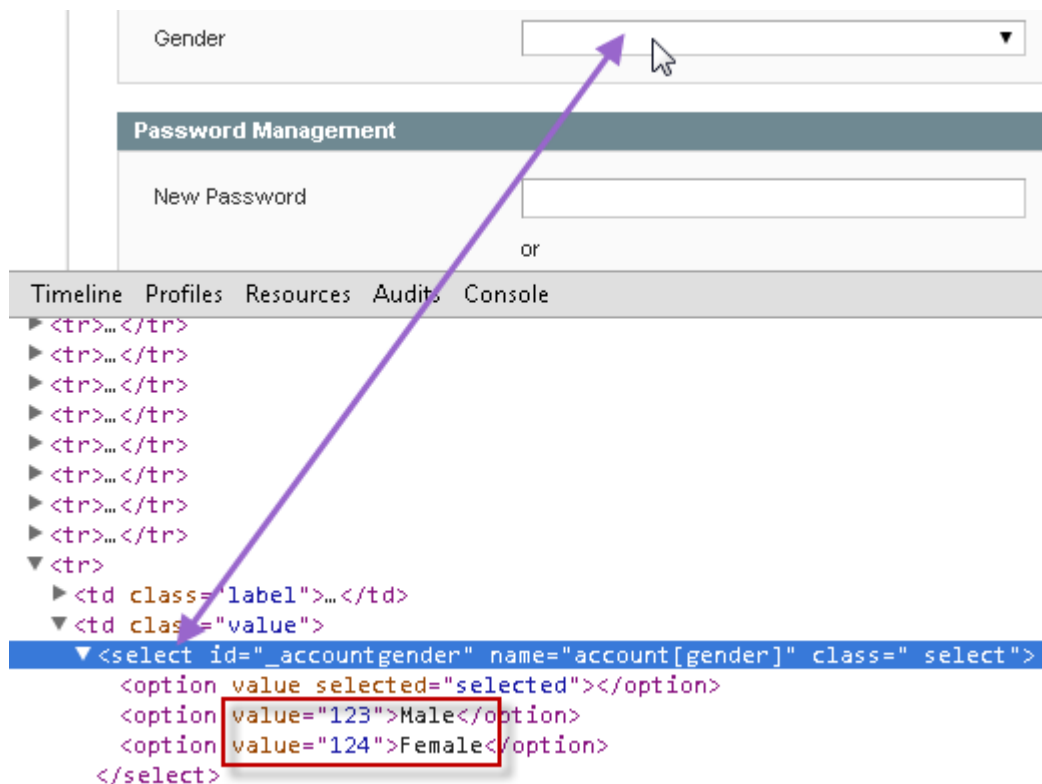
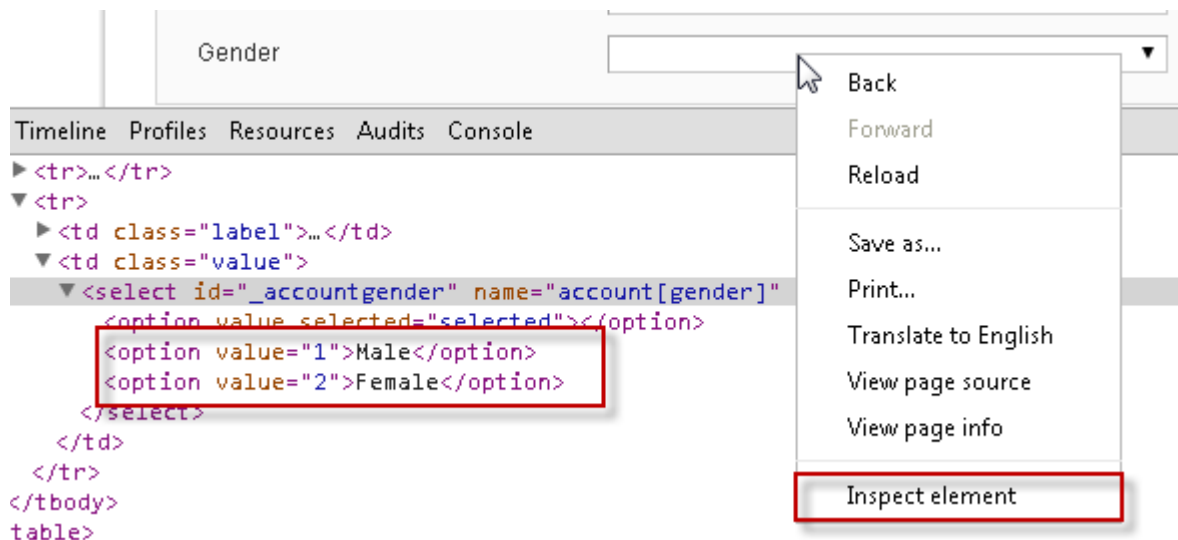
By default, a newly created customer (through the API) will receive a website value of "Admin".

Gender

In older versions of Magento, "1" is used for Male and "2" for Female.

In more recent versions, "123" is used for Male and "124" for Female.

You can confirm which values your system uses by checking the dropdown values of the gender dropdown for an existing customer. Go to the Customer screen in the admin backend, and pick a customer. Right-click the gender dropdown and click "Inspect Element." Use the values from the "<option>" for the particular sex you are looking for



Region Id

The value for Region_Id will be an integer. You can find the integer which matches the region you are looking for by checking the "State/Providence" dropdown, and looking for the "<option>" values.

The dropdown values will change depending on the current Country selected.

When the Country dropdown is set to “United States”, the “State/Providence” dropdown will be populated with all 50 states, as well as Armed Forces locations.:

```
▼ <select name="address[3][region_id]" id="_item3region_id" class="required-entry select" title>
  <option value></option>
  <option value="0">-- Please select --</option>
  <option value="1">Alabama</option>
  <option value="2">Alaska</option>
  <option value="3">American Samoa</option>
  <option value="4">Arizona</option>
  <option value="5">Arkansas</option>
  <option value="6">Armed Forces Africa</option>
  <option value="7">Armed Forces Americas</option>
  <option value="8">Armed Forces Canada</option>
  <option value="9">Armed Forces Europe</option>
  <option value="10">Armed Forces Middle East</option>
  <option value="11">Armed Forces Pacific</option>
  <option value="12">California</option>
  <option value="13">Colorado</option>
  <option value="14">Connecticut</option>
```

When the Country dropdown is set to “Canada”, the “State/Providence” dropdown will be populated with Canadian Providences:

```
▼ <select name="address[3][region_id]" id="_item3region_id" class="required-entry select" title>
  <option value></option>
  <option value="0">-- Please select --</option>
  <option value="66">Alberta</option>
  <option value="67">British Columbia</option>
  <option value="68">Manitoba</option>
  <option value="69">Newfoundland and Labrador</option>
  <option value="70">New Brunswick</option>
  <option value="71">Nova Scotia</option>
  <option value="72">Northwest Territories</option>
  <option value="73">Nunavut</option>
  <option value="74">Ontario</option>
  <option value="75">Prince Edward Island</option>
  <option value="76">Quebec</option>
  <option value="77">Saskatchewan</option>
  <option value="78">Yukon Territory</option>
</select>
```

3.4) Item - Update / Create Products

Allows the user to both add new products to a Magento webstore, as well as update certain fields on existing products.

Magento link for Product information:

[http://www.magentocommerce.com/wiki/1 - installation and configuration/adding-a-new-product](http://www.magentocommerce.com/wiki/1_-_installation_and_configuration/adding-a-new-product)

General tab

These are some differences in the data below depending on the version of Magento being used. Please see Appendix 1 – Magento Versions towards the end of this document.

Fields	Notes
SKU	Required. Magento unique SKU value
Name	Required. Full product name.
Description	Required. Long product description which will show in the “Details” section of the product page.
Short_Description	Required. Short product description which will be used in the catalog page and as the “Quick overview” on the product page.
Weight	Required. Product weight field. If the weight of the product is not known, Magento recommends setting this value to “1”.
Status	Required. Product status. Product status must be “Enabled” in order to be visible in the catalog. If you are managing stock (Inventory Tab → Manage Stock), then the Stock Availability must be set to “In Stock” to show. 1 – Enabled 2 – Disabled (blank) – Sets dropdown to “Please Select”
Country_Of_Manufacture	Country code for Manufacturing country.

Prices tab

These are some differences in the data below depending on the version of Magento being used. Please see Appendix 1 – Magento Versions towards the end of this document.

Fields	Notes
Price	Ordinary selling price for product.
MSRP_Enabled	“Apply MAP” in Magento admin 0 – No 1 – Yes 2 – Use Configuration
MSRP_Display_Actual_Price_Type	“Display Actual Price” in Magento admin 0 1 – On Gesture 2 – In Cart 3 – Before Order Confirmation 4 – Use Configuration
MSRP	
Enable_Googlecheckout	

Meta Information tab

Meta information is primarily used for SEO purposes.

Fields	Notes
Meta_Title	
Meta_Keyword	
Meta_Description	

Inventory tab

Fields	Notes
Inventory	Available quantity of product
Is_In_Stock	1 – In Stock (Any value other than “1”) – Not In Stock

Notes

SKU is the only product field which is required to create a product.

Status is required for the product to show up in the admin backend.

There are certain fields which appear “required” in the Magento backend screen, but when creating products using the API, the system will allow blank values in some of these fields. When you go into the Admin backend, however, you will be unable to save the product until the required fields have been completed.

It is our recommendation that all required fields be filled out.

3.5) RAPID Web Sync

Below is an overview of all the features of the RAPID Web Sync.

Inventory

The RAPID Web Sync extension is capable of managing Inventory within Magento using the inventory data present within your ERP platform of choice. Inventory updates are automatic and can be synced as often as needed.

Categories

The RAPID Web Sync extension allows for automatic creation of categories in Magento directly from Sage 300 ERP and Sage ERP X3. These categories are generated using the category IDs from Magento. Their name and description can then be permanently set within the Magento web admin by navigating to **Catalog -> Manage Categories** within the Magento admin panel.

Magento Extension

The RAPID Web Sync Magento Extension gives you inside knowledge of the status of your sync jobs without having to leave Magento. From the Magento Admin Panel, select **RAPID Web Sync -> View Logs**.

EC Internet RAPID Web Sync Log

Show

100 ▼

 entries

Search:

	Date	Job Status	Started At	Completed At	Duration	Job ID
✓	September, 29, 2014	5 of 5 products inserted / updated successfully.	11:05:08 PM	11:05:08 PM	< 1s	5429e5a41ba99
✓	September, 29, 2014	5 of 5 products updated successfully.	09:16:34 PM	09:16:34 PM	< 1s	5429cc32426b5
✓	September, 29, 2014	5 of 5 products updated successfully.	09:11:34 PM	09:11:34 PM	< 1s	5429cb0647ae6
✓	September, 29, 2014	5 of 5 products updated successfully.	09:06:34 PM	09:06:34 PM	< 1s	5429c9da49118
✓	September, 29, 2014	5 of 5 products updated successfully.	09:01:34 PM	09:01:34 PM	< 1s	5429c8ae383fa

This gives you a complete list of every job run on the site sorted by order ran, along with the type of job that was run, the number of items completed, and the duration of the job.

Note: All times shown by the RAPID Web Sync Log are displayed using the same timezone your PHP installation is configured to.

3.6) Shipment - Update / Create Shipments

Allows the user to both update existing Shipments, as well as create new Shipments for existing orders.

Fields	Notes
Shipment Number	Magento Shipment Number. When attempting to Update, this field will be the key. When attempting Insert, field will be written back to with new Shipment Number.
Magento Order ID	Magento Order Number. Field is necessary for Insert, but not for Update. When attempting Insert, we will attempt to create a new Shipment for this order.
Comment	New Shipment Comment Not necessary for Updates or Inserts If set, Comment will be added to the Shipment.
Carrier	Tracking Information "Carrier". Sets "Carrier" on new Tracking information record. Values can be obtained from inspecting "Carrier" dropdown (use the <option>'s value attribute, eg: dhl, fedex, ups): <pre><select name="carrier" class="select" style="width:110px" onchange= <option value="custom">Custom Value</option> <option value="dhl">DHL</option> <option value="fedex">Federal Express</option> <option value="ups">United Parcel Service</option> <option value="usps">United States Postal Service</option></pre>
Title	Tracking Information "Title". Sets "Title" on new Tracking Information record.
Tracking Number	Tracking Information "Tracking Number". Sets "Number" on new Tracking Information record.
Updates	Update / Insert
Write-back Fields	Shipment Number (on Insert)

Completed Shipment Tracking Information Record:

3.7) Sage 300 ERP Examples

Pulling products from Sage 300 ERP

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 and P.CURRENCY = D.CURRENCY and D.DPRICETYPE = 1)

where ALLOWONWEB = 1

4) Magento Reader Reference

This sections covers the Magento Reader for IMan, which allows the user to pull data from Magento back into IMan.

4.1) Export Types Supported

Export Type	Supported Features
Orders	Order Details Order Detail Options
Items	Product Tier Prices
Customers	
Invoices	Invoice Items Invoice Comments
Shipments	Shipment Details Shipment Tracking Numbers Shipment Comments
Credit Memos	Credit Memo Items Credit Memo Comments

4.2) Orders - Download from Magento

Allows the user to export Sales Orders from Magento by “Order Status”.

The first Payment method for this order is in-line with the Order Header.

Supported Types

Orders	Order Header Record
Order Details	Order Detail Record
Order Detail Options	Product Options for Order Details
Order Payments	Payments made for Order

Common Fields

Orders

Billing_Address_*	Customer Billing Address fields
Increment_Id	Magento Order Number
Customer_Email	Customer Email Address
Shipping_Address_*	Customer Shipping Address fields
Payment_*	First Payment record

Order Details

Name	Product name
Price	Product price
Qty_Ordered	Product quantity
SKU	Product sku
Weight	Product weight

Order Detail Options

Price	Option price
SKU	Option SKU
Title	Option title

Order Payments

Payment_Method	Method of payment
Payment_Amount_Ordered	Payment for ordered products.
Payment_Base_Shipping_Amount	Payment for shipping amount.
Payment_CC_Exp_Month	Credit Card expiration month.
Payment_CC_Exp_Year	Credit Card expiration year.
Payment_CC_Last4	Last 4 digits of Credit Card.

4.3) Products - Download from Magento

Allows the user to download product data for all products in their Magento catalog.

Supported Types

Items	Product Data
Product Tier Prices	Tier Prices set for Product

Common Fields

Items

Cost	Product cost
Description	Product description
Meta_*	Meta data fields
Name	Product name
Price	Product price
Product_Id	Product Id
SKU	Product Sku
Status	Product status
Updated_At	Last updated date time
URL_*	URL data fields

Visibility	Level of product visibility
Websites	List of websites which use the product.
Weight	Product weight

Product Tier Prices

Customer_Group_Id	Customer Group Id
Price	Tier price
Qty	Qty break for tier
Website	Which websites use this product tier

4.4) Customers - Download from Magento

Allows the user to download Customer data for all Customers in their Magento webstore.

Supported Types	Customers – Customer Data
Common Fields	Customer_Id – Customer Id Email – Customer email Firstname Customer first name Group_Id – Customer Group Id Lastname – Customer last name Updated_At Last updated date time

4.5) Invoices - Download from Magento

Allows the user to download Invoice data for all Invoices in their Magento webstore.

Supported Types	Invoices – Invoice Data Invoice Items – Invoice Line Item Data Invoice Comments – Comments on Invoices
Common Fields	Invoices Invoice Items Invoice Comments

4.6) Shipments - Download from Magento

Allows the user to download Shipment data for all Shipments in their Magento webstore.

Supported Types	Shipments – Shipment Data Shipment Details – Shipment Line Item Data Shipment Tracking Numbers – Tracking Number records Shipment Comments – Comment records for Shipment
Common Fields	Shipments Shipment Details Shipment Tracking Numbers Shipping Comments

4.7) Credit Memos - Download from Magento

Allows the user to download Credit Note data for all Credit Notes in their Magento webstore.

Supported Types	Credit Memos – Credit Memo data Credit Memo Items – Credit Memo Line Item data Credit Memo Comments – Credit Memo Comment records
Common Fields	Credit Memos Credit Memo Items Credit Memo Comments
Notes (May)	There are a few fields on the Credit Memo header which seem redundant: - order_id and order_increment_id - creditmemo_id and increment_id order_id and creditmemo_id can be thought of as the 'shorthand' value for the 'full' value of order_increment_id and increment_id respectively. Example: order_id = 1 & order_increment_id = 100000001 Creditmemo_id = 2 & increment_id = 100000002 In both cases the "increment_id" values include the store id (1 in this case) followed by 0's, and then finally the order_id / creditmemo_id value.

4.8) Order Download Settings

The Order Download takes advantage of IMan's **Options** which allow you to set parameters on the script which talks to the Magento API and pulls down the orders.

Export Entity

Order ▼

Order Status (Defaults to 'pending')

Pending

Number of Records to Pull (Defaults to 'all')

5

Starting Increment Id Number (Defaults to '1' if empty or if Store_ID list is blank)

Comma-Separated Custom Header Field List

Comma-Separated Custom Detail Field List

Comma-Separated Store_ID List (Leave blank for 'all')

1

Export Entity

The export entity must be set to **Order** for these options to take effect.

Order Status

Limit the orders downloaded to those with a certain Order Status.

Common values: "Pending", "Processing".

Defaults to "Pending" which is the default Magento status for a new order.

Number of Records to Pull

Set a limit of the amount of records to download at once.

Defaults to pulling all records.

Starting Increment_Id Number

Set the initial Magento Order Number to start downloading orders. Orders will only be downloaded if they have an order number equal to, or greater than this value.

Defaults to “1” if empty, or if store_id list is empty.

Comma-Separated Custom Header Field List

Put any custom fields here which will be on the header.

Comma-Separated Custom Detail Field List

Put any custom fields here which will be on the item (detail).

Comma-Separated Store_Id List

Put a list of any stores to pull orders from.

Defaults to all stores if empty.

5)Appendix 1 – Magento Version Notes (randy)

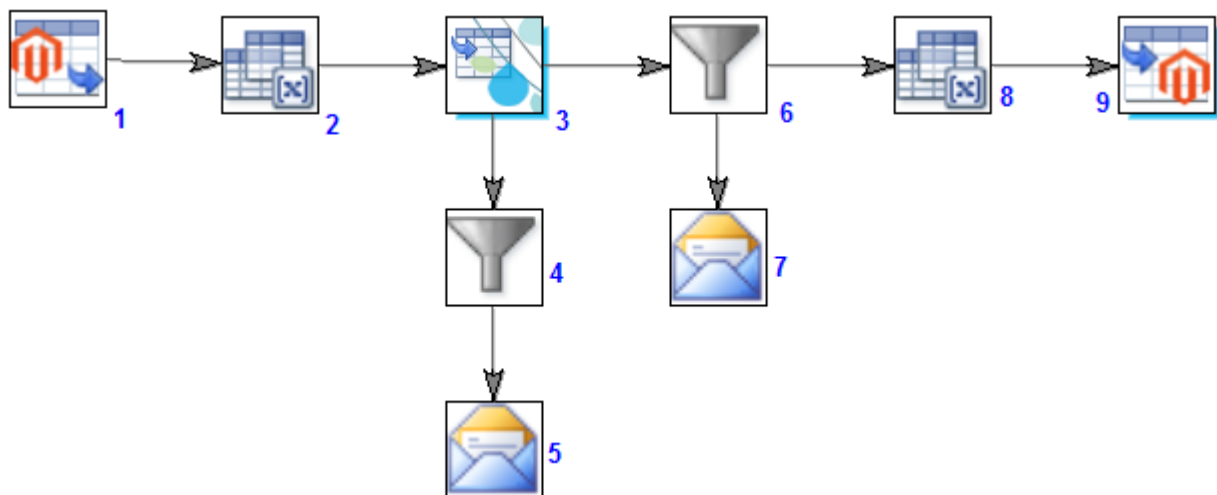
The RAPID Web Sync Extension has been tested on as far back as Magento 1.4.2 using the Magento API to complete imports, and as far back as 1.8 using Magmi to complete imports.

Magento's fields can sometimes change between versions. Below is a list of all known differences between Magento Versions of importance to these Sync jobs.

<u>Version</u>	<u>Change</u>
Magento 2.x and Up	1. Reindexing after importing products is no longer required.
Magento 1.7.0 and Up	1. Some fields previously found in the General Tab have been moved to the Description Tab. 2. Under the Prices Tab, the cost field is no longer used and/or will not appear.

6)Appendix 2 – IMan Jobs

6.1) Download Orders and Update Magento



1. Magento Reader – Pulls new orders from Magento.
2. Map Transform – Adds Fields
 - a. CustomerAccountSet. Required for Order import.
 - b. OrderNumber. For capturing write-back data for generated accounting system order number.

- c. WarningMessage. For capturing write-back data for any warnings from import.
 - d. ErrorMessage. For capturing any write-back data for any errors from import.
- 3. Sage 300 Connector – Attempts an Order Import into the accounting system.
- 4. Filter – Filter out any orders which incurred an error.
- 5. Email Task – Compose email for orders which did not import correctly due to an error.
- 6. Filter – Filter out any orders which were successfully imported.
- 7. Email Task – Compose email for orders which were successfully imported.
- 8. Map Transform – Add field for new Order Status in Magento for downloaded orders.
- 9. Magento Connector – Update Order Status in Magento.

6.2) Update Orders

Shipmented, voided,...

6.3) Upload Inventory

6.4) Adding & Updating Customers