



WEB API BUSINESS OVERVIEW

QuickPivot
33 Arch Street
9th Floor
Boston, MA 02110
quickpivot.com
617.880.4000

© Copyright 2014



WEB SERVICES – API METHODS

Table of Contents

Introduction	11
Message Manager Web Methods	12
AddJobNotes	12
ApproveBroadcast	12
ChangeBroadcastType	12
CloneBroadcast	12
CloneBroadcast_Supp	12
CloneBroadcastInFolderWithMsg	13
CloneBroadcastWithMsg	13
CreateBroadcast	13
CreateBroadcastWithAutoResponse	13
CreateBroadcastWithMsg	13
CreateBroadcastWithMsgandProjectID	13
CreateBroadcastwithProjectID	13
CreateListFromTable	14
CreateListFromTableWithPrepFlags	14
CreateListFromTableWithType	14
CreateLiteSender	14
CreateNewBroadcast	14
CreateRecipientList	14
CreateSender	14
CreateSplitBroadcastWithMsg	14
CreatPostBroadcastNotification	15
EndRecipientList	15
GetActivityDet	15
GetAlleBJobs	15
GetAOLContent	15
GetAOLSplitContent	15

QuickPivot
33 Arch Street
9th Floor
Boston, MA 02110
quickpivot.com
617.880.4000



GetAppTransferURL 15

GetBCCCounts 16

GetBounceCount 16

GetBounceDet 16

GetBounceDetByDate 16

GetBounceDetBySender 16

GetBroadcastID 16

GetBroadcastsByDate 16

GetBroadcastsByFolder 16

GetBSDetailsByDate 16

GetCampaignInfo 17

GetContentFromCM 17

GetCreativeSent (located in Auxilliary Email Web Services) 17

GetDeliveryDate 17

GetEJobID 17

GetEJobsByDate 17

GetEmailList 17

GetF2FContent 17

GetFromInfo 18

GetHTMLContent 18

GetHTMLCount 18

GetHTMLDet 18

GetHTMLSplitContent 18

GetLastProofStatus 18

GetLinksTracked 18

GetListCount 18

GetMobileContent 18

GetOODet 19

GetOptOffCount 19

GetOptOffList 19

GetPrepReportXML 19

GetPrepStatus 19

GetPreview (located in Auxilliary Email Web Services) 19

GetProcessedDet 19

GetProofList 19

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000



GetReplyToInfo 20

GetSender 20

GetSharedContent 20

GetSPAMScore 20

GetStatus 20

GetSubject 20

GetSuppressedNames 21

GetSuppressionList 21

GetTextContent 21

GetTextSplitContent 21

GetTokensUsed 21

GetTotalClicksCount 21

GetTrackCount 21

GetTrackDet 21

GetUniqueClicksCount 22

GetViewAsWebPageContent 22

LoadContent 22

LoadContentTokens 22

LoadEncodedContent 22

LoadF2FContent 22

LoadList 22

LoadListFile 22

LoadListFileWithPrepFlags 23

LoadListWithPrepFlags 23

LoadLMList 23

LoadLMListWithFlags 23

LoadLMSegment 23

LoadLMSegmentWithFlags 23

LoadMobileContent 23

LoadSharedContent 24

LoadSplitContent 24

LoadUpLoadedList 24

LoadUpLoadedListWithPrepFlags 24

LoadViewAsWebPageContent 24

ModifyCampaignInfo 24

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000



ModifyDeliveryDate 25

ModifyFromInfo 25

ModifyPrepOptions 25

ModifyReplyToInfo 25

ModifySettings 25

ModifySplitSettings 25

ModifySplitSubject 25

ModifySubject 25

ProofBroadcast 25

ProofBroadcastToList 25

ProofBroadcastWithZones 26

ReprepList 26

SendPrepReportEmail 26

SetSocialSharing 26

SetSocialSharingWithDesc 26

SPAMTest 26

SubmitBroadcast 26

SubmitBroadcastWithFlags 27

SubmitRPTest 27

TestAutoEB 27

TrackAllLinks 27

UploadImages 27

SetSocialSharing (located in Auxilliary Email Web Services) 27

SetSocialSharingWithDesc (located in Auxilliary Email Web Services) 27

Data Manager Web Methods 29

 AddTableColumns 29

 AppendList 29

 AppendListFromFTP 29

 CheckExportListFTPRequestStatus 29

 CheckExportListStatus 29

 CheckGenerateListRequestStatus 29

 CheckGenerateSplitListRequestStatus 29

 CheckNewListRequestStatus 30

 CheckUpdateOrAppendOrReplaceListRequestStatus 30

 CreateSegment 30

QuickPivot
 33 Arch Street
 9th Floor
 Boston, MA 02110
 quickpivot.com
 617.880.4000



DeleteList.....	30
ExportList.....	30
ExportListToFTP.....	30
ExportSegment.....	30
ExportSegmentToFTP.....	30
ExportSet.....	31
ExportSetToFTP.....	31
GenerateList.....	31
GenerateSplitList.....	31
GetFolderContent.....	31
GetFolderDetails.....	31
GetFolderIDsByName.....	31
GetFolderTree.....	31
GetListColumnDetails.....	31
GetListDetails.....	32
GetListIDByName.....	32
GetListPrepStatus.....	32
GetPrepReportXml.....	32
GetSegmentDetails.....	32
GetSegmentIDByName.....	32
GetSegmentParameterList.....	32
GetSetDetails.....	33
GetSetIDByName.....	33
GetSetParameterList.....	33
LoadLargeUploadList.....	33
LoadUploadedList.....	33
OptIn.....	33
OptOut.....	34
PrepList.....	34
ReplaceList.....	34
ReplaceListFromFTP.....	34
Resubscribe.....	34
SendPrepReportEmail.....	34
UpdateList.....	34
UpdateListFromFTP.....	35

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000



UploadNewList.....	35
UploadNewListFromFTP	35
UploadZipFileFromFTP	35
ViewList.....	35
Content Manager Web Methods.....	36
CreateFolder	36
CreateNewHTMLContent.....	36
CreateNewHTMLContentFromFTP	36
CreateNewTextContent	36
CreateNewTEXTContentFromFTP	36
GetContentDetails	36
GetContentIDsByName	36
GetFolderContent.....	36
GetFolderDetails.....	37
GetFolderIDsByName.....	37
GetFolderTree.....	37
GetFormDetails.....	37
GetFormIDsByName.....	37
GetLayerDetails	37
GetLayersByZone	37
GetReferencedZones.....	37
GetResourceDetails.....	38
GetResourceIDsByName	38
GetZoneDetails.....	38
GetZoneIDsByName	38
PublishContent	38
UploadNewResourceFromFTP	38
UploadResourceToFTP	38
Report Manager Web Methods.....	39
GetActivityDet	39
GetBCCCounts.....	39
GetBounceCount.....	39
GetBounceDet	39
GetBounceDetByDate	39
GetBounceDetBySender	40

QuickPivot
 33 Arch Street
 9th Floor
 Boston, MA 02110
 quickpivot.com
 617.880.4000



GetBroadcastsByDate	40
GeteBJobsByDate	40
GetEventsByDate	40
GetEventsByDateDX	40
GetHTMLCount	40
GetHTMLDet	40
GetLinksTracked	41
GetOODet	41
GetOptOffCount	41
GetOptOffList	41
GetProcessedDet	41
GetSMTPBcastID	41
GetTotalClicksCount	42
GetTrackCount	42
GetTrackDet	42
GetUniqueClicksCount	42
TestService	42
Campaign Results Export Web Methods	43
ActivityReportByDateRange	43
ActivityReportByRelativeTimeframe	43
BounceReportByDateRange	43
BounceReportByRelativeTimeframe	43
ComplaintReportByDateRange	43
ComplaintReportByRelativeTimeframe	43
DownloadRequestedReport	43
F2FReportByDateRange	44
F2FReportByRelativeTimeframe	44
GetReportFileName	44
GetReportRequestDetails	44
GetReportRequestStatus	44
LinkAliasReportByDateRange	44
LinkAliasReportByRelativeTimeframe	44
OpenReportByDateRange	44
OpenReportByRelativeTimeframe	45
OptoutReportByDateRange	45

QuickPivot
 33 Arch Street
 9th Floor
 Boston, MA 02110
 quickpivot.com
 617.880.4000



OptoutReportByRelativeTimeframe.....	45
RawBounceDataByDateRange.....	45
RawBounceDataByRelativeTimeframe.....	45
RawComplaintDataByDateRange.....	45
RawComplaintDataByRelativeTimeframe.....	45
RawF2FDataByDateRange.....	46
RawF2FDataByRelativeTimeframe.....	46
RawOpenDataByDateRange.....	46
RawOpenDataByRelativeTimeframe.....	46
RawOptoutDataByDateRange.....	46
RawOptoutDataByRelativeTimeframe.....	46
RawSentDataByDateRange.....	46
RawSentDataByRelativeTimeframe.....	47
RawSharingDataByDateRange.....	47
RawSharingDataByRelativeTimeframe.....	47
RawTrackDataByDateRange.....	47
RawTrackLinkDataByDateRange.....	47
RawTrackLinkDataByRelativeTimeframe.....	47
RawTracksDataByRelativeTimeframe.....	48
SentReportByDateRange.....	48
SentReportByRelativeTimeframe.....	48
SharingReportByDateRange.....	48
SharingReportByRelativeTimeframe.....	48
TrackLinkReportByDateRange.....	48
TrackLinkReportByRelativeTimeframe.....	48
TrackReportByDateRange.....	49
TrackReportByRelativeTimeframe.....	49
Workflow Manager Web Methods.....	50
WFEvent.....	50
WFStart.....	50

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000

© Copyright 2014



QuickPivot
33 Arch Street
9th Floor
Boston, MA 02110
quickpivot.com
617.880.4000

© Copyright 2014



INTRODUCTION

The QuickPivot web services serve as an API layer to allow external programs to accomplish many of the same tasks that might be accomplished using the QuickPivot product web interfaces. Operations such as create, modify, proof, and approve broadcasts and messages; upload lists, run queries, export content, and insert content are available using web service calls into the QuickPivot API. The web services also provide the ability to trigger the delivery of saved email messages previously created in the QuickPivot Message Manager application as well as maintain and modify the lists hosted within QuickPivot's Data Manager.

The robustness and functionality of the web services make QuickPivot uniquely positioned to provide integration of its delivery capabilities with any customized email application, content, or e-marketing application. The ability to integrate our email delivery and tracking capabilities, list management, and content management functions with our customer's applications comes from our understanding the most common functions associated with eMarketing applications and wrapping those functions in a series of web service calls. Since some customers operate in a hybrid model (using both the web interface and the web services) you will find that all operations that are surfaced within the web services will be reflected immediately within the interface as well. So the creation of a broadcast using the web services will result in a new entry visible in the user interface. As a result, customers using the web services can adopt a either a fully automated model using only the web services or a hybrid model using a combination of web services and the user interface. This unique approach ensures that the customers who wish to automate portions of their process such as list or content uploading can do so while allowing functions such as proofing and approval to be accomplished more manually. Both approaches are fully interchangeable due to the sophisticated architecture of the QuickPivot application toolset thereby giving our customers maximum flexibility. All web services were designed to integrate with database and work-process requirements of our clients using standard web 2.0 methods and architecture.

The web services sit behind the same protective layer that the QuickPivot MessageManager application sits. The web services API resides on an HTTPS site (SSL) and is protected by IP address through the QuickPivot firewall. This ensures that only IP addresses that have been authorized to utilize the web services are passed through the firewall and are able to make calls to the web service. There are 3 basic methods to invoke each web service; SOAP, HTTP GET, and HTTP POST. The web services are "self documenting" in that once one arrives at the specific web service HTML page users are presented with information regarding how to make calls to the web service as well as the command syntax for making the calls. This document will serve to mostly reiterate what is on the web service pages and provide a brief overview of the parameters necessary to make a successful call to the web services.

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000

© Copyright 2014



MESSAGE MANAGER WEB METHODS

Message Manager Web Services allow integration partners to create, modify, and send various types of broadcasts. The web services can be used to accomplish virtually any task that can be completed using the User Interface. The Message Manager Web Service provides numerous API methods to accomplish the same basic task. For example there are 5 different web methods to create a new broadcast and over 6 different web methods to load recipients into a broadcast. The variety of functions to accomplish the same basic task provides partners with the ability to tailor their integration for virtually any situation.

The following sections provide a basic overview of each of the Message Manager web service methods. For a more detailed description of input parameters and expected return values please utilize the QuickPivot Web API documentation.

AddJobNotes

Allows for the addition of notes that are maintained with the specified job. Email Manager allows notes to be added to any job to allow for multiple users to share information while creating a job. Notes are added automatically by the system for certain operations however this method allows API users to insert notes for future examination.

ApproveBroadcast

Provides for the approval of an existing and ready broadcast. Broadcast approval is a necessary step toward the deployment of an email/message.

ChangeBroadcastType

Provides for the ability to change whether a previously created broadcast is meant to be a broadcast or transactional message (a broadcast or message). Since the system treats broadcasts and messages in much the same way a broadcast can generally be converted to a transactional message and vice versa.

CloneBroadcast

Allows for the cloning (or copying) of any existing broadcast. Cloning is used to copy a previously deployed broadcast or copy a broadcast template to be deployed in the future. The cloning of a broadcast can be limited to simply cloning the setup parameters, the lists, the content, or any combination of these options. This function simply returns the cloned broadcast's ID.

CloneBroadcast_Supp

Identical to the CloneBroadcast web service (including the integer return type) but allows for modifying the data preparation options in the cloned broadcast. The cloning of a



broadcast can be limited to simply cloning the setup parameters; the email lists the one time suppression lists, the content, or any combination of these options. In addition to the above selectable parameters the user can choose to use or not use the original broadcast's one time suppression list with the cloned broadcast.

CloneBroadcastInFolderWithMsg

Identical to the CloneBroadcast web service with the exception that it accepts one additional parameter representing the folder ID in which to create the new broadcast.

CloneBroadcastWithMsg

Identical to the CloneBroadcast web service with the exception of the return data type. This web service will provide the calling application with any resulting error messages rather than simply the broadcast ID.

CreateBroadcast

Allows for the creation of a new broadcast for setup and delivery.

CreateBroadcastWithAutoResponse

Identical to the CreateBroadcast web service with the exception that this method allows the user to setup an auto-response message for this broadcast. Auto-reply messages are used to respond to any recipient that "replies" to the original broadcast message. This is often used to inform recipients who replied to the original message that the mailbox is unmonitored or provide additional customer support instructions.

CreateBroadcastWithMsg

Identical to the CreateBroadcast web service with the exception of the return data type. This web service will provide the calling application with any resulting error messages rather than simply the broadcast ID.

CreateBroadcastWithMsgandProjectID

Identical to the CreateBroadcastwithMsg web service with an additional parameter that allows the user to set the project ID attribute of the broadcast.

CreateBroadcastwithProjectID

Identical to the CreateBroadcast web service. This method allows the user to set the Project ID for this broadcast along with the other broadcast attributes.

CreateListFromTable

Used for internal purposes only. This method must be passed the physical table name used within the QuickPivot data schema. It allows other processes within QuickPivot to refer to an existing table within the client's database and use that as the basis for the list.

CreateListFromTableWithPrepFlags

Used for internal purposes only. Identical to CreateListFromTable method but provides the ability to change the data preparation flags (suppression flags) used when prepping the email recipient list data in the table.

CreateListFromTableWithType

Used for internal purposes only. Identical to CreateListFromTable method but allows assigning the type of list the table will represent in the broadcast (email list, one time suppression list, proof list, seed list).

CreateLiteSender

Creates a new lite sender in the account identified in the calling arguments. Lite Senders provide the ability to capture "brand" specific optoffs and receipt preferences without requiring IP and domain setups.

CreateNewBroadcast

Identical to the CreateBroadcast web service with the exception that this method allows the user to set ALL of the various parameters associated with a new broadcast such as splits, project ID, and auto reply within a single web service call.

CreateRecipientList

Obsolete

CreateSender

Creates a new sender in the account. Senders are used to capture "granular" level optoffs and can have domains and IP address ranges associated with them that are different from the overall account settings.

CreateSplitBroadcastWithMsg

Creates a new split type broadcast. The call is essentially the same as CreateBroadcastwithMsg but instead of a standard broadcast this method creates a split type broadcast.

QuickPivot

33 Arch Street
9th Floor
Boston, MA 02110
quickpivot.com
617.880.4000

CreatPostBroadcastNotification

Creates an entry for which a post-broadcast notification email message is sent. Post broadcast notifications are emails sent to one or more recipients after a broadcast has completed sending. Post Broadcast notifications typically will have links to standard broadcast reports and optoff list downloads.

EndRecipientList

Obsolete

GetActivityDet

Allows calling programs to get broadcast details for jobs that have been submitted and sent. Generally this call would be made once a job status is set to DONE. The web service returns an XML string containing the details for all of the records where there was activity. The web service will return details for any recipient who opened the email, clicked a tracked URL, opted off the broadcast, or whose email bounced. The XML string will have a number of records equal to the sum of HTMLCount, BounceCount, OptOffCount. The XML string will contain all of the fields that were associated with an individual recipient in the original list upload. Therefore all user data will be included in this details report.

GetAlleBJobs

This method retrieves the job/broadcast IDs for all the Message Manager broadcasts associated with a specific account and sender.

GetAOLContent

Retrieves the AOL content associated with a broadcast. This call can be made either pre or postbroadcast and will the AOL content that is saved in the broadcast. It will not return substituted values should merge fields/tokens be present. It will contain the track link tokens for links that are to be tracked as well as substituted Content Token values.

GetAOLSplitContent

Retrieves the AOL content associated with a specific split number for a broadcast. This call can be made either pre or postbroadcast and will provide back to the calling program the AOL content that is saved in the broadcast. It will not return substituted values should merge fields be present. It will contain the track link tokens for links that are to be tracked as well as substituted Content Token values.

GetAppTransferURL

Retrieves a URL that when clicked automatically logs a user into a specific application module. This function is used for applications requiring "single sign-on". It allows the



calling application to transmit the username, password, and application module and it returns a valid URL that redirects to the QuickPivot module without requiring the user to log in.

GetBCCounts

Obsolete - Identical to the Report Manager web service of the same name

GetBounceCount

Obsolete - Identical to the Report Manager web service of the same name

GetBounceDet

Obsolete - Identical to the Report Manager web service of the same name

GetBounceDetByDate

Obsolete - Identical to the Report Manager web service of the same name

GetBounceDetBySender

Obsolete - Identical to the Report Manager web service of the same name

GetBroadcastID

Retrieves the QuickPivot broadcast ID associated with a specific Job ID. For certain types of split broadcasts the broadcast ID may be different from the Job ID sent by the system backend processor.

GetBroadcastsByDate

Obsolete - Identical to the Report Manager web service of the same name

GetBroadcastsByFolder

Retrieves all the broadcasts, their status, the description, and campaign ID saved within a specific folder.

GetBSDetailsByDate

Retrieves all the broadcast details such as broadcast ID, Broadcast description, broadcast campaign ID, status, and other setup details for all broadcasts sent within a given date range.



GetCampaignInfo

Retrieves the Campaign information of an existing broadcast. The web service enables either pre or post broadcast identification of the Campaign information.

GetContentFromCM

Inserts the content for the broadcast from saved Content Manager content. This method is used to insert "template" content into a broadcast and allow saved content to be reused across several broadcasts.

GetCreativeSent (located in Auxilliary Email Web Services)

Allows the calling application to receive the exact HTML content, including all token and zone/layer substitutions, for a given broadcast and email address. The HTML is returned as a string of HTML. In this case the broadcast must be in a DONE status and the email address entered must have received the broadcast

GetDeliveryDate

Retrieves the Delivery Date information of an existing broadcast. The web service enables either pre or post broadcast identification of the Delivery Date information.

GeteBJobID

Obsolete

GeteBJobsByDate

Jobs that have been submitted and are in the DONE status within a given date range. Used to retrieve all broadcast IDs sent in a given time range.

GetEmailList

Allows the retrieval of the email list or lists associated with a broadcast. This call can be made either pre or post broadcast and will provide an XML string containing all the names uploaded as part of that broadcast's email recipient lists. The function does not provide information regarding which list the recipient name came from but rather a consolidated list of names uploaded as part of the individual email list(s).

GetF2FContent

Retrieves the Forward to Friend (F2F) content associated with a broadcast. This call can be made either pre or post broadcast and will provide the F2F content that is saved in the broadcast. It will not return substituted values should merge fields be present. It will contain the track link tokens for links that are to be tracked as well as substituted Content Token values.

QuickPivot

33 Arch Street
9th Floor
Boston, MA 02110
quickpivot.com
617.880.4000

GetFromInfo

Retrieves the From information of an existing broadcast. The web service enables either pre or post broadcast identification of the From information.

GetHTMLContent

Retrieves the HTML content associated with a broadcast. This call can be made either pre or post broadcast and will provide the HTML content that is saved in the broadcast. It will not return substituted values should merge fields be present. It will contain the track link tokens for links that are to be tracked as well as substituted Content Token values.

GetHTMLCount

Obsolete - Identical to the Report Manager web service of the same name

GetHTMLDet

Obsolete - Identical to the Report Manager web service of the same name

GetHTMLSplitContent

Retrieves the HTML content associated with a specific split number for a broadcast. This call can be made either pre or post broadcast and will provide the HTML content that is saved in the broadcast. It will not return substituted values should merge fields be present. It will contain the track link tokens for links that are to be tracked as well as substituted Content Token values.

GetLastProofStatus

Retrieves the status of the last proof broadcast sent for this broadcast. Allows calling applications to know if the proof is still pending (NEW), was sent successfully (DONE), or errored (ERROR)

GetLinksTracked

Obsolete - Identical to the Report Manager web service of the same name

GetListCount

Retrieves the number of records in a specific list type associated with a broadcast. This call can be made either pre or post broadcast and returns the number of records in either the proof list, email list, one time suppression list associated with a broadcast.

GetMobileContent

Retrieves the Mobile content associated with a broadcast. This call can be made either pre or post broadcast and will provides the Mobile content that is saved in the

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000

© Copyright 2014



broadcast. It will not return substituted values should merge fields be present. It will contain the track link tokens for links that are to be tracked as well as substituted Content Token values.

GetOODet

Obsolete - Identical to the Report Manager web service of the same name

GetOptOffCount

Obsolete - Identical to the Report Manager web service of the same name

GetOptOffList

Returns the list of individuals who chose to OptOff from the broadcast. This method is identical to the method of the same name in the Reporting Web Service.

GetPrepReportXML

Returns an XML string containing all the categories used for list suppression for this broadcast and the number of records that were suppressed due to this category of suppression. For example if the broadcast had 23 duplicate email addresses in its uploaded lists you would see the <DuplicateEmail> category with a 23 in the InnerText portion of the tag.

GetPrepStatus

Checks the status of a broadcast or message that is in the process of being created and returns the status of the list prepping process. The list prepping process is the process by which the uploaded lists are “scrubbed” of names that have malformed or blank email addresses, opt offs are removed, domains may be suppressed, and other name removal actions are accomplished. As a result, the process of prepping a list is a necessary step in preparing a broadcast. Allows the calling program to check on the status of the prepping process for a particular job.

GetPreview (located in Auxilliary Email Web Services)

Allows the calling application to receive a preview HTML content, including all token and zone/layer substitutions, for a given broadcast given and XML record representing the data for a recipient.

GetProcessedDet

Obsolete - Identical to the Report Manager web service of the same name.

GetProofList

Retrieves the proof list or lists associated with a broadcast. This call can be made either pre or postbroadcast and will provide an XML string containing all the names uploaded



as part of that broadcast's proof lists. The function does not provide information regarding which list the proof name came from but rather a consolidated list of names uploaded as part of the individual proof list(s). The web service returns an XML string containing whatever fields and information was part of the original proof lists that were uploaded.

GetReplyToInfo

Retrieves the Reply To information of an existing broadcast. The web service enables either pre or post broadcast identification of the Reply To information.

GetSender

Returns the Email Manager sender that is associated with an existing Email Manager job.

GetSharedContent

Retrieves the Sharing content associated with a broadcast. This call can be made either pre or postbroadcast and will provide the Sharing content that is saved in the broadcast. It will not return substituted values should merge fields be present. It will contain the track link tokens for links that are to be tracked as well as substituted Content Token values.

GetSPAMScore

Retrieves the results of a previously submitted SPAM Test. The web service provides the results of the test in a single string variable. Should details behind the score be required users must log into Email Manager for the details.

GetStatus

Returns the status of a broadcast or message that is in the process of being created and returns a list of the required elements that are missing prior to the broadcast being sent. The web service returns a comma separated string with the remaining steps. Examples of what could be returned are: "Lists,Content,Proofs,Approve,Submit" for a broadcast that has only been created and no other calls have been associated with the broadcast. Another example may be "Proofs,Approve,Submit" for a broadcast that was created and currently has lists and content associated with the broadcast but no proofs have been sent, the broadcast is not approved, and the broadcast has not been submitted. A return string of "Submit" implies all steps associated with a broadcast have been completed except for the submission of the broadcast to the backend systems.

GetSubject

Retrieves the Subject of an existing broadcast. The web service enables either pre or post broadcast identification of the Subject line.

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000

© Copyright 2014



GetSuppressedNames

Retrieves the list of all the names that were removed from the email recipient lists due to the suppression process. It lists all the names that were removed along with a code for the reason why the name was removed. This call can be made either pre or postbroadcast and will provide an XML string containing all the names removed as part of that broadcast's suppression and data prepping process.

GetSuppressionList

Retrieves the one time suppression list or lists associated with a broadcast. This call can be made either pre or postbroadcast and will provide an XML string containing all the names uploaded as part of that broadcast's one time suppression lists. The function does not provide information regarding which list the name came from but rather a consolidated list of names uploaded as part of the individual one time suppression list(s).

GetTextContent

Retrieves the Text content associated with a broadcast. This call can be made either pre or postbroadcast and will provide the Text content that is saved in the broadcast. It will not return substituted values should merge fields be present. It will contain the track link tokens for links that are to be tracked as well as substituted Content Token values.

GetTextSplitContent

Retrieves the TEXT content associated with a specific split number for a broadcast. This call can be made either pre or postbroadcast and will provide the TEXT content that is saved in the broadcast. It will not return substituted values should merge fields be present. It will contain the track link tokens for links that are to be tracked as well as substituted Content Token values.

GetTokensUsed

Retrieves a listing of all tokens and their type (system or list) used within the content/creative of a broadcast.

GetTotalClicksCount

Obsolete - Identical to the Report Manager web service of the same name

GetTrackCount

Obsolete - Identical to the Report Manager web service of the same name

GetTrackDet

Obsolete - Identical to the Report Manager web service of the same name

GetUniqueClicksCount

Obsolete - Identical to the Report Manager web service of the same name

GetViewAsWebPageContent

Retrieves the View as Web Page content associated with a broadcast. This call can be made either pre or postbroadcast and will return the View as Web Page content that is saved in the broadcast. It will not return substituted values should merge fields be present. It will contain the track link tokens for links that are to be tracked as well as substituted Content Token values.

LoadContent

Inputs and saves the content/creative for any given identified broadcast.

LoadContentTokens

Loads HTML and/or text content as replacement fields in an existing broadcast. Content Tokens represent "mail-merge" type fields that can be replaced across an entire broadcast. The web service provides a programmatic way to replace large sections of content for a specific broadcast.

LoadEncodedContent

Provides the means by which non-ansi encoded content can be loaded into an existing broadcast. This is useful for special applications or double-byte language support in messaging.

LoadF2FContent

Loads the Forward to Friend (F2F) content for any given identified broadcast. This method allows marketers to load different content for use in messages forwarded using the Forward to Friend functionality in the tool. Using this feature provides a means to deliver alternate content than what was sent to the original recipient.

LoadList

Allows for the loading of lists of recipients for a broadcast. The list can be a list of recipient names, proof names, seed names, or optoff/suppression names. The web service allows the calling program to send the list in a comma separated or XML data stream as part of the web service call. This function is appropriate to use for lists not exceeding 5000 names.

LoadListFile

Allows for the loading of lists for a broadcast. The list can be a list of recipient names, proof names, seed names, or optoff/suppression names. The web service allows the

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000



calling program to send the FTP location and login credentials of where the list can be found. This function is appropriate to use for any size list that is already in a file format.

LoadListFileWithPrepFlags

Identical to LoadListFromFile with the exception that the data preparation flags (suppression flags) used when prepping the email recipient list data in the file can be altered from the default values that are saved for the account.

LoadListWithPrepFlags

Identical to LoadList with the exception that the data preparation flags (suppression flags) used when prepping the email recipient list data in the file can be altered from the default values that are saved for the account.

LoadLMList

Allows the use of a table previously saved in Data Manager to be used within a broadcast. This method provides for the reuse of tables/lists that have been uploaded and saved within the Data Manager application and the association of such lists with existing broadcasts.

LoadLMListWithFlags

Identical to LoadLMList with the exception that the data preparation flags (suppression flags) used when prepping the email recipient list data in the file can be altered from the default values that are saved for the account.

LoadLMSegment

Allows the use of a segment/query that was previously saved in Data Manager to be used within a broadcast. This method provides for the reuse of queries that have been created and saved within the Data Manager application and the association of such resultant lists with existing broadcasts.

LoadLMSegmentWithFlags

Identical to LoadLMSegment with the exception that the data preparation flags (suppression flags) used when prepping the email recipient list data in the file can be altered from the default values that are saved for the account.

LoadMobileContent

Loads the Mobile content for any given identified broadcast. This method allows marketers to load different content for use in messages where the recipient clicks the View on Mobile device functionality in the tool. Using this feature provides a means to deliver alternate content than what was sent to the original recipient.



LoadSharedContent

Loads the Shared content for any given identified broadcast. This method allows marketers to load different content for use in messages that are Shared to a social media site from within the message using the Share to Social functionality in the tool. Using this feature provides a means to deliver alternate content than what was sent to the original recipient. This is particularly useful when wanting to remove specific personalization from a message to encourage the user to share content to their Social Media sites.

LoadSplitContent

This web service inputs and saves the content for any given identified broadcast for a specific split number associated with the broadcast. Splits represent A/B testing broadcasts that may or may not include holdbacks.

LoadUpLoadedList

Allows for the loading of a list that has already been transmitted into the system. This allows lists that have already been uploaded into the QuickPivot system using one of the List FTP functions such as TransferFileFromFTP. The TransferFileFromFTP web service simply moves files from external FTP locations to an internal client location. Files that have been transferred can now be used in broadcasts or loaded into the Data Manager application. This method allows such lists to be associated with a broadcast.

LoadUpLoadedListWithPrepFlags

Identical to LoadUpLoadedList with the exception that the data preparation flags (suppression flags) used when prepping the email recipient list data in the file can be altered from the default values that are saved for the account.

LoadViewAsWebPageContent

Loads the View as Web Page content for any given identified broadcast. This method allows marketers to load different content that will be displayed in a user's browser when they select the View as Web Page link in their original message. Using this feature provides a means to deliver alternate content than what was sent to the original recipient. This is particularly useful when wanting to use the full functionality of the web in the marketing message. Scripting, controls, and "active" content that would typically be blocked by email clients can be enabled on the View as Web Page content and allow a more full featured user experience.

ModifyCampaignInfo

Allows for the modification of the campaign information of an existing broadcast. This function can be particularly useful when Cloning a previously delivered broadcast.



ModifyDeliveryDate

Allows for the modification of delivery date information of an existing broadcast. This function can be particularly useful when Cloning a previously delivered broadcast.

ModifyFromInfo

Allows for the modification of the From information of an existing broadcast. This function can be particularly useful when Cloning a previously delivered broadcast.

ModifyPrepOptions

Allows for the modification of the list suppression or "prep flags" for a specific broadcast. This function can be particularly useful when Cloning a previously delivered broadcast.

ModifyReplyToInfo

Allows for the modification of the Reply To information of an existing broadcast. This function can be particularly useful when Cloning a previously delivered broadcast.

ModifySettings

Allows for the modification of any of the setup parameters of an existing broadcast. This function can be particularly useful when Cloning a previously delivered broadcast.

ModifySplitSettings

Allows for the modification of any of the setup parameters of an existing split broadcast. This function can be particularly useful when Cloning a previously delivered broadcast.

ModifySplitSubject

Allows for the modification of the subject line in any one of the numbered splits. This function can be particularly useful when Cloning a previously delivered broadcast.

ModifySubject

Allows for the modification of subject line of an existing broadcast. This function can be particularly useful when Cloning a previously delivered broadcast.

ProofBroadcast

Provides for the sending of proofs/tests of a broadcast to an individual email address or a list of names.

ProofBroadcastToList

Provides for the sending of proofs/tests of a broadcast to the names on an uploaded Proof list.



ProofBroadcastWithZones

Provides for the sending of proofs/tests of a broadcast substituting specific Zones and Layers to be included in the proof/test to an individual email address or a list of names.

ReprepList

Submits the lists associated with the specified broadcast to the list prepping process. This can be used to rerun suppressions to keep broadcasts lists up to date.

SendPrepReportEmail

Sends the Prep Report associated with a specific broadcast to one or more individuals. Prep reports list all the categories used in the suppression process and the counts of email addresses that were removed from the broadcast due to that category.

SetSocialSharing

Allows the calling application to turn on/off the various Social Sharing site icons within the Sharing Token replacement. The method accepts the broadcast ID and a string of XML that contains the names of the Sharing Sites (e.g. Facebook, MySpace, StubbleUpon, etc.). For a list of social site supported by the Sharing Token (@[@Sharing]) please consult the Email Manager application.

SetSocialSharingWithDesc

Allows the calling application to turn on/off the various Social Sharing site icons within the Sharing Token replacement. The method accepts the broadcast ID and a string of XML that contains the names of the Sharing Sites (e.g. Facebook, MySpace, StubbleUpon, etc.). This method allows the calling application to also specifically set the Title and Description parameters to be used by the Sharing Sites. In addition, this method allows the setting of specific sharing sites, titles, and descriptions for individual split pieces of content. For a list of social site supported by the Sharing Token (@[@Sharing]) please consult the Email Manager application.

SPAMTest

Submits the broadcast to the SPAM Assassin application and returns a score. The web service allows for testing of SPAM scores and ensures deliverability decisions can be programmed into the calling application's logic.

SubmitBroadcast

Provides for the submission of an existing and ready broadcast.

SubmitBroadcastWithFlags

Provides for the submission of an existing and ready broadcast with values for enabling and setting throttling, enabling and setting frequency suppression flags, and enabling and setting list sorting.

SubmitRPTest

Provides for the submission of an existing broadcast to the Return Path test list that has been assigned and designated for this account as part of the account setup.

TestAutoEB

Provides a “heartbeat” function for calling applications to check the status of the web service. It accepts no parameters. Upon invoking the TestAutoEB function the user will receive a success or failure. Success is measured when the system successfully connects to the Message Manager backend database and the overall system master database.

TrackAllLinks

Allows the user to insert link tracking tokens into uploaded content. This web service is called AFTER content has already been loaded and associated with a broadcast (i.e. LoadContent, LoadContentTokens). The call is nondiscriminate in that it will insert link tracking for all links (e.g. href=http://....) in ALL forms of uploaded content (i.e. HTML, text, Sharing, Mobile, F2F, View as Web Page).

UploadImages

Allows calling programs to send a binary stream of images into the system for hosting. These images can then be locally referenced in the uploaded creatives/content. The image stream contains identifying information about the image file and the binary stream representing the image data

SetSocialSharing (located in Auxilliary Email Web Services)

Turns on/off the various Social Sharing site icons that get displayed within the Sharing Token replacement. The method accepts the broadcast ID and a string of XML that contains the names of the Sharing Sites (e.g. Facebook, MySpace, StubbleUpon, etc.). For a list of social site supported by the Sharing Token (@[@Sharing]) please consult the Message Manager application.

SetSocialSharingWithDesc (located in Auxilliary Email Web Services)

Turns on/off the various Social Sharing site icons within the Sharing Token replacement. The method accepts the broadcast ID and a string of XML that contains the names of the Sharing Sites (e.g. Facebook, MySpace, StubbleUpon, etc.). This method allows the calling application to also specifically set the Title and Description parameters to be used by the Sharing Sites. In addition, this method allows the setting of specific sharing

QuickPivot

33 Arch Street
9th Floor
Boston, MA 02110
quickpivot.com
617.880.4000



sites, titles, and descriptions for individual split pieces of content. For a list of social site supported by the Sharing Token (@[@Sharing]) please consult the Email Manager application.

QuickPivot
33 Arch Street
9th Floor
Boston, MA 02110
quickpivot.com
617.880.4000

© Copyright 2014



DATA MANAGER WEB METHODS

AddTableColumns

Add columns to a specific table/list stored in Data Manager. Note that new columns are assumed to be Varchar (Character String) column types

AppendList

Appends/updates to a table/list that already exists within the Data Manager application. String or XML data records are added to the bottom of the previously existing list. This is used primarily when adding records that are being gathered through the use of a form hosted at an external site. The status of this upload request can be checked by calling the CheckUpdateOrAppendOrReplaceListRequestStatus

AppendListFromFTP

Appends/updates to a table/list that already exists within the Data Manager application. Data files located on an external FTP site are uploaded and each line in the file is inserted as a row into the Data Manager destination table. The status of this upload request can be checked by calling the CheckUpdateOrAppendOrReplaceListRequestStatus

CheckExportListFTPRequestStatus

Allows users to check the status of a previously requested list export via FTP. This function is typically used within a loop which continuously checks the export request until the return value of this function is "Completed".

CheckExportListStatus

Allows users to check the status of a previously requested list export that is delivered into the User Interface. This function is typically used within a loop which continuously checks the export request until the return value of this function is "Completed".

CheckGenerateListRequestStatus

Allows users to check the status of a previously requested list generation command. List generation is often associated with the running of a query, set, or export. This function is typically used within a loop which continuously checks the request until the return value of this function is "Available".

CheckGenerateSplitListRequestStatus

Allows users to check the status of a previously requested list split command. List splits are often associated with the generation of 2 or more resultant tables that contain recipients either randomly chosen from the original table or simply splitting by

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000

© Copyright 2014



percentage. This function is typically used within a loop which continuously checks the request until the return value of this function is "Available".

CheckNewListRequestStatus

Allows users to check the status of a previously requested list upload. The Data Manager system requires a finite amount of time to both upload and import new lists before they are available, as tables, for other Data Manager operations. This function is typically used within a loop which continuously checks the request until the return value of this function is "Available".

CheckUpdateOrAppendOrReplaceListRequestStatus

Allows users to check the status of a previously requested List operation (Update, Append, Replace). The Data Manager system requires a finite amount of time to both upload, append, replace, and import new lists before they are available, as tables, for other Data Manager operations. This function is typically used within a loop which continuously checks the request until the return value of this function is "Available".

CreateSegment

Provides the ability to define new segments/queries for a specific table. The segment definition "language" is an XML formatted string that provides pseudo-SQL which is translated and turned into a query.

DeleteList

Provides the ability to delete a list from Data Manager

ExportList

Returns the requested list/table as an XML string.

ExportListToFTP

Returns the requested list/table as a formatted file and placed on the FTP location identified in the method parameters.

ExportSegment

Returns the list/table which is generated by running the input segment as an XML string.

ExportSegmentToFTP

Returns the list/table which is generated by running the segment as a formatted file and placed on the input FTP location.

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000

© Copyright 2014

ExportSet

Returns the list/table which is generated by running the input set as an XML string.

ExportSetToFTP

Returns the list/table as an XML string.

GenerateList

Returns the list/table which is generated by running the segment as a formatted file and placed on the input FTP location.

GenerateSplitList

Generates a set of lists/tables from a single list. Allows naming the splits, setting the total number of splits, identifying the method (random or sequential) of splitting the main list/table, and selection of the destination Data Manager folder. This is equivalent to generating a split list from using the RightClick option in the Data Manager interface.

GetFolderContent

Provides a listing of all the Data Manager items in a folder.

GetFolderDetails

Provides the attributes of a folder in an XML string format. Attributes such as folder name, retention policy, ownership, parent folder ID, and security are passed as name value pairs in a returned string.

GetFolderIDsByName

Provides an XML listing of all the Folder IDs that correspond to the input folder name. Since name uniqueness is not enforced in Data Manager, a folder name may be associated with several unique IDs and therefore when retrieving a folder ID from a given name the function returns a list of folder IDs that have the corresponding input parameter name. This allows the user to programmatically identify the folder for use in subsequent web service calls.

GetFolderTree

Provides an XML representation of the entire folder tree that is visible to the user.

GetListColumnDetails

Provides an XML representation of the columns that are in the list/table. It provides information such as the table's column name and data type.

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000

GetListDetails

Provides the details and attributes associated with a given list/data in an XML representation. The method will return the columns in the table, the list/table name, the containing folder name, the list/table owner, and the access rights of the list/table.

GetListIDByName

Provides the list ID for a specific list/table based on its name. Since name uniqueness is not enforced in Data Manager a list name may be associated with several unique IDs and therefore when retrieving a list ID from a given name the function returns a list of list IDs that have the corresponding input parameter name.

GetListPrepStatus

Checks the status of a Data Manager Prep List Request. The list prepping process is the process by which the uploaded lists are “scrubbed” of names that have malformed or blank email addresses, opt offs are removed, domains may be suppressed, and other name removal actions are accomplished. Allows the calling program to check on the status of the prepping process for a particular table.

GetPrepReportXml

Returns an XML string containing all the categories used for list suppression for this broadcast and the number of records that were suppressed due to this category of suppression. For example if the broadcast had 23 duplicate email addresses in its uploaded lists you would see the <DuplicateEmail> category with a 23 in the InnerText portion of the tag.

GetSegmentDetails

Provides the details and attributes associated with a given Segment in an XML representation. The method will return the attributes such as ownership & access, parent folder ID, and name of the Segment. The specific columns requested, filter field name and value, list used for Segment, and approximate count will also be part of the returned information.

GetSegmentIDByName

Provides the Segment ID for a specific Segment based on its name. Since name uniqueness is not enforced in Data Manager a Segment name may be associated with several unique IDs and therefore when retrieving a Segment ID from a given name the function returns a list of Segment IDs that have the corresponding input parameter name.

GetSegmentParameterList

Provides an XML representation of the Segment. It provides information such as the list name, the field name, operator, and the value for the Segment. The method retrieves the



Segment XML which can then be modified and then submitted back into the Data Manager and run against a different table to retrieve a result.

GetSetDetails

Provides the details and attributes associated with a given set in an XML representation. The method will return the attributes such as ownership & access, parent folder ID, and name of the set. The specific columns requested, filter field name and value, list used for set, and approximate count will also be part of the returned information.

GetSetIDByName

Provides the Set ID for a specific Set based on its name. Since name uniqueness is not enforced in Data Manager a Set name may be associated with several unique IDs and therefore when retrieving a Set ID from a given name the function returns a list of Set IDs that have the corresponding input parameter name.

GetSetParameterList

Provides an XML representation of the Set. It provides information such as the list name, the field name, operator, and the value for the segment. The method retrieves the Set XML which can then be modified and then submitted back into the Data Manager and run against a different table to retrieve a result.

LoadLargeUploadList

Allows streaming a large list into Data Manager. When not using the FTP methods for getting lists into Data Manager, this method provides a streaming approach so that data packets can be transferred without data loss for large (> 10,000 rows) lists.

LoadUploadedList

Allows for the loading of a list that has already been transmitted into the system. This allows lists that are already available in the system through the use of the FTP web service. The web service allows the calling program to input the list name and type. The web service also accepts parameters that govern the behavior of suppressions, optoffs, and deduping functions that are run against the list.

OptIn

Allows removing bounce-out or optout flags for an email address. This is often used when an email address is inadvertently bounced-out due to an ISP issue and the email address needs to be "re-enabled". This method can also be used when an email address is "remarketed".

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000

© Copyright 2014

OptOut

Allows setting the optout flag for a specific email address in a specific account and for a specific sender. This is used to programmatically insert optoffs that are being gathered through a third party.

PrepList

Allows running a list suppression against a list/table loaded in the Data Manager application.

ReplaceList

Allows replacing a list that was previously loaded in Data Manager. This is useful when segments, workflows, transactional messages, and sets are referencing a list by its ID and you wish to do a periodic data refresh.

ReplaceListFromFTP

Allows replacing a list that was previously loaded in Data Manager using FTP transfer. This is useful when segments, workflows, transactional messages, and sets are referencing a list by its ID and you wish to do a periodic data refresh.

Resubscribe

Allows resubscribing an email address that had previously opted out or bounced out. This method is different from the OptIn method such that this method clears the optout and bounce out flags for the email address AND removes the email address from all Data Manager maintained suppressions lists. This method will not clear optout flags if the recipient had registered a complaint OR if the user has a valid optout registered AFTER the date passed in the resubscribe call.

SendPrepReportEmail

Sends the Prep Report associated with a specific List Prepping to one or more individuals. Prep reports list all the categories used in the suppression process and the counts of email addresses that were removed from the broadcast due to that category.

UpdateList

Allows updating the values or rows in an existing Data Manager list. Updates can perform 2 different operations, an append OR an update. The update method utilizes the KEY column that had been identified in the data table and uses that to determine whether an append or update to the table occurs. If the KEY already exists in the table then an update occurs otherwise the record gets appended to the bottom of the table. The status of this upload request can be checked by calling the CheckUpdateOrAppendOrReplaceListRequestStatus method.

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000

© Copyright 2014



UpdateListFromFTP

Allows updating the values or rows in an existing Data Manager list using a list uploaded from an FTP site. Updates can perform 2 different operations, an append OR an update. The update method utilizes the KEY column that had been identified in the data table and uses that to determine whether an append or update to the table occurs. If the KEY already exists in the table then an update occurs otherwise the record gets appended to the bottom of the table. The status of this upload request can be checked by calling the CheckUpdateOrAppendOrReplaceListRequestStatus method.

UploadNewList

Allows for the loading of a list into a target folder in Data Manager. The web service allows sending the list in a comma separated or XML data stream as part of the web service call. This function is appropriate to use for lists not exceeding 10,000 names. The status of this upload request can be checked by calling the CheckUpdateOrAppendOrReplaceListRequestStatus method.

UploadNewListFromFTP

Allows for the loading of a list into a target folder in Data Manager. The web service allows the picking up of a list/data file from an FTP location and loads this file into a table within Data Manager. The status of this upload request can be checked by calling the CheckUpdateOrAppendOrReplaceListRequestStatus method.

UploadZipFileFromFTP

Allows for the loading of a list into a target folder in Data Manager. The web service allows sending the list in a comma separated or XML data file that is zipped and placed on an FTP site. The status of this upload request can be checked by calling the CheckUpdateOrAppendOrReplaceListRequestStatus method.

ViewList

Returns the entire list contents of an existing Data Manager list.



CONTENT MANAGER WEB METHODS

CreateFolder

Creates a new folder in the application.

CreateNewHTMLContent

Adds new HTML content to Content Manager by sending the HTML as a string parameter in the web service.

CreateNewHTMLContentFromFTP

Adds new HTML content to Content Manager by retrieving the HTML as a file on an FTP site.

CreateNewTextContent

Adds new Text content to Content Manager by sending the Text as a string parameter in the web service.

CreateNewTEXTContentFromFTP

Adds new Text content to Content Manager by retrieving the Text as a file on an FTP site.

GetContentDetails

Returns all the attributes of a piece of content saved in Content Manager. Attributes such as name, folder, size, last modified by and date, created by and date, and content type are returned in an XML format.

GetContentIDsByName

Returns an XML listing of all the Content IDs that correspond to the input name. Since name uniqueness is not enforced in Content Manager a piece of content name may be associated with several unique IDs and therefore when retrieving a Content ID from a given name the function returns a list of Content IDs that have the corresponding input parameter name.

GetFolderContent

Returns a listing of all the List Manager items in a folder. The method accepts a folder ID as its primary input. This is generally used to retrieve a list of all items in a folder and then loop through the items to examine attributes OR make changes and re-upload.

GetFolderDetails

Returns the attributes of a folder in an XML string. Attributes such as folder name, retention policy, ownership, parent folder ID, and security are passed as name value pairs and returned in an XML string.

GetFolderIDsByName

Returns an XML listing of all the Folder IDs that correspond to the input name. Since name uniqueness is not enforced in Content Manager a folder name may be associated with several unique IDs and therefore when retrieving a folder ID from a given name the function returns a list of folder IDs that have the corresponding input parameter name.

GetFolderTree

Returns an XML representation of the entire folder tree that is visible to the user. The returned XML lists each folder ID and name. The return will include all folders even those folders that are child folders.

GetFormDetails

Returns the attributes for a form that was created in Content Manager in an XML string. The attributes can then be used to make modifications to the form or to programmatically create an input routine that feeds the form.

GetFormIDsByName

Returns with an XML listing of all the Form IDs that correspond to the input name. Since name uniqueness is not enforced in Content Manager a Form name may be associated with several unique IDs and therefore when retrieving a Form ID from a given name the function returns a list of Form IDs that have the corresponding input parameter name.

GetLayerDetails

Returns the attributes for a Layer that was created in Content Manager in an XML string. The attributes returned include the HTML & text content, the conditions associated with the layer, the evaluation order, and the default flag. This can be used to query the parameters needed to successfully “fire” the condition on this layer.

GetLayersByZone

Returns the Layer IDs loaded into a Zone.

GetReferencedZones

Returns all the zones in which a specific piece of Content Manager content is used as a layer.

GetResourceDetails

Returns the attributes for a Resource that was created in Content Manager in an XML string.

GetResourceIDsByName

Returns an XML listing of all the Resource IDs that correspond to the input name. Since name uniqueness is not enforced in Content Manager a Resource name may be associated with several unique IDs and therefore when retrieving a Resource ID from a given name the function returns a list of Resource IDs that have the corresponding input parameter name.

GetZoneDetails

Returns the attributes for a Zone that was created in Content Manager in an XML string. The attributes returned include name, list ID associated with the zone, folder, and layer details. This method can be used to obtain the list ID used in a zone and then call the Data Manager web service to obtain the details and structure of the data table.

GetZoneIDsByName

Returns an XML listing of all the Zone IDs that correspond to the input name. Since name uniqueness is not enforced in Content Manager a Zone name may be associated with several unique IDs and therefore when retrieving a Zone ID from a given name the function returns a list of Zone IDs that have the corresponding input parameter name.

PublishContent

Publishes a Form that was previously saved in Content Manager.

UploadNewResourceFromFTP

Adds a new Resource to Content Manager by retrieving the Resource as a file on an FTP site. Resources can be any “hostable” piece of web content such as images and/or documents.

UploadResourceToFTP

Downloads a Resource from Content Manager by placing the Resource on an FTP site.

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000

© Copyright 2014



REPORT MANAGER WEB METHODS

GetActivityDet

Returns the broadcast details for jobs that have been submitted and sent. Generally this call would be made when a job status is "DONE". The web service returns an XML string containing the details for all of the records where there was activity. The web service will return details for any recipient who opened the email, clicked a tracked URL, opted off the broadcast, or whose email bounced. The XML string will have a number of records equal to the sum of HTMLCount, BounceCount, OptOffCount. The XML string will contain all of the fields that were associated with an individual recipient in the original list upload. Therefore all user data will be included in this details report.

GetBCCounts

Returns the broadcast statistics for jobs that have been submitted and sent. Generally this call would be made when a job status is "DONE". The web service returns an XML string containing the count totals for Sent, OptOffs, HTML Senses (Opens), and Bounces.

GetBounceCount

Returns the broadcast statistics for jobs that have been submitted and sent. Generally this call would be made when a job status is "DONE". The web service returns an integer containing the total number of email addresses that bounced either for hard or soft reasons for the job.

GetBounceDet

Returns the broadcast details for jobs that have been submitted and sent. Generally this call would be made when a job status is "DONE". The web service returns an XML string containing the details for all of the Bounces that occurred in the job. The XML string will have a number of records equal to the total number of Bounces returned by the GetBounceCount web service. The XML string will contain all of the fields that were associated with an individual recipient in the original list upload. Therefore all user data will be included in this details report.

GetBounceDetByDate

Identical to GetBounceDet while allowing the input of a date range for which to return the results. The web service returns an XML string containing the details for all of the Bounces that occurred in the time window between an entered start and end date. The XML string will contain all of the fields that were associated with an individual recipient in the original list upload. Therefore all user data will be included in this details report.

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000

© Copyright 2014



GetBounceDetBySender

Identical to GetBounceDet while allowing the input of the sender and a date range for which to return the results. The web service returns an XML string containing the details for all of the Bounces that occurred in the time window between an entered start and end date. The XML string will contain all of the fields that were associated with an individual recipient in the original list upload. Therefore all user data will be included in this details report.

GetBroadcastsByDate

Returns the list of all the broadcasts submitted for a specific account and sender over a user specified period of time. The web service returns an XML string containing the names, descriptions, broadcast dates, status, and total records processed for the broadcasts occurring between the start date and stop date entered as part of the call.

GeteBJobsByDate

Returns the Job ID of all broadcasts that have been submitted and are in the "DONE" status within a given date range.

GetEventsByDate

Returns an XML string containing all the events associated with an account and sender for a given date range. The XML contains a record for each broadcast and broadcast's event (sent, open, clicked link, optout, bounce, etc.) containing the email address of the recipient for which the action was related along with the timestamp of the event.

GetEventsByDateDX

Internal Use Only

Returns an XML string containing all the events associated with a specific broadcast for a given date range. The XML contains a record for each event (sent, open, clicked link, optout, bounce, etc.) containing the email address of the recipient for which the action was related along with the timestamp of the event. This method is identical to the GetEventsByDate with the exception that it contains 5 specific data fields used for customer analysis.

GetHTMLCount

Returns the total number of opens (HTML Senses) for a specified broadcast. Generally this call would be made when a job status is "DONE".

GetHTMLDet

Returns an XML string containing the details for all of the HTML senses that occurred in the job. Generally this call would be made when a job status is "DONE". The XML string



will have a number of records equal to the total number of HTML Senses returned by the GetHTMLCounts web service. The XML string will contain all of the fields that were associated with an individual recipient in the original list upload. Therefore all user data will be included in this details report.

GetLinksTracked

Returns an XML string containing the total number of tracks and the total number of unique tracks recorded for EACH of the links that are tracked within the broadcast. Generally this call would be made when a job status is "DONE".

GetOODet

Returns an XML string containing the details for all of the OptOffs that occurred in the job. Generally this call would be made when a job status is "DONE". The XML string will have a number of records equal to the total number of Tracks returned by the GetOptOffCounts web service. The XML string will contain all of the fields that were associated with an individual recipient in the original list upload. Therefore all user data will be included in this details report.

GetOptOffCount

Returns an integer containing the total number of people who opted off for the job. Generally this call would be made when a job status is "DONE".

GetOptOffList

Returns the list of all the individuals who have opted off from broadcasts submitted for a specific account and sender over a user specified period of time. The web service returns an XML string containing the names, opted off between the start date and stop date entered as part of the call across all broadcasts. If one requires the list of opt off names associated with a specific broadcast it is recommended to use the GetOODet web service call.

GetProcessedDet

Returns an XML string containing the details for all of the records sent in the job. Generally this call would be made when a job status is "DONE". The XML string will have a number of records equal to the total number of records returned in the Processed variable by the GetBCCCounts web service. The XML string will contain all of the fields that were associated with an individual recipient in the original list upload. Therefore all user data will be included in this details report.

GetSMTPBcastID

Returns the broadcast ID for jobs that have been submitted via the SMTP message handler given a specific SMTP sent broadcast ID. This method allows for the cross



referencing of the users submitted broadcast ID and the system's created unique broadcast ID. The web service returns an integer containing the internal broadcast ID for a given SMTP broadcast ID.

GetTotalClicksCount

Returns an integer containing the total number of URL clicks for the job including repeat clicks of the same link. Generally this call would be made when a job status is "DONE".

GetTrackCount

Performs the same function as GetTotalClicksCount. It is included for legacy support purposes.

GetTrackDet

Returns an XML string containing the details for each of the URL Tracks that occurred in the job. Generally this call would be made when a job status is "DONE". The XML string will have a number of records equal to the total number of Tracks returned by the GetTrackCounts (or GetTotalClickCounts) web service. The XML string will contain all of the fields that were associated with an individual recipient in the original list upload. Therefore all user data will be included in this details report.

GetUniqueClicksCount

Returns an integer containing the total number of unique URL clicks for the job regardless of how many different links may have been clicked in the same email message. Generally this call would be made when a job status is "DONE". The count removes those URL clicks that occurred more than once by any of the email recipients.

TestService

Sends a test call to the reporting web service to allow application to test the responsiveness of the service prior to submitting report requests.



CAMPAIGN RESULTS EXPORT WEB METHODS

ActivityReportByDateRange

Inserts a request into the reporting backend processor to generate a standard broadcast activity report for all broadcasts sent in a given date range.

ActivityReportByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a standard broadcast activity report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

BounceReportByDateRange

Inserts a request into the reporting backend processor to generate a bounce report for all broadcasts sent in a given date range. The bounce report provides email address level data on all individuals whose email address bounced. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the bounce transaction occurred.

BounceReportByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a bounce report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

ComplaintReportByDateRange

Inserts a request into the reporting backend processor to generate a complaint report for all broadcasts sent in a given date range. The complaint report provides email address level data on all individuals who registered a complaint and the ISP that was the source of the complaint. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the complaint transaction occurred.

ComplaintReportByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a complaint report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

DownloadRequestedReport

Streams a previously requested report as a bitstream. This web service call retrieves the actual report data that had previously been requested.

F2FReportByDateRange

Inserts a request into the reporting backend processor to generate a Forward to a Friend report for all broadcasts sent in a given date range. The F2F report provides email address level data on all individuals who forwarded the message to another recipient. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the forward transaction occurred.

F2FReportByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a Forward to a Friend report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

GetReportFileName

Retrieves the filename for the report generated in a previous report request.

GetReportRequestDetails

Retrieves the detailed status for the report generated in a previous report request.

GetReportRequestStatus

Retrieves the report request status for a previous report request.

LinkAliasReportByDateRange

Inserts a request into the reporting backend processor to generate a Link Alias report for all broadcasts sent in a given date range. The link alias report provides the user with a list of all link alias names used in broadcast link tracking and the number of unique and total clicks for each.

LinkAliasReportByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a Link Alias report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

OpenReportByDateRange

Inserts a request into the reporting backend processor to generate an open report for all broadcasts sent in a given date range. The open report provides email address level data on all individuals who opened a broadcast in the given date range. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the open transaction occurred.



OpenReportByRelativeTimeframe

Inserts a request into the reporting backend processor to generate an open report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

OptoutReportByDateRange

Inserts a request into the reporting backend processor to generate an optout report for all broadcasts sent in a given date range. The optout report provides email address level data on all individuals who opted out of a broadcast in the given date range. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the optout transaction occurred.

OptoutReportByRelativeTimeframe

Inserts a request into the reporting backend processor to generate an optout report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

RawBounceDataByDateRange

Inserts a request into the reporting backend processor to generate a bounce report for all broadcasts sent in a given date range. The raw bounce report provides all the information that is captured in the backend logs for a bounce transaction. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the bounce transaction occurred.

RawBounceDataByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a raw bounce report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

RawComplaintDataByDateRange

Inserts a request into the reporting backend processor to generate a complaint report for all broadcasts sent in a given date range. The raw complaint report provides all the information that is captured in the backend logs for a complaint transaction. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the complaint transaction occurred.

RawComplaintDataByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a raw complaint report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000

© Copyright 2014

RawF2FDataByDateRange

Inserts a request into the reporting backend processor to generate a Forward to Friend report for all broadcasts sent in a given date range. The raw Forward to Friend report provides all the information that is captured in the backend logs for an F2F transaction. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the F2F transaction occurred.

RawF2FDataByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a raw F2F report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

RawOpenDataByDateRange

Inserts a request into the reporting backend processor to generate an open report for all broadcasts sent in a given date range. The raw open report provides all the information that is captured in the backend logs for an open transaction. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the open transaction occurred.

RawOpenDataByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a raw open report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

RawOptoutDataByDateRange

Inserts a request into the reporting backend processor to generate an Optout report for all broadcasts sent in a given date range. The raw optout report provides all the information that is captured in the backend logs for an optout transaction. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the optout transaction occurred.

RawOptoutDataByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a raw optout report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

RawSentDataByDateRange

Inserts a request into the reporting backend processor to generate a sent report for all broadcasts sent in a given date range. The raw sent report provides all the information that is captured in the backend logs for a sent transaction. Filters are present to allow



the user to determine if the date range should be measured from when the broadcast was sent or from when the sent transaction occurred.

RawSentDataByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a raw sent report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

RawSharingDataByDateRange

Inserts a request into the reporting backend processor to generate a sharing report for all broadcasts sent in a given date range. The raw sharing report provides all the information that is captured in the backend logs for a share transaction. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the share transaction occurred.

RawSharingDataByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a raw sharing report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

RawTrackDataByDateRange

Inserts a request into the reporting backend processor to generate a "click" report for all broadcasts sent in a given date range. The raw "click" report provides all the information that is captured in the backend logs for a "click" transaction. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the "click" transaction occurred.

RawTrackLinkDataByDateRange

Inserts a request into the reporting backend processor to generate a link tracking report for all broadcasts sent in a given date range. The raw link tracking report provides all the information that is captured in the backend logs for a tracked link transaction grouped by the individual link URL rather than the individual/recipient. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the track transaction occurred.

RawTrackLinkDataByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a raw link tracking report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

QuickPivot

33 Arch Street

9th Floor

Boston, MA 02110

quickpivot.com

617.880.4000

© Copyright 2014



RawTracksDataByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a raw “click” report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

SentReportByDateRange

Inserts a request into the reporting backend processor to generate a sent report for all broadcasts sent in a given date range. The sent report provides recipient level detail for a sent transaction. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the sent transaction occurred.

SentReportByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a sent report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

SharingReportByDateRange

Inserts a request into the reporting backend processor to generate a social sharing report for all broadcasts sent in a given date range. The social sharing report provides recipient level detail for a sent transaction. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the social sharing transaction occurred.

SharingReportByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a social sharing report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)

TrackLinkReportByDateRange

Inserts a request into the reporting backend processor to generate a link tracking report for all broadcasts sent in a given date range. The link tracking report provides track URL level detail for a link tracking transaction. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the link click transaction occurred.

TrackLinkReportByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a link tracking report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)



TrackReportByDateRange

Inserts a request into the reporting backend processor to generate a “click” report for all broadcasts sent in a given date range. The “click” report provides recipient level detail for a “click” transaction. Filters are present to allow the user to determine if the date range should be measured from when the broadcast was sent or from when the “click” transaction occurred.

TrackReportByRelativeTimeframe

Inserts a request into the reporting backend processor to generate a “click” report for all broadcasts sent in a given date range where the date range is identified as a relative timeframe (i.e. yesterday, last week, this week, etc.)



WORKFLOW MANAGER WEB METHODS

WFEvent

Sends an EVENT to a running workflow. This method requires a specific workflow identifier as well as the Instance ID for the specific person's running workflow instance.

WFStart

Injects a record into a previously Published workflow within Workflow Manager.