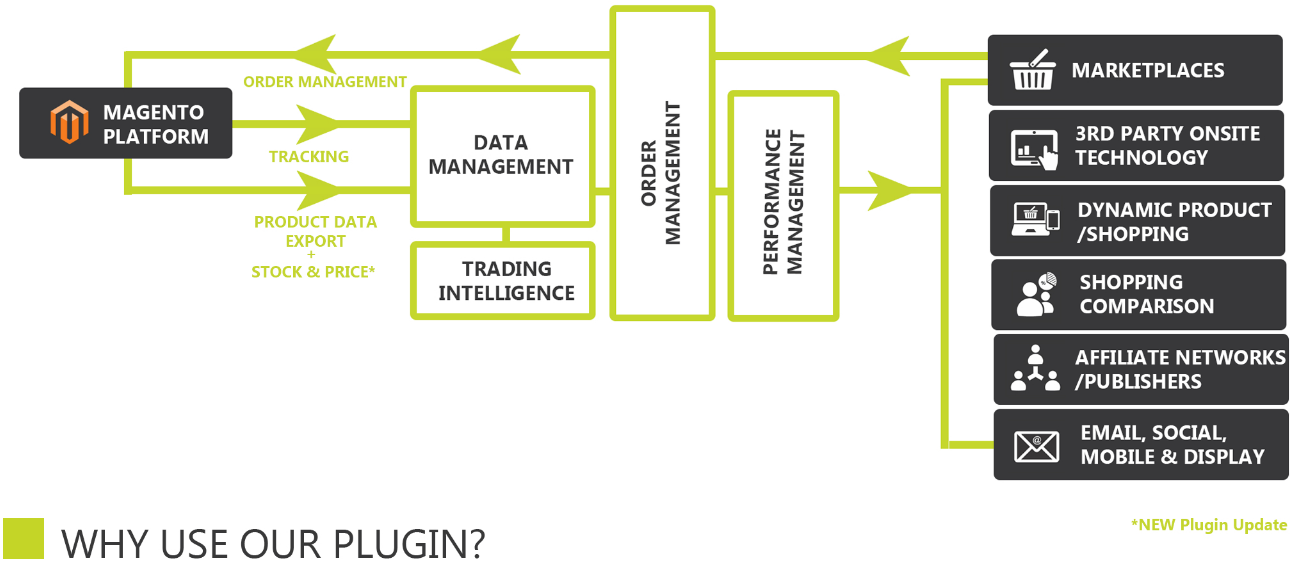
MAGENTO PLUGIN
User installation & setup guide

IntelligentReach

Our platform ingests your products, running automatic content analysis and optimisation, making high quality data available for any of our 1000+ digital commerce channels at the click of a button. Our IR module integrates Magento with the Intelligent Reach platform, enabling true product-level marketing.

HOW IT WORKS



WHY USE OUR PLUGIN?

Automate the synchronisation of your product data. This ensures data that goes to your marketing partners is as accurate as possible by including prices and stock levels for all your products to the Intelligent Reach platform. Products that are out of stock are not advertised and the product prices match your site.

Instantly deploy all of the IR tracking tags. This allows the IR platform to see all the performance metrics required to power product level performance optimisation to deliver more profitable sales revenue for your business.

Enables automatic synchronisation of order status with the IR platform which allows you to manage orders placed on multiple marketplaces across the globe seamlessly from your main Magento order screen.

COMPATIBILITY

Latest Version	Stability	Compatibility
1.0.37	**Stable**	**1.5, 1.6, 1.6.1, 1.6.2.0, 1.7, 1.8, 1.8.1, 1.9, 1.9.1, 1.13, 1.13.1, 1.14, 1.14.1**

PRODUCT BUILD

The Magento plugin exports product data directly from the site database, which allows merchants to have control over the data that will be exported. This avoids potential discrepancies with the data caused by load limitations. It delivers a much quicker turn around time for refreshing the data in product feeds.

The IR plugin currently provides 2 options for importing data from your site database:

IR IMPORTER

Product data is fetched from the site database by the IR platform through the IR Importer, scheduled according to client's requirements.

CRON JOB

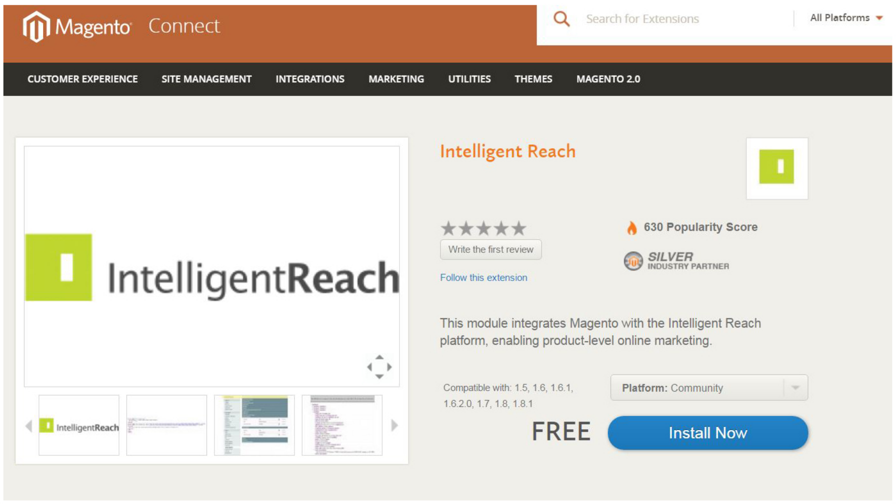
***Recommended for large product catalogues**

A cron job is scheduled by merchant which provides control over scheduled times for more resource friendly feed builds along with contingency options.

INSTALLATION

Firstly you will need to get the key for the IntelligentReach extension. You can do this by going to the IntelligentReach extension landing page.

You will be directed to the homepage of the extension, ensure that you are logged in and then click “Install Now”. Agree to the extension license agreement and click “Get Extension Key”. The key will now be shown, copy the whole address.



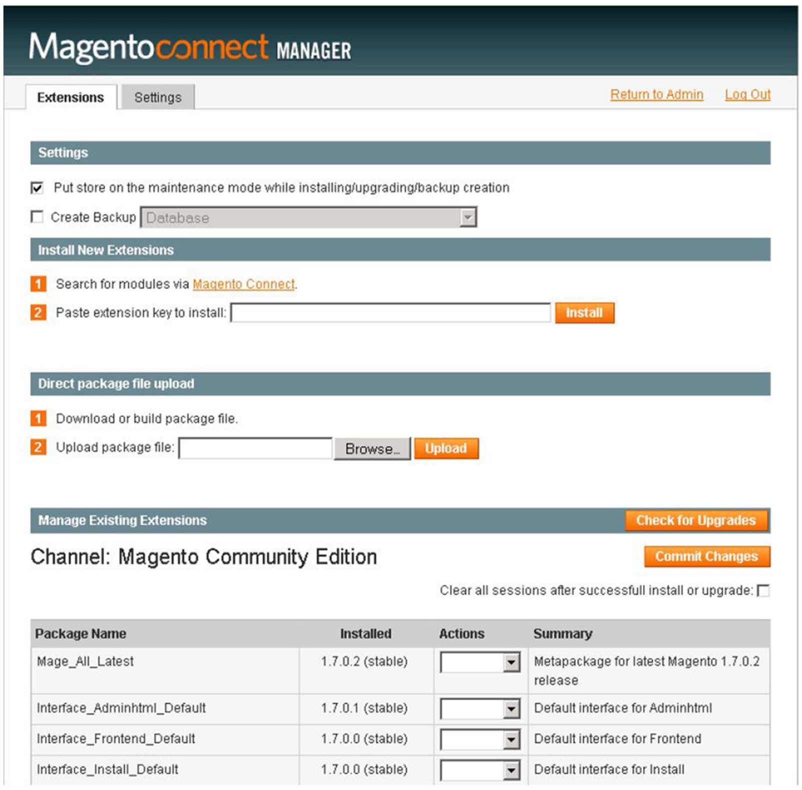
Intelligent Reach

This module integrates Magento with the Intelligent Reach platform, enabling product-level online marketing.

Compatible with: 1.5, 1.6, 1.6.1, 1.6.2.0, 1.7, 1.8, 1.8.1

Platform: Community

Then go to Magento Connect Manager, you will need to be logged into your Magento installation with the admin account to access this page.



Settings

☑ Put store on the maintenance mode while installing/upgrading/backup creation

☐ Create Backup Database

Install New Extensions

1 Search for modules via Magento Connect.

2 Paste extension key to install: Install

Direct package file upload

1 Download or build package file.

2 Upload package file: Browse... Upload

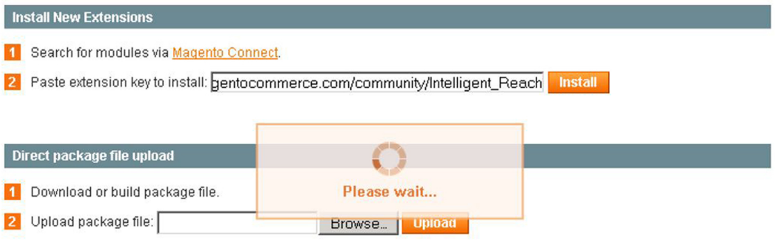
Manage Existing Extensions Check for Upgrades

Channel: Magento Community Edition Commit Changes

Clear all sessions after successfull install or upgrade: ☐

Package Name	Installed	Actions	Summary
Mage_All_Latest	1.7.0.2 (stable)		Metapackage for latest Magento 1.7.0.2 release
Interface_Adminhtml_Default	1.7.0.1 (stable)		Default interface for Adminhtml
Interface_Frontend_Default	1.7.0.0 (stable)		Default interface for Frontend
Interface_Install_Default	1.7.0.0 (stable)		Default interface for Install

In the "Install New Extensions" section paste the key into the field and click install as shown below.



Install New Extensions

1 Search for modules via Magento Connect.

2 Paste extension key to install: gentocommerce.com/community/Intelligent_Reach Install

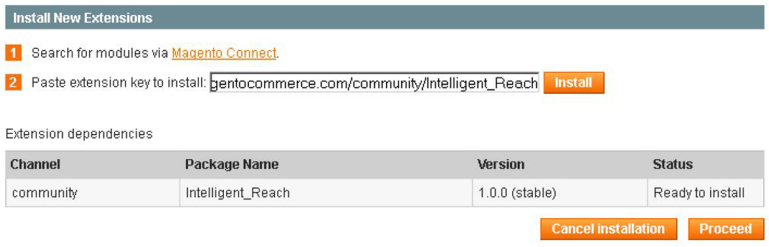
Direct package file upload

1 Download or build package file.

2 Upload package file: Browse... Upload

Please wait...

Once you click install a table should appear with the options to "Cancel Installation" or "Proceed". If you want to proceed then click the button and the installation will begin.



Install New Extensions

1 Search for modules via Magento Connect.

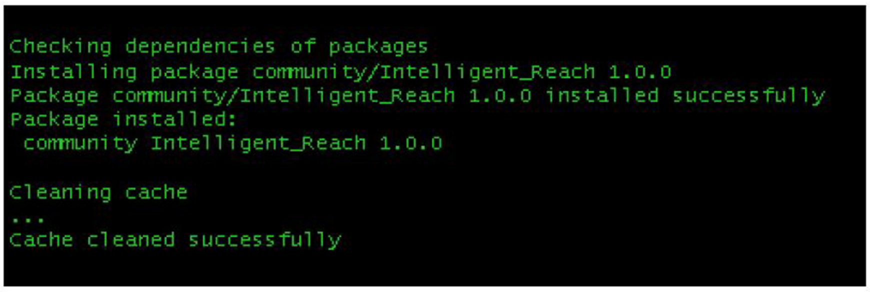
2 Paste extension key to install: gentocommerce.com/community/Intelligent_Reach Install

Extension dependencies

Channel	Package Name	Version	Status
community	Intelligent_Reach	1.0.0 (stable)	Ready to install

Cancel installation Proceed

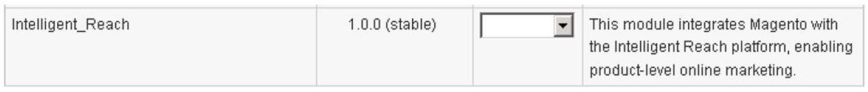
The output screen will display if the package was successfully installed and if it was not then the errors that occurred during installation will be shown.



```
Checking dependencies of packages
Installing package community/Intelligent_Reach 1.0.0
Package community/Intelligent_Reach 1.0.0 installed successfully
Package installed:
 community Intelligent_Reach 1.0.0

Cleaning cache
...
Cache cleaned successfully
```

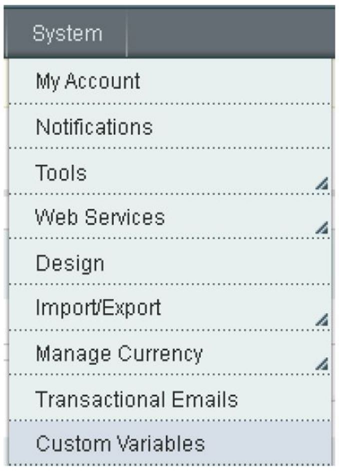
Once the output screen displays the "cache cleaned successfully" line, refresh the page and you should see the row below added to the table of existing extensions.



Intelligent_Reach	1.0.0 (stable)		This module integrates Magento with the Intelligent Reach platform, enabling product-level online marketing.

SETUP

Now that installation is complete the only thing left to do is add your company Id to the Magento system, this Id will then be picked up by the extension.

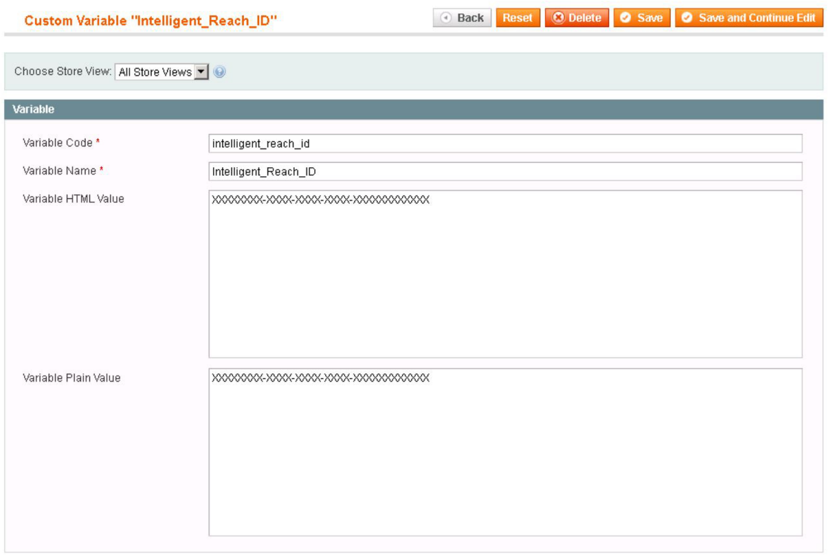


First go to the admin panel of your Magento installation; on the System tab in the main navigation bar, click on the menu item "Custom Variables".

Then on the Custom Variables landing page click the "Add New Variable" button.

Fill out the variable form that is displayed ensuring that the Variable Code has the value "intelligent_reach_id" (case sensitive), if this is entered incorrectly then the extension will not be able to pick up your Id. The name can be whatever you like but we recommend that you simply name it "Intelligent_Reach_ID".

In the two text areas that follow enter your company Id with the hyphens included. Finally click the “Save” button.

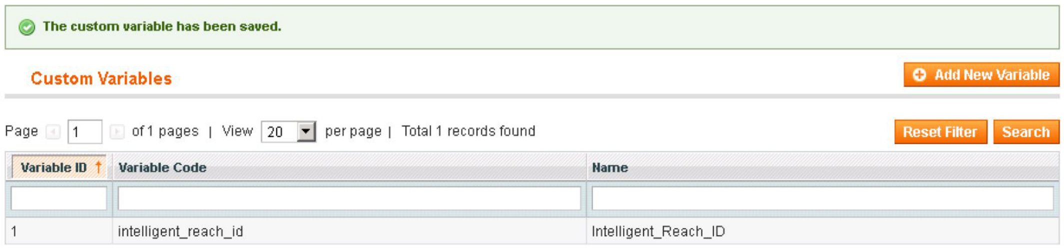

Custom Variable "Intelligent_Reach_ID"

Choose Store View: All Store Views

Variable

Variable Code *	intelligent_reach_id
Variable Name *	Intelligent_Reach_ID
Variable HTML Value	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
Variable Plain Value	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX

Once you are redirected to the Custom Variables landing page and you see the success message then you are all done, the extension has been successfully installed and setup for use with your Magento store.


The custom variable has been saved.

Custom Variables

Page 1 of 1 pages | View 20 per page | Total 1 records found

Variable ID	Variable Code	Name
1	intelligent_reach_id	Intelligent_Reach_ID

CATEGORY EXCLUSIONS

The category exclusions custom variable will allow you to exclude certain product category paths that you do not want to be included in the feed, for example if the category “Sale” is excluded then any category path that contains the category “Sale” will be excluded from the feed.

If you are adding category exclusions, then you will need to create another custom variable but this time enter the Variable Code “intelligent_reach_category_exclusions” (case sensitive), and the name again can be whatever you prefer but we think “Intelligent_Reach_Category_Exclusions” should suffice.

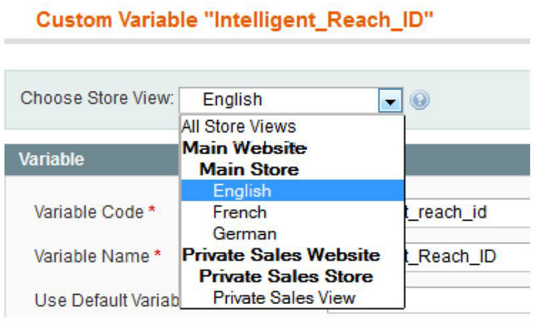
In the two text areas enter every Category name you wish to exclude, each Category should be separated by a pipeline symbol (|) with no space between them e.g. Sale|On Sale. Lastly click the “Save” button to persist the category exclusion values to your database. If you have different category exclusions for multiple stores then you will need to follow the instructions in the following section to setup category exclusions for each store.

FOR MULTIPLE STORE VIEWS

If you are using multiple stores then create the Custom Variable and fill out the fields as instructed, the values inputted should be the Intelligent Reach Unique ID for your Default store; so if the default store was English then you would input the Unique ID for that.

Now "click Save and Continue Edit".

The variable should have been saved and a drop down should appear above which allows you to choose a store view.



Custom Variable "Intelligent_Reach_ID"

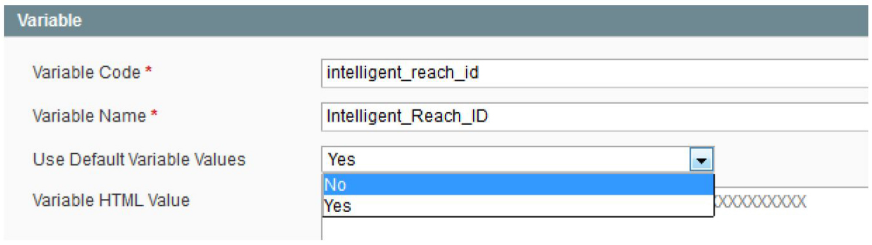
Choose Store View: English

All Store Views
Main Website
Main Store
English
French
German
Private Sales Website
Private Sales Store
Private Sales View

Variable

Variable Code *	_reach_id
Variable Name *	_Reach_ID
Use Default Variab	

Select the next store view and select "No" on the drop down labelled "Use Default Variable Values".



Variable

Variable Code *	intelligent_reach_id
Variable Name *	Intelligent_Reach_ID
Use Default Variable Values	Yes (No / Yes)
Variable HTML Value	XXXXXXXXXX

Now enter the Company ID that is linked to that Store View.

Please note that you will need to repeat these steps for each Store view; otherwise the tracking will fail.

If you have added category exclusions and if these exclusions vary for each store then you will also need to perform the actions mentioned in this section and set the exclusion values for each store, so that each store uses the correct exclusions rather than the exclusions that were set in the Default Variable.

CRON COMPATIBLE SCRIPTS

If you would like to use CRON jobs to produce your product and quantity feeds, there are feed generation scripts that can be run to do this, these are:

ircronscripts/intelligentreach_integration_cron.php – this is the first product feed generation script, the way it pulls the product data is different to the second; the product collection with all of the product attributes are loaded first and then written to the file.

ircronscripts/intelligentreach_integration_cron_pre.php – this is the second product feed generation script, this loads each product and all its attributes individually and then writes them to the file.

ircronscripts/intelligentreach_integration_cron_qty_price.php – this is the quantity and price feed generation script, it pulls each product quantity and price and writes them to the file.

The setup of a CRON job is beyond the scope of this guide. Please refer to your Operating Systems user guide or other resources for help on how to do this.

When setting up the CRON jobs please set the product feed script to be run once every day, this will cause a little bit of load on your servers so please set it to run at a time when your site is least busy. The quantity and price script will not require as much resources and so can be updated often - please set this to be run at least every 15 minutes.

During the feed review stage, we will need to see the data returned from both product feed generation scripts. For this, please temporarily change the mame of the file generated in each script as they will overwrite each other if left with the default value.

You can do this by changing the value of the variable $_fileName. Once our integration team has finished the review, we will inform you of which product feed script to use going forward, and you can revert the change.

SCRIPT SETTINGS

There are settings within the script that can be changed to suit your particular needs. By default you do not need to change any of these settings. Typically, after reviewing the feed that is produced, we will inform you of any discrepancies in the data and the changes you will need to make to the settings to resolve them.

A list of the settings that can be changed and some brief information on what they do:

$_outputDirectory – defaulted to "output", this is the directory where the generated feed will be output. If you would like the generated feed file output to another directory then you can change this as required.

$_fileName – defaulted to "Feed" in the main product scripts and "Feed_Quantity_And_Price" in the _qty script. If you need to change the name of the feed file to something else then this is the setting you will need to edit.

$_amountOfProductsPerPage – defaulted to 1000, this is used when pulling the products from your database, we use paging to decrease the load. If you increase this number, then more products will be pulled per page, which will decrease the time taken to produce the feed, but will also increase the resources required. If you decrease this number, fewer products will be pulled per page; increasing the time taken to produce the feed but decreasing the resources required.

$_gzipFile – defaulted to true, this will compress the feed file to a GZIP file, and leave the file as it is if set to false.

$_includeAllParentFields – defaulted to false in the main product scripts and not available in the _qty script. If set to true, will write all the parent product fields for each product to the feed file, and will not if left as false.

$_stripInvalidChars – defaulted to false in the main product scripts and not available in the _qty script. If your feed is returning invalid XML characters, then setting this to true will remove these so that a valid XML feed file is produced.

$_convertNumberToWord – defaulted to false in the main product scripts and not available in the _qty script. Used only if some of your product attribute fields start with a number, as this is not valid XML and will break the feed. If set to true, this will convert the number to its numerical equivalent.

TESTING

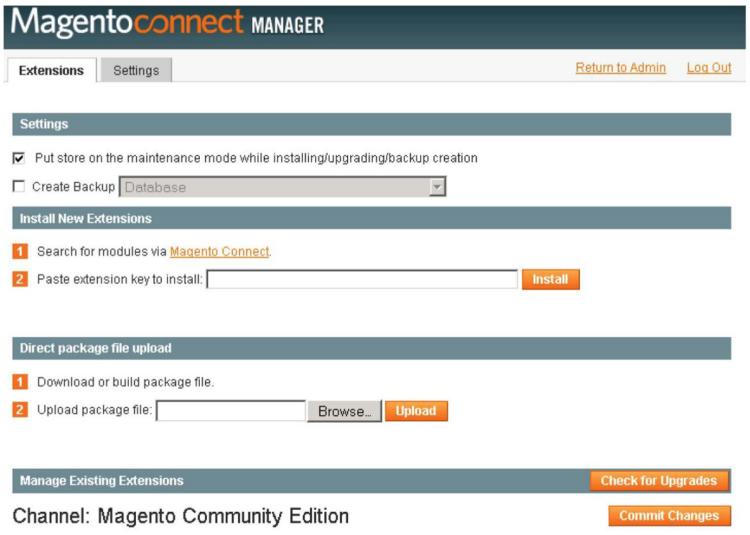
To test if the extension has added the tracking code you could view the source of each page and just below the starting body tags you should see the IntelligentReach tracking code as shown below, please bear in mind that the tracking code may vary for each page.

```
<body class=" catalog-product-view catalog-product-view product-sony-vaio-vgn-txn27n-b-11-1-notebook-pc">
<script type="text/javascript">
  istCompanyId = "XXXXXXXX-XXXX-XXXX-XXXXXXXXXXXXXX";
  istItem = "27";
</script>
<script type="text/javascript" src="//www.ist-track.com/ContainerItemJavaScript.ashx?id=XXXXXXXX-XXXX-XXXX-XXXXXXXXXXXXXX"></script>
<script type="text/javascript" src="//www.ist-track.com/ContainerJavaScript.ashx?id=XXXXXXXX-XXXX-XXXX-XXXXXXXXXXXXXX"></script>
<noscript>
  <img src="//www.ist-track.com/ProcessClickImage.ashx" alt="" width="1" height="1"/>
```

If you don't then this could be a caching issue, to resolve this you should flush the cache of the layout, the exact steps on how to do this is beyond the scope of this guide however there are plenty of resources available for this online.

UPDATING OR UNISTALLING

There are two different ways to update your existing version of the Intelligent Reach extension. The first is to use the built in update function that Magento has; please note that this may not work/exist in earlier versions of the Magento platform. If it does exist, then you can try this method and if unsuccessful move on to the other. Locate the "Check for Upgrades" button in the connect manager; this should be at the top of existing extensions.



Settings

☑ Put store on the maintenance mode while installing/upgrading/backup creation

☐ Create Backup Database

Install New Extensions

1 Search for modules via Magento Connect.

2 Paste extension key to install: Install

Direct package file upload

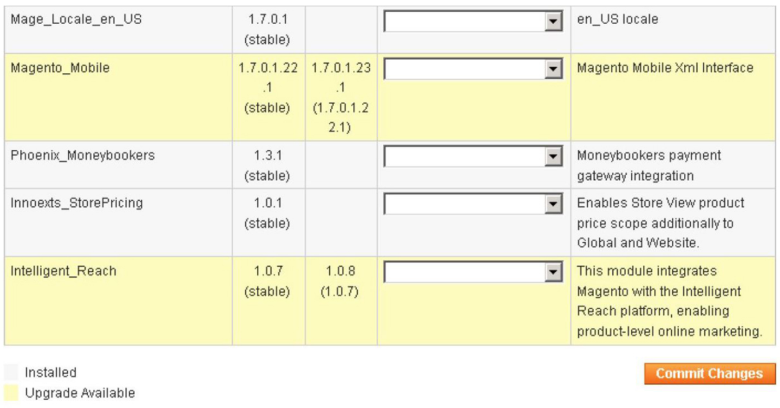
1 Download or build package file.

2 Upload package file: Browse... Upload

Manage Existing Extensions Check for Upgrades

Channel: Magento Community Edition Commit Changes

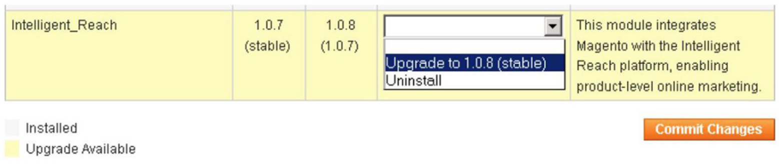
If the Updating function works, then you should see all the available updates for your existing extensions; if it does not show this, then move on to the next method.



Mage_Locale_en_US	1.7.0.1 (stable)			en_US locale
Magento_Mobile	1.7.0.1.22 .1 (stable)	1.7.0.1.23 .1 (1.7.0.1.2 2.1)		Magento Mobile Xml Interface
Phoenix_Moneybookers	1.3.1 (stable)			Moneybookers payment gateway integration
Innoexts_StorePricing	1.0.1 (stable)			Enables Store View product price scope additionally to Global and Website.
Intelligent_Reach	1.0.7 (stable)	1.0.8 (1.0.7)		This module integrates Magento with the Intelligent Reach platform, enabling product-level online marketing.

⬜	Installed
🟨	Upgrade Available

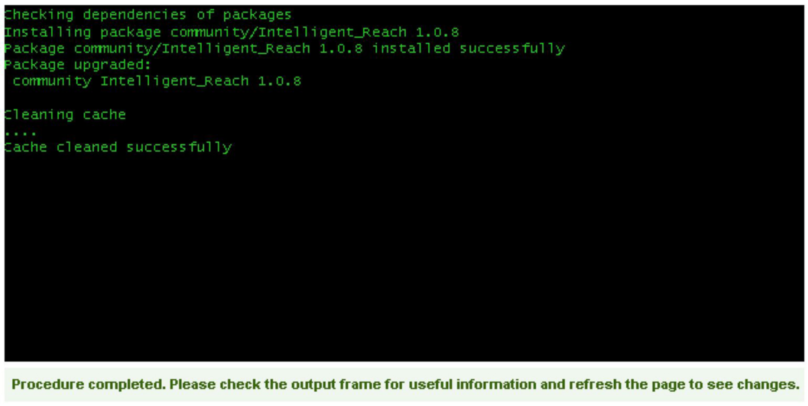
Now use the drop down box for the row representing the Intelligent Reach extension, select first option which should read "Upgrade to x.x.x.", once selected click the "Commit Changes" button.



Intelligent_Reach	1.0.7 (stable)	1.0.8 (1.0.7)	Upgrade to 1.0.8 (stable) / Uninstall	This module integrates Magento with the Intelligent Reach platform, enabling product-level online marketing.

⬜	Installed
🟨	Upgrade Available

The update should run by itself, and should return a success message. If this method fails here for some reason, you can try the other method.

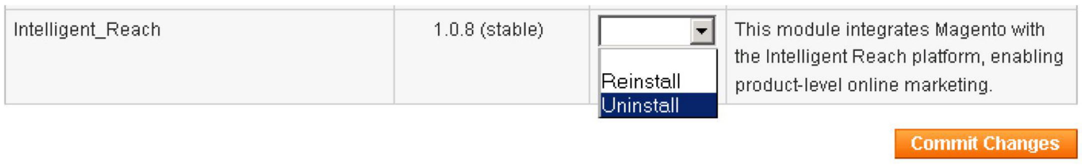


```
Checking dependencies of packages
Installing package community/Intelligent_Reach 1.0.8
Package community/Intelligent_Reach 1.0.8 installed successfully
Package upgraded:
 community Intelligent_Reach 1.0.8

Cleaning cache
....
Cache cleaned successfully
```

Procedure completed. Please check the output frame for useful information and refresh the page to see changes.

For this second method, you simply uninstall the extension and reinstall it again. First step is to select the "Uninstall" option and click Commit Changes. This will uninstall extension and produce a success message.



Intelligent_Reach	1.0.8 (stable)	Reinstall / Uninstall	This module integrates Magento with the Intelligent Reach platform, enabling product-level online marketing.

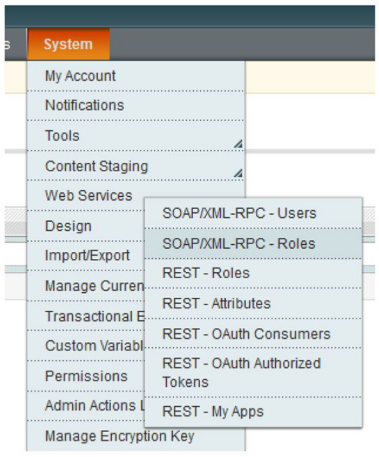
Once uninstalled repeat the installation procedure detailed in this guide, you do not need to reconfigure the custom variables; as this will not be removed automatically.

CREATING AN API USER ACCOUNT

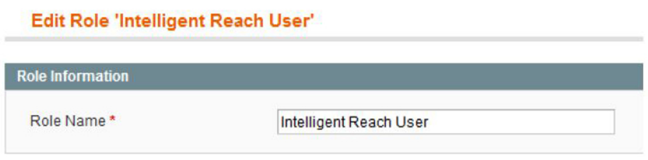
This section following section will detail how you can create an API User account on your system for IntelligentReach to use for Marketplace Integration.

If you are not currently integrating to marketplaces but would like to please contact your account manager before setting up the API User account.

In order to create an API User account you first need to create a role which will be used to assign the access permissions the account will have. In your Magento admin platform go to "System" -> "Web Services" -> "SOAP/XML-RPC – Roles" as shown.



You will then need to enter a Role name, this can be anything you want as it will assist you in remembering what the role is being used for.

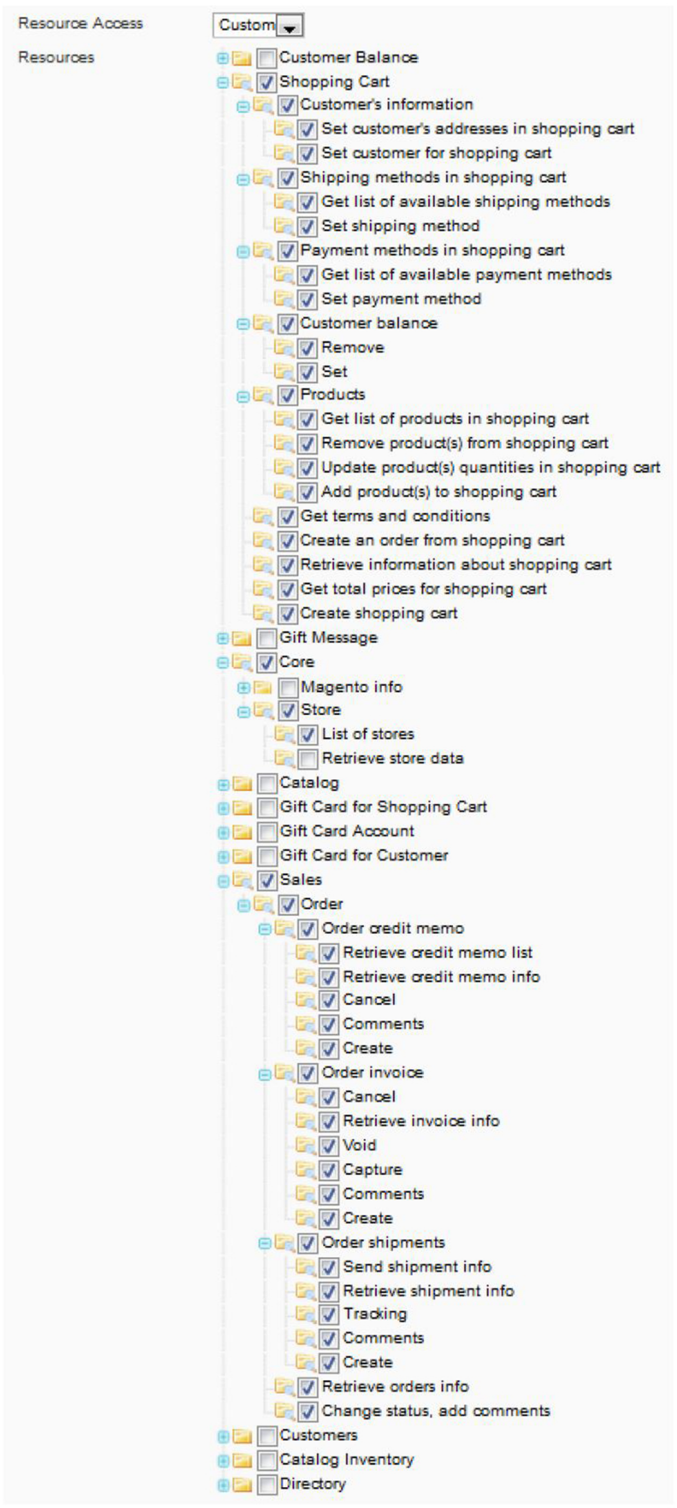


Edit Role 'Intelligent Reach User'

Role Information

Role Name * Intelligent Reach User

Then you will then need to set the Role Resources access rights. We currently only need the following resources, if we require any further access we will communicate this to you at a later date.

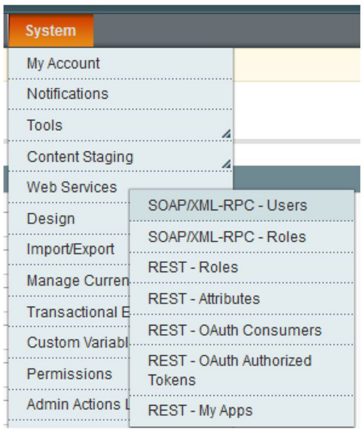


Resource Access: Custom

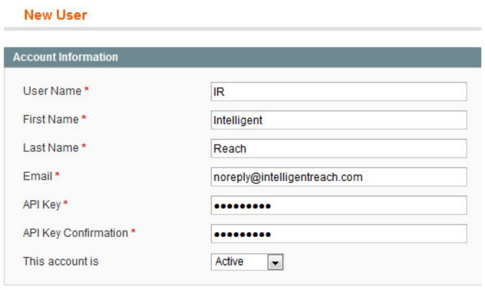
Resources

- [ ] Customer Balance
- [✓] Shopping Cart
  - [✓] Customer's information
    - [✓] Set customer's addresses in shopping cart
    - [✓] Set customer for shopping cart
  - [✓] Shipping methods in shopping cart
    - [✓] Get list of available shipping methods
    - [✓] Set shipping method
  - [✓] Payment methods in shopping cart
    - [✓] Get list of available payment methods
    - [✓] Set payment method
  - [✓] Customer balance
    - [✓] Remove
    - [✓] Set
  - [✓] Products
    - [✓] Get list of products in shopping cart
    - [✓] Remove product(s) from shopping cart
    - [✓] Update product(s) quantities in shopping cart
    - [✓] Add product(s) to shopping cart
  - [✓] Get terms and conditions
  - [✓] Create an order from shopping cart
  - [✓] Retrieve information about shopping cart
  - [✓] Get total prices for shopping cart
  - [✓] Create shopping cart
- [ ] Gift Message
- [✓] Core
  - [ ] Magento info
  - [✓] Store
    - [✓] List of stores
    - [ ] Retrieve store data
- [ ] Catalog
- [ ] Gift Card for Shopping Cart
- [ ] Gift Card Account
- [ ] Gift Card for Customer
- [✓] Sales
  - [✓] Order
    - [✓] Order credit memo
      - [✓] Retrieve credit memo list
      - [✓] Retrieve credit memo info
      - [✓] Cancel
      - [✓] Comments
      - [✓] Create
    - [✓] Order invoice
      - [✓] Cancel
      - [✓] Retrieve invoice info
      - [✓] Void
      - [✓] Capture
      - [✓] Comments
      - [✓] Create
    - [✓] Order shipments
      - [✓] Send shipment info
      - [✓] Retrieve shipment info
      - [✓] Tracking
      - [✓] Comments
      - [✓] Create
    - [✓] Retrieve orders info
    - [✓] Change status, add comments
- [ ] Customers
- [ ] Catalog Inventory
- [ ] Directory

After setting the Role Resources Access, you can save this role. Now that the role has been successfully saved, you can now create the account by going to "System" -> "Web Services" -> "SOAP/XML-RPC – Users" as shown on the next page.



Then enter the details required for the Account, you will need to make a note of the User name and API Key as we will need them when connecting to your platform.

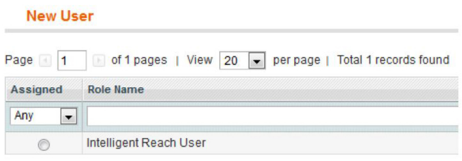


New User

Account Information

User Name *	IR
First Name *	Intelligent
Last Name *	Reach
Email *	noreply@intelligentreach.com
API Key *	•••••••••
API Key Confirmation *	•••••••••
This account is	Active

Finally assign the user to the Role you created earlier and click save. The API User account has now been created.



New User

Page 1 of 1 pages | View 20 per page | Total 1 records found

Assigned	Role Name
Any	
○	Intelligent Reach User

NEXT STEPS

QA - IR & Client Quality Assurance Review
IR Data Quality team will QA the newly built product exports before pushing this into production, test orders will be placed to test the tracking and the order status update functionality.

Onboarding Process - Getting you Live..
The new product exports will become the main source data of the IR platform. Once feeds are pushed to marketplace partners via the IR platform API integrations, orders will flow through from the marketplaces via the IR platform to your Magento order screen.

Post Project Review – Review of the merchant following launch
IR confirms all partner feeds are working as expected following the update and that tracking and orders are being received accurately.