

PerceptiveMCAPI – v 1.3.0: Change Log – BETA 2

Release Notes – General

- A new `AccessType` enumerated value of 'JSON' was added
 - This is future use for v1.3.1 to allow JSON input objects
 - DO NOT USE this value for v1.3.0, this will throw a Not Implemented error.
- A new `OutputType` enumerated value of 'Unformatted' was added
 - This bypasses formatting responses into .NET objects. This is future use for v1.3.1
 - The *methodOutput* object will therefore contain only the values available in the *api_BaseOutput* class and the *methodParms* class. The 'result' property will not be filled.
 - This output type is not valid for the `XmlRpc` access type
 - DO NOT USE this value for v1.3.0, this will throw an Access to Output Type Mismatch error.
- **Support for MailChimp Export Api is ACTIVE**
 - OK, I said it wasn't going to be done until 1.3.1, but I lied – it's ready now. I just got tired of seeing it on the project list.
 - This is a separate .dll (PerceptiveMCAPI.Export.dll), and it references & requires the PerceptiveMCAPI.dll – and uses the same approach and settings as the core wrapper. So your api key defaults, the EnumValues, etc. are used.
 - Place PerceptiveMCAPI.dll, PerceptiveMCAPI.Export.dll, and CookComputing.XmlRpcV2.dll in the same folder.
 - Add references to PerceptiveMCAPI.dll and PerceptiveMCAPI.Export.dll to your project
 - You need the following additional namespaces in your program to use the export:
 - `using PerceptiveMCAPI.Export;`
 - `using PerceptiveMCAPI.Export.Types;`
 - The *api_Response* property contains the original data returned from MailChimp as a `List<string>` property, one entry per row returned including the header; each row is a JSON array.
 - The 'Results' class is comprised of two properties; a header and a data property.
 - header – is a `List<string>` collection that contains the export row field 'names'
 - data – is a `List<List<object>>` collection; values are a list of field value objects, in a list of records
 - `AccessType`, `OutputType`, and `MethodType` values are essentially ignored as the only valid combination is "Serial, JSON, POST"

Check the [Api Methods](#) notes at the end of this document for further method-specific changes.

PerceptiveMCAPI – v 1.3.0: Change Log – BETA 1

Release Notes – General

- This release targets the new functionality and formats used in the MailChimp Api v1.3.0 changes. These notes only reflect changes made since v1.2.5.3 of the PerceptiveMCAPI wrapper.
- Support for JSON objects as input will be deferred until PerceptiveMCAPI v.1.3.1 and combined with other features I think you'll like; think Export and STS.
- The PerceptiveMCAPI_UI ('testing' UI) is dropped, or I suppose you could say temporarily re-purposed. It was too time-consuming to keep going as part of this project and too inflexible for testing the wrapper with the increased number of methods and complexity of formats. Its temporary new role will be one of providing implementation samples in the short term, until its utter demise after help docs & samples are put on my (hopefully) soon-to-be-reworked website.
- One of the main reasons it took this long to get the v1.3 release out was that the 'testing' UI was really built as a prototype for method implementation – not as a testing platform. So I decided to write a testing platform that, while still a work in progress, has already made it easier to run repeatable test scenarios, and target edge cases, etc. I had planned on doing it after this release, but just ran into too much time consuming effort making changes on the UI system – as well as the 'manual' testing time – and decided to bite-the-bullet now rather than later. And since I've been doing only MVC development for the last year or so, looking at changing the web form controls again made my head hurt. Things will go faster from this point forward, but that delayed things quite a bit.
- Please remember that, until I get my website re-worked, the MailChimp site documentation is the primary source for parameter names & values and how things should work, with my documentation as supplemental information about my .NET wrapper specifics.

Release Notes – Api Methods

Campaign Related

campaignEcommOrderAdd

All Ecomm related methods have been 're-worked' to use a common order and order-item definition. This was semi-done last release, but cleaned-up more this time.

campaigns

Filter values as 'multiples separated by commas' is not supported at this time.

I have now tested and confirmed that the comma-delimited multiples option work, as long as you do a ToString() for the filter parameters using enumerated values (status, type).

You may also pass the multiple parm values as a List<object> collection and they will be converted to the 'multiples separated by commas' format – either works.

campaignTemplateContent

Only raw data is returned at this point, nothing formatted. Jesse's message in the MC documentation gives me some pause about trying to format this on a generic basis. I'll take a look at it for Beta 2.

Formatted content is returned as of Beta 2.

Campaign Stats

campaignAbuseReports

Completed, but the MC documentation is incorrect. This method uses the 'total + data' output pattern.

campaignBounceMessages

Server Error (500) is returned when using the 'since' parameter.

Believed to be MC issue, but need to investigate further. All other options work.

MailChimp Fixed.

campaignStats

Timewarp stats are not formatted or output. I still need to get some test data to verify proper formatting. Target is the Release Candidate.

Campaign Report Data

*** No known issues

Ecommerce

ALL

All Ecomm related methods have been 're-worked' to use a common order and order-item definition.

Folder Related

*** No known issues

Helper

chimpChatter

At the time of Beta 1, MailChimp was re-working this internally and the data feed on-site was turned-off. When they turned it back on, all the chatter 'type' values had been changed. I've decided that EnumValues aren't going to work (or be any real value-add) for this field so the 'type' field is now output as a string. I have left the previous EnumValue definition in place at this point for backward compatability, but it is no longer used, and should be considered deprecated.

List Related

listAbuseReports

Always returns a value of 1 for the 'total' column. Believed to be a MC issue, will investigate.

MailChimp Fixed.

listClients

I don't have a large enough sample to test the formatting against. Will try to get more for Beta 2.

Completed in Beta 2.

listInterestGoupingAdd

An EnumValues entry of `grouping_type_field` is used with a default of 'checkboxes'.

listInterestGoupings

An EnumValues entry of `grouping_type_field` is used for the `form_field` column type.

listMemberActivity

I changed the result format from what is in the documentation. Multiple email addresses can be specified in a request, and yet the `email_address` is NOT carried in the output array entries. You had to match the input array entry order with the output array order to know what data goes to who. Not a terribly helpful structure for data manipulation.

So I have included the `email_address` in each result entry to facilitate ease of use.

listMemberInfo

All new data elements are returned except the 'clients' array. I need sample data to work from, so it's returned empty for now. *Target is the Release Candidate.*

lists

The 'modules' array in the results is returned empty as I had no sample data to work from, and the output format is not specified in the MC docs, although it *sounds* like it's a simple string array.

Target is the Release Candidate.

listWebhook functions

Caution: MC does not consider entries to be the same when the protocol is included/omitted, e.g., 'www.dbm.com' and 'http://www.dbm.com' are NOT considered to be the same.

Security Related

*** No known issues

Template Related

*** No known issues

Export Api

Newly added project; see change notes for Beta 2

lists

*** No known issues

Road Map for v1.3.1

- Input to accept JSON objects
- External settings supported.
- y mucho más