



# PRODUCT DATA EXPORT

## START GUIDE FOR PUBLISHERS

ZANOX MARKETPLACE NEW DATA FEED PLATFORM

Frank Hollander, Amsterdam, August 2013

---

## TABLE OF CONTENTS

---

	Introduction .....	3
<b>1</b>	<b><i>Marketplace data feed platform</i></b> .....	<b>4</b>
	1.1 Marketplace interface .....	4
	1.2 Using the data feed interface .....	5
	1.3 Stream download options .....	6
	1.4 Selection streams .....	7
<b>2</b>	<b><i>Provision of product data</i></b> .....	<b>8</b>
	2.1 CSV .....	8
	2.2 XML .....	8
	2.3 Special purpose formats .....	9
	2.4 Encoding .....	9
	2.5 Download process and policies .....	9
	2.6 Handling big streams with the offset function .....	10
	2.7 Using GPP sub-id tracking with data feeds .....	10
	2.8 DF meta information API .....	11
<b>3</b>	<b><i>Data feed usage</i></b> .....	<b>13</b>
	3.1 Essentials of the affiliate DF web concept .....	13
	3.2 Mapping techniques .....	13
	3.2.1 Category mapping .....	14
	3.3 Affiliate data feed software .....	14
	3.4 Schedule automated data import .....	15
	3.4.1 Setting up cron jobs .....	16
	3.4.2 Using Linux crontab .....	18
	3.5 The data feed parser .....	19
	3.5.1 Example import code .....	19
<b>4</b>	<b><i>Zanox M4N DF field definitions</i></b> .....	<b>24</b>
	4.1 Basic fields (and standard shopping segment fields) .....	24
	4.2 Segment fields .....	25
	4.2.1 Shopping (fashion) .....	25
	4.2.2 Travel .....	25
	4.2.3 Mobile .....	26
	4.2.4 Jobs .....	27
	4.2.5 Tickets (airline) .....	28
	4.2.6 Education .....	28
	4.2.7 Daydeal (extra fields) .....	29

---

## PRODUCT DATA EXPORT

---

### Introduction

Apart from banners and text links that point to a homepage or product page, there are other important meanings of advertising. This type of advertising permits publishers to set up a product comparison and get a sites overview. These websites offer their visitors a search function and the possibility of comparing products with regard to price, service, delivery times, and experiences of other customers. These affiliate advertising products are called 'data feeds' or 'product feeds' and can be transferred from the Zanox Marketplace for a large number of our advertisers. This guide will explain the use of Zanox feeds and the Zanox Marketplace new platform DF interface.

### Technical definition of data files

A data file is an automatically generated list that shows all products in a certain merchant's web shop. For each product, a number of data needs to be supplied. If correctly prepared, data files can be regularly read by publishers sites, and used to maintain their own product databases.

Wikipedia describes a 'data feed' as:

*"A data feed is a mechanism for users to receive updated data from data sources. Data feeds usually requires structured data. At the present time unstructured data, e.g. HTML pages, dominate the Web. As a result, data feeds have the potential to make a bigger impact on the Web in the future."*

### History of the Zanox 'new' data feed platform

More than 10 years ago the first elements of the M4N data feed system were born and constructed by the technical originators Klaas Joosten and Willem Cazander. In that time data feeds and affiliate marketing were not integrated as well with each other as they are nowadays. For example the system did not have that many download options like different formats and the ability to include tracking parameters within the deep-links needed to be implemented by the publishers themselves.

Through the years the data feed system evaluated in a spectacular way where more and more features have been added to make 'life' easier for the users. Most contributions and improvements are due to Jaap de Jong who spent more than 8 years working on the system to get it where it is now. Since the acquisition of M4N by Zanox, one of the many initiatives was the integration of the M4N data feed system into the Zanox marketplace.

## 1 Marketplace data feed platform

The Zanox technical department have been working hard behind the scenes to offer the best of both networks to publishers. This chapter explains about the use of the new DF interface and where publishers can find the product feed download locations on the Zanox Marketplace platform.



Figure 1: Go to "Programs" to apply and use data feeds.

### 1.1 Marketplace interface

Before you can use the data feeds from a specific advertiser you need to apply to that program with the desired ad-space connection. You can apply to programs in Marketplace via "Programs" > "Apply to Programs". Click on the "+" symbol next to the desired program to apply to the program.

Program	Conversion Rate	Confirmation Rate (S/L)	Approval Time	Payment Time	EPC	AdRank	Product Data	Launch Date	Actions
La Redoute FR	1.20%	91.1% / 100%	19 days	54 days	0.05 EUR	9	Yes	14/03/08	
Groupon FR	2.80%	95.8% / 0%	31 days	34 days	0.08 EUR	8.6	Yes	12/01/11	

Figure 2: Overview Marketplace "Not Applied" screen.

Program	Conversion Rate	Confirmation Rate (S/L)	Approval Time	Payment Time	EPC	AdRank	Product Data	Launch Date	Actions
Zalando NL	5.10%	72.4% / 0%	48 days	48 days	0.24 EUR	8.6	Yes (01/07/13)	11/09/10	
Omoda NL	5.10%	67% / 0%	23 days	23 days	0.37 EUR	5.5	Yes (01/07/13)	08/07/08	

Figure 3: Overview Marketplace "Accepted" screen.

Once you are accepted for the advertisers program you have access to the data feed download page. You can reach this page by clicking on the orange "yes" link in the "Product Data" column.

## Product feeds

Instead of the default zanox data feed UI you are currently viewing the new data feed interface. The new data feed application is based on M4N software and is available for all zanox Benelux publishers. The tool offers more functionality and flexibility in export formats, search and filtering.

If you have any questions or suggestions concerning this page, please contact [feeds.nl@zanox.com](mailto:feeds.nl@zanox.com). If you do not want to use this tool, you can go to the [default zanox data feed page](#).

ID	Name	Segment	Records	Last checked	Last parsed	Program ID	Check frequency	Currency		
218106	zalando.nl	Shopping	125223	11-Jul-2013 10:21:31	11-Jul-2013 10:24:32	8955	custom	EUR	Selection	
310106	zalando.nl schoenen	Shopping	38796	11-Jul-2013 10:25:05	11-Jul-2013 10:30:59	8955	custom	EUR	Selection	

Figure 4: Data feed download page overview screen.

### Important note:

At this moment the "new data feed export interface" is only reachable through the Zanox Marketplace by Benelux publishers. If your account is not registered as Benelux but for example German, France or UK then you will be shown the old data feed interface.

As an alternative there is a 'backdoor' possibility to reach and download streams from the new DF interface with a non-Benelux account. We have built the alternative page in the Zanox Toolbox:

<http://toolbox.zanox.com/benelux-datafeeds/>


## 1.2 Using the data feed interface

On the data feed download page you get an overview of all the available streams for the selected advertiser and ad-space. In the stream table you can find meta information about the feeds and two download options.

Name	Comment	Example
ID	Stream identification	218106
Name	Stream name (sub name)	"zalando.nl shoes"
Segment	Publisher segment	Shopping, Travel, GSM, etc.
Records	Stream record count	38796
Last checked	Last check advertiser source	1-Dec-1978 00:00:00
Last parsed	Last parsing advertiser source	1-Dec-1978 00:00:00
Program ID	Zanox advertiser identification	8955
Check frequency	Number of daily check attempts	daily, 2hourly, custom, etc.
Currency	Currency format of prices	EUR
"Selection"	Selections download links	Selection
"Stream download"	Stream download options	

Table 1: Stream meta definitions.

## 1.3 Stream download options

The 'most' important part of the DF download pages are the stream download options, this is the screen where you can 'build' your desired download URL's that fits your import system or other usage. You can reach this popup screen by clicking on the green  icon in the "data feed overview table".

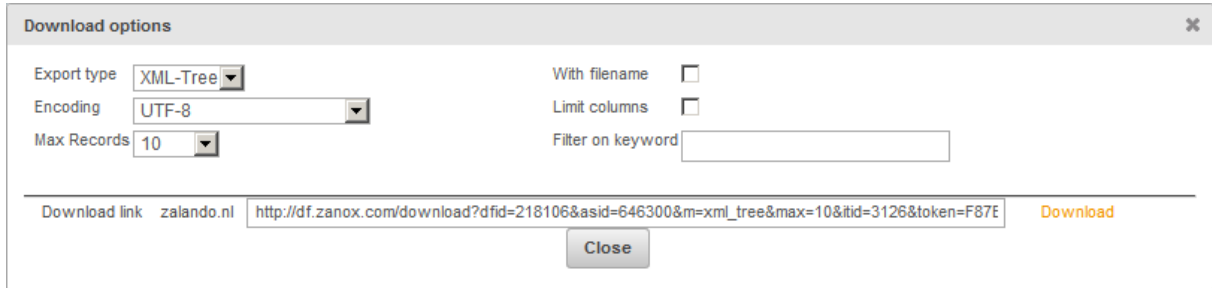


Figure 5: Data feed download options popup screen.

Option name	Comment	Example
Export type	Data download format	CSV, XML-Tree, XML
Encoding	Data encoding type	UTF-8, ISO-8859, etc.
CSV separator	Column-delimiter (CSV only)	Tab (\t), Pipe ( ), semicolon(;), etc.
CSV record quote	Field-limiter (CSV only)	"...", '...', or none
Max. Records	Number of records when truncate stream	10, 50, 1000, 50000, etc.
With filename	Add file extension to the stream	../stream-218106-646300.xml
Limit columns	Let you choose only the desired columns	url, offerid, title, price, etc.
Filter on keyword	Filters on title and description. Not on categories so the use is limited.	"shirt", "pants", "shoes", etc.
Download link	URL to export the data	http://df.zanox.com/download/....

Table 2: The download screen consist a wide scale of options which are explained here.

## 1.4 Selection streams

Instead of downloading complete stream data it is also possible to export only a specific part of a stream based on categories, vendors or sometimes gender. This can be done by following the orange "Selection" link on the data feed download page (see figure 4).

ID	Records	Mapping Catalog
33574025	3	Geslacht
6013	96	Subcategorie
6012	12	Categorie
6015	513	Third Category

Figure 6: Selections overview screen after clicking the "Selection" link.

Hide empty selection

**Search options**

Search




ID	Data	Records	
29492	Cosmetica	3526	
29493	Dames	4964	

Figure 7: Selection download overview, after selected the desired "selection" group.

On the screen shown in *figure 7* you can download the streams (feeds) per category. By clicking on the green  icon you reach the same "stream download options" as described in 1.3.

**Important note:**

The selection downloads can be very useful when you only need to download specific "category" data but this feature has one disadvantage that you should be aware of. Every used category- (or other field) is confederated to an automatic generated ID which is used in the download URL to export the matching data. This ID is identified as 'miid' and exist in the URL for example as "&miid=29492". Once a category name is changed by the advertiser, for example "shirt" into "Shirts", the 'miid' also changes after the next parsing, thereby the current download location is empty. When this occurs you should get a new download URL for the specific stream and category from the data feed interface.

## 2 Provision of product data

Product data in general can be provided in many different formats. The Zanox Marketplace DF platform enables you to download our advertiser streams in the most popular and common data formats. This chapter explains you about our data formats and provision options.

### 2.1 CSV

CSV means 'character-separated values' which is a plain-text based (flat) file containing tabular product data separated by a delimiter. The first line in the file normally describes the column(s) meaningful name(s) known as header row.

#### Columns

Columns are separated from each other by a separator (column-delimiter), e.g. a semicolon “;” or TAB. If possible, select this separator in such a way, that it does match with the technical requirements of your data feed system. We recommend to use the TAB or pipe characters as column-delimiters.

#### Field-limiter

We recommend to enclose all fields with a field-limiter, e.g. a double quote: “.” or single quote: ‘.’. We standard enclose the text in the content fields with a double quote field-limiter. See figure 8.

```
"url" "title" "offerid" "image" "price" "category" "subcategory" "stock" "timetoship"  
"http://domain.nl" "Solid OAKES Longsleeves Groen" "SO422D02T-602" "http://i1.ztat.net/cata
```

Figure 8: CSV stream snipped. TAB separated with double quotes as field-limiter.

### 2.2 XML

XML means 'extensible markup language' and is more flexible than CSV, which format you use depends on your own preference and technical requirements. On the Zanox DF platform we provide two types of common XML formats: XML-Tree and XML, see figures 9 and 10 for an example.

XML can be displayed with XSLT, a language with an XML-based syntax that is used to transform XML documents into other XML documents, HTML, or other, unstructured formats such as plain text.

```
<?xml version="1.0" encoding="UTF-8"?>  
<m4n>  
  <data>  
    <record>  
      <url><![CDATA[http:// ad.zanox.com/]]></url>  
      <title><![CDATA[alterset curlstang 25 kg.]]></title>  
      <price><![CDATA[34.99]]></price>  
      <category><![CDATA[Dagaanbieding]]></category>  
      <subcategory><![CDATA[Halters & gewichten]]></subcategory>  
      <etc.><![CDATA[available]]></etc.>  
    </record>  
  </data>  
</m4n>
```

Figure 9: XML-Tree is the most common used short XML format.



```
<?xml version="1.0" encoding="UTF-8"?>
<m4n xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="datafeeds/ext/m4n-datafeeds.xsd">
  <data>
    <record>
      <column name="url"><![CDATA[http://ad.zanox.com/ column>
      <column name="title"><![CDATA[Halterset curlstang 25 kg.]]></column>
      <column name="price"><![CDATA[34.99]]></column>
      <column name="category"><![CDATA[Dagaanbieding]]></column>
      <column name="subcategory"><![CDATA[Halters & gewichten]]></column>
      <column name="etc."><![CDATA[available]]></column>
    </record>
  </data>
</m4n>
```

Figure 10: XML long format, less used than XML-Tree.

### XML data header

One of the advantages of XML over CSV format is that our XML files have a *dataheader* which provides meta data about the stream like *row count*, *last parse date*, *stream name*, etc. This can be useful information for automating DF import processes in complex data feed systems.

### Record Hash (XML)

Another advantage over CSV data is that our XML streams contain a *record hash* tag for each record. This is an automatically generated unique MD5 hash ID over all the available columns. Every single change in the data has the effect that this ID changes. If you store the hash ID for each record in your database you can perform a check whether records have to be removed and updated or not.

### CDATA tags

All our XML data is contained in CDATA tags because a content can contain special characters like HTML entities or HTML. Especially URL's containing query strings can cause XML parsing problems with your parser because the "&query=something" will be interpreted as an HTML entity rather than a part of the content.

## 2.3 Special purpose formats

Beside XML and CSV you will find two other download format selections in the DF interface, these are 'HTML' and 'Template'. HTML can be used to convey a list of clickable *title* links from a particular stream, this can be used to find deep links or test the DF click URL's. The *template* format is for consuetude and internal use only.

## 2.4 Encoding

Just like you have to select the right download format that suits your own data feed system, you also need to choose the right stream encoding. The encoding should match the one used in your database and code. In general UTF-8 is the current standard.

## 2.5 Download process and policies

The provided data streams are dynamic URL's that download the product data from our database. From this point there is one important rule that you are not allowed to use the streams directly on your webpage / HTML source, the data always needs to be stored on your own server into a local storage environment like a database or file system. With an automated task (cron job) you can periodically

download and store the data from our streams to keep your information up-to-date. For most advertisers an update frequency once a day (night) is enough, although some feeds have a higher update frequency or they need to be updated on specific times (for example day deal feeds).

More information about common used databases: <http://www.mysql.com/> or <http://www.postgresql.org/>

More information about cron jobs: <https://en.wikipedia.org/wiki/Cron> or see part 3.4 of this manual.



Figure 11: Download process affiliate data feed schematic.

## 2.6 Handling big streams with the offset function

In some situations our streams seem too big for your import script and at times errors occur. In this case you can use our offset parameter in the download URL and divide the dataset in as much multiple streams as you like. URL parameters: `&max=10&off=0`

Record count and Offset	URL parameter values
records 0 to 4999	<code>&amp;max=5000&amp;off=0</code>
records 5000 to 9999	<code>&amp;max=5000&amp;off=5000</code>
records 10000 to 14999	<code>&amp;max=5000&amp;off=10000</code>

Table 3: Example if you want max. 5000 records per stream.

## 2.7 Using GPP sub-id tracking with data feeds

GPP (General Purpose Tracking Parameters) is a sub-id tracking system and consists of a series of alpha-numeric parameters available for publishers. GPP is basically a container which publishers can use to place their own tracking information onto conversions. The GPP solution replaces the parameter `SIDE`, which was previously used to manage sub IDs. For more information about GPP, request our *GPP set-up guide for publishers*.

GPP parameters can also be used in data feed deep link URLs. Just like conventional deep links you have to add the parameters and the values yourself. But in the case of data feeds it can be about hundreds or thousands of links where you must add the extra tracking code so you have to automate this in the import process or embed in the output code. How to implement this technically and what kind of information to store is up to you or your programmer. For example you can add the same static information for all deep links, but you can also use placeholders and replace dynamic information on conversions.



Adding the GPP *zpar0* parameter in a data feed deep link URL with placeholder (XX) can be performed in two correct ways as shown in these examples:

Option 1 before the ULP variable:

`http://ad.zanox.com/ppc/?25726164C82068206&zpar6=DF_1&zpar0=XX&ULP=[[deeplink]]`

Option2 after the ULP variable:

`http://ad.zanox.com/ppc/?25726164C82068206&zpar6=DF_1&ULP=[[deeplink]]&zpar0=XX`

In the 2nd option be sure the ULP content is always surrounded by double brackets "[[...]]".

## 2.8 DF meta information API

It is possible to connect to our DF platform without using the user interface. This can be done by using our DF API (application programming interface) which replies an XML, CSV or JSON file with all data feed streams and meta data like stream ID, segment, record count, last parse date, etc.

Format	Request URL
XML	<code>http://df.zanox.com/api/xml/public/datafeeds/list/</code>
CSV	<code>http://df.zanox.com/api/csv/public/datafeeds/list/</code>
JSON	<code>http://df.zanox.com/api/json/public/datafeeds/list/</code>

Table 4: The 3 possible download formats for the data feed API.

The requested file does not contain the stream download URLs, when using the API for bulk download you have to construct this by yourself. Download URLs can be constructed by using information from a *normal* download URL (you can get one from the Marketplace DF interface) combined with the *ID* (stream ID) from the API response.

In the download URLs the following parameters are important to use / replace:

**{dfid}** = ID in API response, get this from the API and replace this for each stream!

**{asid}** = Zanox adpace ID, get this from a download URL in the Marketplace DF interface.

**{token}** = Publisher token, get this from a download URL in the Marketplace DF interface.

Example replace URL parameters:

`http://df.zanox.com/download?dfid={dfid}&asid={asid}&m=xml_tree&itid=2701&token={token}`

Example download URL:

`http://df.zanox.com/download?dfid=218106&asid=646300&m=xml_tree&itid=2701&token=F87B9E944A486B0DCCBA0DD448EB84A8D4C13094CCCAA7F2AEE080648F14967A5ABCBA`

Note: The API replies all our available streams, but you can only use and download the one where your ad-space has access.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<datafeeds>
  <datafeed>
    <name>bcc.nl folder</name>
    <ID>2000219</ID>
    <programId>9945</programId>
    <segment>shopping</segment>
    <lastParsed>2013-09-03T15:16:18</lastParsed>
    <records>87</records>
    <checkFrequency>4daily</checkFrequency>
    <lastChecked>2013-09-05T16:11:03</lastChecked>
  </datafeed>
  <datafeed>
    <name>hunkemoller.be fr </name>
    <ID>762106</ID>
    <programId>5303</programId>
    <segment>shopping</segment>
    <lastParsed>2013-09-05T00:17:02</lastParsed>
    <records>886</records>
    <checkFrequency>custom</checkFrequency>
    <lastChecked>2013-09-05T10:17:07</lastChecked>
  </datafeed>
  <datafeed>
    <name>telsell.nl </name>
    <ID>702106</ID>
    <programId>12246</programId>
    <segment>shopping</segment>
    <lastParsed>2013-09-05T02:11:26</lastParsed>
    <records>412</records>
    <checkFrequency>daily</checkFrequency>
    <lastChecked>2013-09-05T02:11:26</lastChecked>
  </datafeed>
  <datafeed>
    <name>trioticket.nl</name>
    <ID>2000090</ID>
    <programId>13476</programId>
    <segment>tickets</segment>
    <lastParsed>2013-09-05T02:11:05</lastParsed>
    <records>22</records>
    <checkFrequency>daily</checkFrequency>
    <lastChecked>2013-09-05T02:11:04</lastChecked>
  </datafeed>
</datafeeds>
```

Figure 12: API output example (XML)

## 3 Data feed usage

In this chapter we are not going to teach you how to build a full functional data feed concept, because in our position we 'simply' cannot do this. What we will do is try to explain the basics and give you some tips and examples to help start up a DF concept and use our data feeds as an affiliate product.

### 3.1 Essentials of the affiliate DF web concept

At first realize that a data feed is one of the most complex affiliate products available and to generate conversion with feeds successfully it requires a lot of discipline, creativity and time! Some knowledge of SEO and content writing is highly essential nowadays, because Google penalizes every publisher in their rankings who shows duplicated data and does not impart enough unique and valuable information to their visitors.

We (Zanox M4N) as provider of affiliate feeds try to keep our information as rich as possible and we provide a consistent fixed data structure. For more than 10 years M4N uses the same standard fieldnames. However, no single feed has the same content structure, because almost each source is from a different advertiser who all have their own methods, selections and parameters for generating their output data. We as network and you as publisher have to deal with this fact and the publishers apart have the mission to be creative with the data available. **(And this is exactly the point where success lies concealed, where distinctive publishers can generate added value with feeds and earn commission).**

### 3.2 Mapping techniques

One of the most important element in using feeds is the concept of mapping. At first when you start importing DF data into your own database the column mapping between the source fields and your own database fields needs to be linked. It is recommended to check and assign the correct mapping for each feed you will use. All pre-build DF scripts in the market have their own kind of mapping interfaces.

**Field Mapping:**

Product Name: TEXT/NAME  
 Product Description: TEXT/DESC  
 Image URL: URI/MIMAGE  
 Buy URL: URI/AWTRACK  
 Price: PRICE/BUYNOW

**Optional Index Fields:**

Category: [ ] or use field: CAT/AWCAT  
 Brand: [ ] or use field: BRAND/BRANDNAME

Buttons: Register, Register, Register and Full Import

Field	Sample Data
PROD	
PROD-ID	
TEXT	
TEXT/NAME	by Website Design
TEXT/DESC	Website Design from Mr Site, you&#39;ll have all the tools you need to create a prof...
URI	
URI/AWTRACK	http://www.awin1.com/pclick.php?p=113453739&a=42160&m=1599
URI/MIMAGE	http://brain.pan.e-merchant.com/4/5/00674554/L_00674554.jpg
PRICE	

Figure 13: Field mapping interface in the popular *Price Tapestry* affiliate DF software.

## 3.2.1 Category mapping

A more complicated, but not required, kind of mapping is 'category mapping'. This time-consuming and ongoing activity aims to link the web shop categories in a feed to your own defined standard categories. This can be useful when different advertisers use other synonyms for the same categories and you want them on your own website covering all the category varieties. By using category mapping you can decide which name you want to use. How this technically can be arranged and in what level this is automated is up to the programmer.

*Example category mapping:*

*Suppose you want to show products in your own 'logical' category "Jackets", but several advertisers use different names in their category paths for products that in your understanding are the same:*

*Advertiser A    clothing > jackets  
Advertiser B    Clothes > Jacket  
Advertiser C    Clothing > Coats  
Advertiser D    Apparel > Outerwear*

*With category mapping techniques you can assign all these synonyms for the 'same' product group to your own defined category like "Clothing > Jackets".*

## 3.3 Affiliate data feed software

On the market there are several pre-build data feed scripts available, commercial and non-commercial. It is important that before you choose the script that suits your needs you need to think what you want to use it for. Stand-alone application? Or if the DF data needs to be implemented in a web CMS like Wordpress or Joomla? You can probably use parts of a script or rebuild components. Or if none of the scripts are sufficient, you can decide to build your own script from scratch.

The list below is a collection of useful DF software and tools with some extra information and links to the source website.

DF software / tool	Information	Link
Price Tapestry	One of the first DF scripts but still in development! It works standalone but has also plug-ins for Wordpress and Joomla. Recommended script! Great support!	<a href="http://www.pricetapestry.com">http://www.pricetapestry.com</a>
Magic parser	Standalone parse script to download feed data into your own MySQL database. Create automated imports in combination with cronjobs.	<a href="http://www.magicparser.com">http://www.magicparser.com</a>
MDR (M4N datafeed reader)	No longer in development, but the PHP source code is open and may be used as example.	<a href="http://blog.m4n.nl/m4n-datafeed-import-reader-script">http://blog.m4n.nl/m4n-datafeed-import-reader-script</a>

DF manager	A commercial application for quickly building shops from affiliate feeds.	<a href="http://www.dfmanager.com">http://www.dfmanager.com</a>
WP All Import	Powerful Wordpress plugin that makes it easy to import XML and CSV feeds.	<a href="http://wordpress.org/extend/plugins/wp-all-import/">http://wordpress.org/extend/plugins/wp-all-import/</a>
Digitalquill data feed import plugin	Import XML, RSS and CSV data feeds into Wordpress.	<a href="http://www.digitalquill.co.uk/datafeedplugin/">http://www.digitalquill.co.uk/datafeedplugin/</a>
Wordpress store plug-in	Gives the ability to add a store to your Wordpress site. Feeds from all premium affiliate networks supported.	<a href="http://wordpress.org/plugins/tradetracker-store">http://wordpress.org/plugins/tradetracker-store</a>
Alfie product feed tool	Free and easy to use product feed tool. Standalone and Wordpress plug-in.	<a href="http://www.productfeedtool.nl/site">http://www.productfeedtool.nl/site</a>
Affiliate datafeeds	Data feed component for Joomla.	<a href="http://www.affiliatefeeds.nl">http://www.affiliatefeeds.nl</a>
Datafeed studio	A self-hosted web application to create multiple types of product driven affiliate web pages.	<a href="http://datafeedstudio.com">http://datafeedstudio.com</a>
DataFeedFile	Affiliate data feed information source.	<a href="http://www.datafeedfile.com/">http://www.datafeedfile.com/</a>
Datafeedr	To build and embed an affiliate store into your WordPress blog. No programming required. At this moment only supports our data from M4N and Affiliatewindow.	<a href="http://www.datafeedr.com">http://www.datafeedr.com</a>

*Important note:*

*On none of this software, scripts and plug-ins we give technical support, use is at own responsibility. Some developers gives free support and / or has open support forums.*

### 3.4 Schedule automated data import

For automated data feed import it is necessary (or highly recommended) that you use a scheduled task. Assume that you have a data feed script that requires a background importer to run at a certain time to refresh your database periodically with the advertisers information.

Cron jobs are created with the command line program named crontab. They require that your website is hosted on a Unix-like web server, such as Linux. In general, if your site is running on a Windows based web server, you cannot use this kind of *cron jobs* like we explained here because Windows has

its own scheduler software. Most data feed scripts are built with PHP or Perl and run on a Linux hosted web server.

You also need to be able to connect to your web host using shell (SSH/Telnet) access with software like puTTY. Otherwise, your hosting provider must give you a way to set the crontab tasks by using their control panel.

We assume that you can connect to your web server using shell and work from the command line interface.

*Important note:*

*When you use a free web host, you probably do not have shell access. Neither do most of them allow you to set cron jobs even if this is technically possible with the control panel interface. In general only commercial provided webhosting facilitates shell access.*

### 3.4.1 Setting up cron jobs

At first you have to figure out the schedule for your desired data feed import task(s). Do you want it updated daily? or on an hourly basis? Be sure that if you use shared webhosting, where many websites reside on the same servers, you should not execute your cron job too frequently or you will affect the performance of your own and others activities hosted on the same server. If you really need such a high update frequency for your feeds, you may need to consider a virtual private server (VPS) or a dedicated server.

The second thing you need to do is to write your schedule in the syntax so that crontab will understand it. The crontab format is somewhat cryptic, so this is explained step by step with examples. The basic format of a crontab schedule consists of 6 positions, separated by spaces, formatted as follows:

*minute hour day month day-of-week command-line-to-execute*

Positions	Value range
minute	0-59
hour	0-23
day	1-31
month	1-12
Day of week	0-7 (where both 0 and 7 mean Sun, 1 = Mon, 2 = Tue, etc)
Command line to execute	The program used to run the parameters to the command line if there are any.

The positions have to be in the above exact order, with no empty or missing positions. If you want a particular schedule to run, say, once every day, you can use an asterisk (\*).



Run time	Cron time	Comment
6:30	30 6	Minute and hour AM.
18:30	30 18	Minute and hour PM.
12th January 9:15	15 9 12 1	Minute, hour, day and month.
Once every day at 21:40	40 21 * * *	Minute, hour and every day, month and week-day.
Run every hour on the day	0 * * * *	Minute and every hour, day, month and week-day.
Run every 2 hour on the day	0 */2 * * *	The second field, "*/2", means every second hour.
Run every 12 hour on the day	0 */12 * * *	Means every half natural day.
14:00 on the 1st and 20th of every month	0 14 1,20 * *	The comma, ",", means "and".
3:14 on the 4th and 5th of May (5) <b>and</b> on every Sunday (7)	14 3 4,5 5 7	

If your script must be opened in a web browser, like "import.php" on a Wordpress plug-in script, you need to use the Linux command called "wget". Technically, wget is not really a browser, but it works sufficiently for our purpose this case, which is to simply get the web server to run a specified script , for example "import.php".

```
20 11 3 * * /usr/bin/wget http://www.yourdomain.com/import.php
```

You need to specify the full path to wget since, for security reasons, the cron program usually executes in a 'clean' environment, which means that it will not know how to find the command "wget" unless you tell exactly it where it is.

Instead of "wget" you can use "curl" for some reasons:

```
20 11 3 * * usr/bin/curl http://www.yourdomain.com/import.php
```

Or just straight to the PHP command-line interpreter:

```
20 11 3 * * /usr/local/bin/php -q  
/home/admin/domains/yourdomain.nl/public_html/import.php
```

Example of a cron string with command line value to export and store data on the file system:

```
15 2 * * * /usr/bin/wget http://df.zanox.com/download/stream.csv -O  
/home/admin/domains/yourdomain.nl/public_html/downloads/somefeed.csv
```

**Important note:**

*The absolute path to the program that you want to run (like wget or php) may vary per server. Please contact your hosting for more information about program locations in the operating system.*

Another important thing to know about the crontab command you are constructing is that the complete line (schedule code and command to execute) must fit into one row without line breaks. You cannot put it in two lines for some reasons even if your command is very long.

In case you have a lot of data feeds (streams) which need to be imported frequently, you can construct a text file containing all the needed cron lines (crons.txt) and 'feed' this file to crontab. Be sure that you create a (UNIX) ASCII text file with a program like Notepad++. Do not use (even to edit) a rich word processor like Microsoft Word or similar.

## 3.4.2 Using Linux crontab

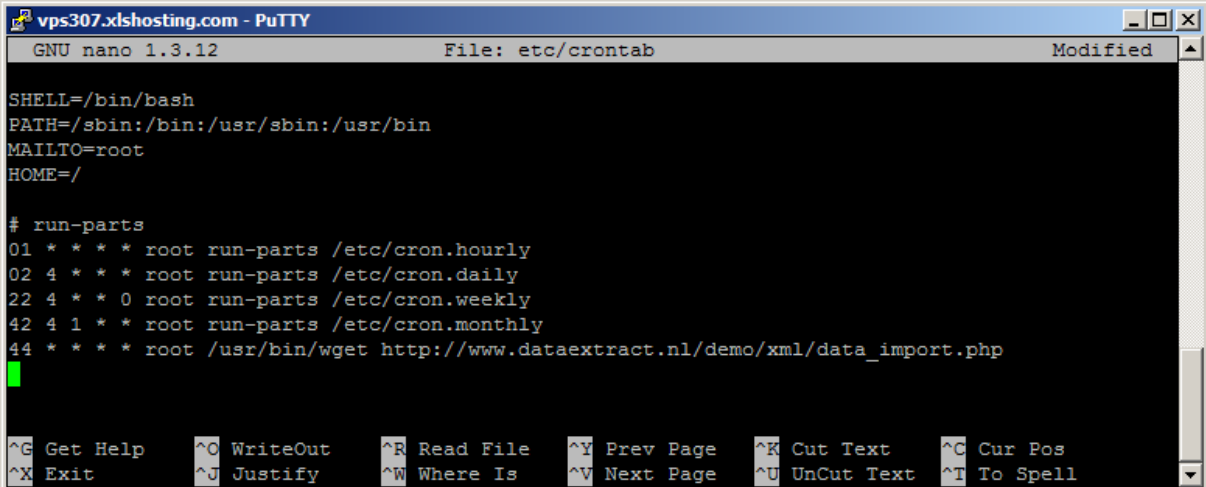
Crontab can be fed from the command line interface (SSH secure shell or Telnet) line by line or as earlier described from a ASCII text file for mass cron lines.

Before you can add or edit cron jobs you have to localize the *crontab* program, this is like:

```
usr/bin/crontab or etc/crontab
```

Knowing the correct path you can open crontab with, for example, 'vi' or 'nano' editor:

```
# nano usr/bin/crontab
```



```
vps307.xdshosting.com - PuTTY
GNU nano 1.3.12 File: etc/crontab Modified
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
44 * * * * root /usr/bin/wget http://www.dataextract.nl/demo/xml/data_import.php

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Figure 14: Example of Linux shell interface, editing crontab with nano.

In case of errors you can define a mail address or if you do not want to send any emails to crontab send the message to /dev/null dump.

Example command with email notification:

```
MAILTO=your@mailaddress.com
30 11 * * * /usr/bin/wget
```

Example command with dump location:

```
30 11 * * * /usr/bin/wget >/dev/null
```

In case you use a mass text file then you first have to upload this file (with your favorite FTP client) to the desired directory on your web server. From the shell command line go to the location where you uploaded the "crons.txt" file and call crontab to load this file from that location:

```
# user/bin/crontab crons.txt
```

Where crons.txt contains all cron lines that you want to schedule. For more information about cron jobs and crontab you can visit: <http://en.wikipedia.org/wiki/Cron>

## 3.5 The data feed parser

Last (but not least) we will explain the principles of parsing, manipulation and storing data.

In the world of data feeds, a parser is one of the components in an interpreter or importer script that takes input texts and builds a data structure giving a structural representation of the input, checking for correct syntaxes in the process. The parsing may be preceded or followed by other steps, like manipulating data and importing the data into a database.

Well-informed programmers know how to work with databases, parsers and data manipulation scripts so we will not elaborate this technical matter.

### 3.5.1 Example import code

Based on a proven and stable script we will give an example of an import script with the basic ZanoX fields that you can use in combination with the *Magicparser* parser software and MySQL database. *Magicparser* can be obtained at <http://www.magicparser.com>.

With the 4 printed separate PHP files and the obtained magicparser.php file you can fill a MySQL database table with XML data based on a selection of fields from your ZanoX feeds and publish it on a HTML page. The highlighted code may be copied and changed for own purposes.

First, construct a *database.inc.php*, this is the connection file that has to be included in the other PHP files in order to connect to your created database. Change lines 4-7 as required for your server and database. You may first need to create the MySQL database using your web hosting shell or control panel.

#### *database.inc.php*

```
<?php
// edit the following lines to match your MySQL environment
// you may need to create the database in $databaseName if it does not

$databaseServer    = "localhost";
$databaseName      = "dbexample";
$databaseUsername  = "root";
$databasePassword  = "";

function database_querySelect($sql, &$rows)
{
    global $databaseServer;
    global $databaseName;
    global $databaseUsername;
    global $databasePassword;

    $link = mysql_connect($databaseServer, $databaseUsername, $databasePassword);

    mysql_select_db($databaseName, $link);

    $result = mysql_query($sql, $link);
```

```
if (!$result)
{
    print "[".$sql."][".mysql_error()."];
}

$rows = array();

while($row = mysql_fetch_array($result,MYSQL_ASSOC))
{
    $rows[] = $row;
}

return mysql_num_rows($result);
}

function database_queryModify($sql,&$insertId)
{
    global $databaseServer;
    global $databaseName;
    global $databaseUsername;
    global $databasePassword;

    $link = mysql_connect($databaseServer,$databaseUsername,$databasePassword);

    mysql_select_db($databaseName,$link);

    $result = mysql_query($sql,$link);

    if (!$result)
    {
        print "[".$sql."][".mysql_error()."];
    }

    $insertId = mysql_insert_id();

    return mysql_affected_rows();
}

function database_safe($text)
{
    return mysql_escape_string($text);
}
?>
```

Second, you have to construct *create\_table.php*. You have to run this file once to create the table with the correct field definitions related to the fields from the XML file(s) that you need imported into the database.

#### *create\_table.php*

```
<?php
require("database.inc.php");

$sql = "
DROP TABLE IF EXISTS `products`
";

$sql = "
CREATE TABLE `products`
(
id INT(11) NOT NULL auto_increment,
`url` VARCHAR(255) NOT NULL default '',
`title` VARCHAR(255) NOT NULL default '',
`description` TEXT NOT NULL default '',
```

```
`offerid` VARCHAR(255) NOT NULL default '',
`image` VARCHAR(255) NOT NULL default '',
`price` DECIMAL(10,2) NOT NULL default '0.00',
`category` VARCHAR(255) NOT NULL default '',
`subcategory` VARCHAR(255) NOT NULL default '',
`stock` VARCHAR(255) NOT NULL default '',
`timetoship` VARCHAR(255) NOT NULL default '',
`ean` VARCHAR(255) NOT NULL default '',
`price_shipping` VARCHAR(255) NOT NULL default '',
`price_old` DECIMAL(10,2) NOT NULL default '0.00',
`vendor` VARCHAR(255) NOT NULL default '',
`category_path` VARCHAR(255) NOT NULL default '',
`thirdcategory` VARCHAR(255) NOT NULL default '',
`largeimage` VARCHAR(255) NOT NULL default '',
`color` VARCHAR(255) NOT NULL default '',
`material` VARCHAR(255) NOT NULL default '',
`sku` VARCHAR(255) NOT NULL default '',
`size` VARCHAR(255) NOT NULL default '',
`gender` VARCHAR(255) NOT NULL default '',
`fourth_category` VARCHAR(255) NOT NULL default '',

PRIMARY KEY (id)
)
ENGINE=MyISAM;
";

database_queryModify($sql,$result);

print "Done. Table `products` created.";
?>
```

Third, the *import.php* file is created. This can technically be seen as the 'mapping page' where the XML fields are connected to the right MySQL table fields. In this example we gave the table fields the exact same names as the XML feed fields to keep it clear, of course normally you can devise your own desired names. Notice that when your source is XML, the fieldnames in the import code must be defined all UPPER\_CASE, when CSV this must be just like in the source.

Important is the *\$filename* variable which holds the download URL obtained from the Zanox marketplace DF interface. In this example we used a XML test stream, limited to 10 records and clean deep link URLs.

Also you see the constructed format string based on the element structure from the XML source file:

```
"xml|M4N/DATA/RECORD/"
```

For example in case of CSV with TAB separator and double quote field surroundings:

```
"csv|9|1|34"
```

### *Import.php*

```
<?php
require("database.inc.php");

require("MagicParser.php");

$count = 0;

function myRecordHandler($record)
{
    global $count;
```

```
$sql = "
INSERT INTO `products` SET

`url` = '".database_safe($record["URL"])."',
`title` = '".database_safe($record["TITLE"])."',
`description` = '".database_safe($record["DESCRIPTION"])."',
`offerid` = '".database_safe($record["OFFERID"])."',
`image` = '".database_safe($record["IMAGE"])."',
`price` = '".database_safe($record["PRICE"])."',
`category` = '".database_safe($record["CATEGORY"])."',
`subcategory` = '".database_safe($record["SUBCATEGORY"])."',
`stock` = '".database_safe($record["STOCK"])."',
`timetoship` = '".database_safe($record["TIMETOSHIP"])."',
`ean` = '".database_safe($record["EAN"])."',
`price_shipping` = '".database_safe($record["PRICE_SHIPPING"])."',
`price_old` = '".database_safe($record["PRICE_OLD"])."',
`vendor` = '".database_safe($record["VENDOR"])."',
`category_path` = '".database_safe($record["CATEGORY_PATH"])."',
`thirdcategory` = '".database_safe($record["THIRDCATEGORY"])."',
`largeimage` = '".database_safe($record["LARGEIMAGE"])."',
`color` = '".database_safe($record["COLOR"])."',
`material` = '".database_safe($record["MATERIAL"])."',
`sku` = '".database_safe($record["SKU"])."',
`size` = '".database_safe($record["SIZE"])."',
`gender` = '".database_safe($record["GENDER"])."',
`fourth_category` = '".database_safe($record["FOURTH_CATEGORY"])."'

"; // no comma after last! -----&

database_queryModify($sql,$result);

$count++;
}

$sql = "TRUNCATE `products`";

database_queryModify($sql,$result);

$filename = "http://df.zanox.com/download/stream-218106-
8433.xml?dfid=218106&duid=8433&dhid=268A&test=true&m=xml_tree&max=10&itid=3126&el=u
ser_";

if (!MagicParser_parse($filename,"myRecordHandler","xml|M4N/DATA/RECORD/"))
{
    die(MagicParser_getErrorMessage());
}

print "Done. ".$count." records imported.";
?>
```

The fourth and last file is a very 'simple' script to request the database fields and for example to display in a web browser. With a basic SQL command we print all the columns from the first 10 records.

```
"SELECT * FROM `products` LIMIT 10"
```

It is clear that this PHP script is a very basic example of a data feed web concept and can be developed to a full functional and designed web page with the desired elements of the original source XML feed.

## data\_display.php

```
<?php
require("database.inc.php");

$sql = "SELECT * FROM `products` LIMIT 10";

if (database_querySelect($sql,$prods))
{
    foreach($products as $prod)
    {
        print "<h1><a href='". $prod["url"]. "'>". $prod["title"]. "</a></h1>";
        print "<img src='". $prod["image"]. "' />";
        print "<p>". $prod["description"]. "</p>";
        print "<p>". $prod["offerid"]. "</p>";
        print "<p>". $prod["price"]. "</p>";
        print "<p>". $prod["category"]. "</p>";
        print "<p>". $prod["subcategory"]. "</p>";
        print "<p>". $prod["stock"]. "</p>";
        print "<p>". $prod["timetoship"]. "</p>";
        print "<p>". $prod["ean"]. "</p>";
        print "<p>". $prod["price_shipping"]. "</p>";
        print "<p>". $prod["price_old"]. "</p>";
        print "<p>". $prod["vendor"]. "</p>";
        print "<p>". $prod["category_path"]. "</p>";
        print "<p>". $prod["thirdcategory"]. "</p>";
        print "<p>". $prod["largeimage"]. "</p>";
        print "<p>". $prod["color"]. "</p>";
        print "<p>". $prod["material"]. "</p>";
        print "<p>". $prod["sku"]. "</p>";
        print "<p>". $prod["size"]. "</p>";
        print "<p>". $prod["gender"]. "</p>";
        print "<p>". $prod["fourth_category"]. "</p>";
    }
}
?>
```

Upload these 4 files to the desired folder on your (Linux) web server together with *magicparser.php* and follow these steps:

1. Create a new MySQL database (with your hosting control panel or from the shell command line).
2. Change database connection settings in *database.inc.php* in accordance to your created database.
3. Run once: [http://yourdomain.com/somefolder/create\\_table.php](http://yourdomain.com/somefolder/create_table.php)
4. Run (on desired frequency\*): [http://yourdomain.com/somefolder/data\\_import.php](http://yourdomain.com/somefolder/data_import.php)
5. Run to test: [http://yourdomain.com/somefolder/data\\_display.php](http://yourdomain.com/somefolder/data_display.php)

\*Run *data\_import.php* every day at 4:00 from a cron job that may look like:

```
0 4 * * * USERNAME /usr/bin/wget http://yourdomain.com/somefolder/data_import.php
```

## 4 Zanox M4N DF field definitions

Because the requirements of publishers vary per industry, we define “data feed segments”. Each segment has its own definition of standard fields which are applicable to the specific industry. Within each set of fields we strongly recommend our advertisers to deliver their source feed(s) as complete as possible, some fields are mandatory but unfortunately often not all fields can be provided.

We define the following main segments: Shopping, Tickets, Jobs, Education, Travel and Mobile.

All our data feeds contain a set of basic fields which practically applies to all segments.

The fields given below are no guarantee that advertisers actually complete these data, this always depends on the advertiser.

### 4.1 Basic fields (and standard shopping segment fields)

Field name	Format	Comment
URL	URL	Deep link linking to the product detail overview page.
Title	Text	Title/name of the product
Description	Text	Description of the product.
Offerid	Text	Unique product identifier on webshop level
Image	URL	Absolute path to the product image
Price	Number	Price of the offered product or service, shipping price excluded, VAT included.
Category	Text	Main category of the product.
Subcategory	Text	Sub category of the product.
Large_image	URL	Absolute path to the product large image.
Stock	Text	Stock indication or stock count (advertiser decision).
Timetoship	Text	Time that the product can be delivered.
Price_shipping	Number	Shipping costs.
Price_old	Number	The previous “from” price of the offered product. Can be used to display the product as an offer. Shipping price excluded, VAT included.
Vendor	Text	The brand or manufacturer of the product.
Category_path	Text	Sequence of hierarchical category levels (separated).



## 4.2 Segment fields

### 4.2.1 Shopping (fashion)

Field name	Format	Comment
Size	Text	Size of the product.
Size_stock	Text	For each size the number of products in stock.
Color	Text	Color of the product
Material	Text	Material of the product
Gender	Text	Target gender of the product: male, female, unisex, e.g.
Model	Text	The model name of the product.
EAN	Number	European Article Number (unique identifier on product level).
Third category	Text	Third category of the product.
Condition	Text	Status off the product
Base price / unit	Text	The base price per unit: base price is EUR 1.99 per 100 ml

### 4.2.2 Travel

Field name	Format	Comment
Date_from	Text	Start or arrival date.
Date_to	Text	End or departure date.
Destination_country	Text	Destination country name or ISO 3361-1 alpha-2 country code.
Destination_name	Text	Name of the destination e.g. hotel name, park name, etc.
Destination_region	Text	Region of destination.
Destination_type	Text	Type of destination e.g. hotel, motel, pension, cruise, etc.
Travel_arrangement	Text	weekend, midweek, christmas, etc.
Travel_description	Text	Description of destination
Travel_duration	Number	Travel duration in days.

Travel_pax_min	Number	Minimum number of participants.
Travel_pax_max	Number	Maximum number of participants.
Travel_photo_1	URL	Absolute path to the travel large image.
Travel_rating	Number	Travel/Hotel rating.
Travel_transport	Text	Bus, Airplane, Car, etc.
Travel_type	Text	Sun vacation, winter sport, etc.

## 4.2.3 Mobile

Field name	Format	Comment
Price_initial	Number	Initial costs like setup fee (excluding phone price).
Model	Text	Phone model.
Provider	Text	Provider name.
Action	Text	Network deal; e.g. "First three months 50% discount!"
Present	Text	Welcome present.
Abo_price_effective	Number	Effective costs per month, discounts and offers taken into account.
Abo_price	Number	Subscription costs per month. This is also the "action" price, if an offer.
Abo_time	Number	Subscription duration in months.
Abo_type	Text	Name of the subscription plan.
Abo_minutes	Number	Number of minutes included in the subscription.
Abo_sms	Number	Number of text messages included in the subscription.
Abo_internet	Text	Internet yes / no or internet subscription name.
Abo_data	Number	Number of MBs data traffic included in the subscription.
Abo_internet_price	Number	Disaggregated cost per month for data subscription.
Abo_price_standard	Number	Standard subscription costs per month. This is the "normal" price of the subscription if there is an action.

Action_duration	Number	Network deal duration in months.
Subscription_renewal	Text	Renewal of subscription, yes or no.
Operating_system	Text	Operating system on the phone, e.g. "Android".

## 4.2.4 Jobs

Field name	Format	Comment
Jobs_group	Text	Job field.
Jobs_date	Text	Vacancy date.
Jobs_location	Text	Location of the job.
Jobs_education	Text	Required education level.
Jobs_hours	Text	The number of hours per week.
Jobs_region	Text	Region of the job location.
Jobs_employername	Text	Name of the employer.
Jobs_jobtype	Text	The type of the job, e.g. "fixed contract", "part time", etc.
Jobs_intermediate_location	Text	City where the agency is located.
Jobs_intermediate_postalcode	Text	Postal code of the agency location.
Jobs_intermediate_country	Text	Country of the agency location.
Jobs_intermediate_region	Text	Region of the agency location.
Jobs_salary	Text	Salary.
Jobs_employment	Text	Job employment.
Jobs_experience	Text	Level of experience.

## 4.2.5 Tickets (airline)

Field name	Format	Comment
departure_id	Text	ID from departing location, e.g. AMS
departure_name	Text	Name of departing location
departure_country	Text	Country of departure
departure_region	Text	Region or continent of departure
destination_id	Text	ID from departing location, e.g. GVA
destination_name	Text	Name of the destination
destination_country	Text	Destination country
destination_region	Text	Region or continent of destination
airline_id	Text	Airline ID
airline_name	Text	Airline name, e.g. KLM, Etihad Airways, Transavia, etc.
airline_logo	URL	Image of the airline logo
date_from	Text	Departure date
date_to	Text	Returning date
terms_stay_min	Text	Minimum number of nights
terms_stay_max	Text	Maximum number of nights

## 4.2.6 Education

Field name	Format	Comment
education_time	Text	The number of months the course will take.
education_max		Maximum number of participants in the course.
education_min		Minimum number of participants in the course.

education_location		The city where the course will take place, incase e.g. "Amsterdam"
education_level		Minimum level of finished education needed to start this course.
education_daypart		Description of the course hours.
education_type		The type of the course, e.g. "Home studies", "training", etc.
education_previous		Minimum level of finished education needed to start this course.
education_next		Possibilities for continuation course.
education_institute		Institute name.
education_field		Possible field of work after finishing the course
education_start_fee		Initial or start price
education_month_price		Monthly course price

#### 4.2.7 Daydeal (extra fields)

Field name	Format	Comment
Publisher	Text	Name of the daydeal merchant.
Time	Text	Time that the deal will change. e.g. 0:00 or 12:00
Logo	URL	Logo of the daydeal merchant.
Merchant_id	Text	Program ID of the daydeal merchant.
Zupid	Text	Old Zanox ID, used for daydeals if no offerID exist. MD5 hash.

---

## DISCLAIMER

---

Zanox M4N is not responsible for, and expressly disclaims all liability for, damages of any kind arising out of use, reference to, or reliance on any information contained within this document.

Misprints, typos and errors in the example scripts reserved.

You are allowed to copy, use and change the example scripts in this document, but please do not request us for support on the scripts, techniques and programming languages published.

We do give support on our Marketplace en data feed interface as well as on stream data quality and (export) problems regarding to the content. As best as we can we will try to solve issues in co-operation with the concerning advertiser.

