XML API
Developer's Guide

ENGAGE

Winter '14

February 2014

Table of Contents

About This Guide & Legal 6
Copyright, Trademark, and Distribution Notice 6
Usage Disclaimer 6
What's New in the XML API 7
Introduction 8
Requests 10
Submitting an API Request 11
API Limits 11
Responses 12
Processing Responses 14
Errors 14
Authentication with the APIs 15
Authentication Method: OAuth 2.0 Granted Access 15
Getting Started with OAuth in Engage 15
Getting Refresh Tokens 16
Getting Access Tokens 17
Legacy Authentication Method: JSESSIONID User Sessions 20
User Login 20
User Logout 21
Contact XML Interfaces 22
Mailing Management Interfaces 22
Send a Mailing to a Contact 22
Forward Mailing to a Friend 24
Get Contact Mailing Details 25
Purge Data 27
Database Management Interfaces – Contact 29
Add a Contact 30

Double Opt-in a Contact 34
Update a Contact 35
Opt-out a Contact 40
Select Contact Details 42
User XML Interfaces 46
Database Management Interfaces – User 46
Import to a Database 46
Export from a Database 57
Add a Column to a Database 60
Get Database Details 61
List Contact Mailings 63
Remove a Contact 64
Get a List of Databases 66
Create a Relational Table 69
Associate Relational Data with Contacts in a Database 71
Insert and Update Records in a Relational Table 72
Delete Records From a Relational Table 75
Import to a Relational Table 77
Export from a Relational Table 81
Purge Data from a Relational Table 83
Delete a Relational Table 84
Create a Contact List 86
Add a Contact to a Contact List 87
Add a Contact to a Program 89
Create a Query of a Database 90
Calculate the Current Contacts for a Query 97
Set a Column Value 98
Automation - User 99

Get a List of Programs.....99
Get a Contact's Relationships to Programs.....102
Scoring Models - User.....108
Get a List of Scoring Models.....108
Get a Contact's Scoring Model Scores.....111
Reporting Interfaces – User.....114
Export Raw Contact Events.....116
Export Web Tracking Events.....125
Get a List of Report IDs for a Mailing.....131
Get a List of Sent Mailings for an Org.....133
Get a List of Sent Mailings for a User.....136
Get a List of Sent Mailings for a Database.....140
Get Aggregate Tracking Metrics for a Mailing.....144
Get Aggregate Tracking Metrics for an Org.....147
Get Aggregate Tracking Metrics for a User.....151
Get Status of a Data Job.....158
Delete a Data Job.....161
Get Folder Path for a Folder or Object.....162
Mailing Management Interfaces – User.....164
Create or Update a Mailing Template.....164
Schedule a Mailing.....169
Preview a Mailing Before Sending.....172
Get Automated Message Group Details.....174
Add a Dynamic Content Ruleset.....175
Import a Dynamic Content Ruleset.....181
Export a Dynamic Content Ruleset.....213
List Dynamic Content Rulesets for a Mailing.....214
Get Dynamic Content Ruleset Details.....216

Replace a Dynamic Content Ruleset 219
Validate a Dynamic Content Ruleset 222
Delete a Dynamic Content Ruleset 224
Get a List of Mailing Templates 225
Note: 226
Export a Mailing Template 227
Appendices 228
Frequently Asked Questions: Non Email Key (NEK) Databases 228
Code Samples 230
PHP 230
Java 238
API Error Codes 239
Sample Error Response 243

About This Guide & Legal

Copyright, Trademark, and Distribution Notice

Usage Disclaimer

This document describes the XML application programming interface (API) for automating certain functions of Silverpop Engage.

The purpose of this document (and of the XML API in general) is to assist developers in automating the interactions between their current software systems and Silverpop Engage. If used incorrectly, you can alter or delete critical data.

Usage of Silverpop's XML API is generally governed by the same terms and conditions as set forth in your Silverpop Engage License Agreement or Silverpop Services Agreement (an Agreement). Usage of Silverpop's XML API is strictly prohibited unless you have a valid, unexpired Agreement in place.

Specifically, Silverpop hereby grants you a limited, nonexclusive, nontransferable, nonsublicensable, worldwide right to use the XML API in support of your development efforts only. All rights not expressly granted to you are reserved by Silverpop.

You shall not (i) license, sublicense, sell, resell, transfer, assign, distribute or otherwise commercially exploit or make available to any third party the XML API or Engage in any way; (ii) reverse engineer or access the XML API or Engage in order to (a) build a competitive product or service, (b) build a product using similar ideas, features, functions or graphics, or (c) copy any ideas, features, functions or graphics, except with Silverpop's prior written approval in each instance. Licenses cannot be shared or used by more than one individual.

What's New in the XML API

The following documentation update is included in the Engage Winter '14 XML API Developer Guide:

- Added a feature to the GetMailingTemplates API so you can optionally request to only see templates that have been made available to your CRM Integration. Alternatively, you can exclude these templates. The added feature is optional and is fully backward compatible with your existing uses of this API.
- Added more clarification to the API Limits section.
- Added more clarification to the OAuth section on Getting Access Tokens.

Introduction

Silverpop's XML API library provides an efficient mechanism to accomplish many automation and integration tasks. The XML API interface defines standard XML requests and responses for the following functionality:

Session Management

- User Login
- User Logout

Database and Contact List Management

- Add a Contact
- Double Opt-In a Contact
- Update a Contact
- Opt-out a Contact
- Select Contact Details
- Import to a Database
- Export from a Database
- Remove a Contact
- Get a List of Databases
- Get Database Details
- Create a Contact List
- Add a Contact to a Contact List
- Create a Query of a Database
- Calculate the Current Contacts for a Query
- Add a Contact to a program
- Get Contact Mailing Details
- Purge Data
- Set a Column Value

Relational Table Management

- Create a Relational Table
- Associate Relational Data with Contacts in a Database
- Insert and Update Records in a Relational Table
- Delete Records From a Relational Table
- Import to a Relational Table
- Export from a Relational Table

- Purge Data from a Relational Table
- Delete a Relational Table

Template and Mailing Management

- Send a Mailing to a Contact
- Forward Mailing to a Friend
- Preview a Mailing Before Sending
- Schedule a Mailing
- Get a List of Mailing Templates
- Export a Mailing Template
- Get Automated Message Group Details

Dynamic Content

- Add a Dynamic Content Ruleset
- Import a Dynamic Content Ruleset
- Export a Dynamic Content Ruleset
- List Dynamic Content Rulesets for a Mailing
- Get Dynamic Content Ruleset Details
- Replace a Dynamic Content Ruleset
- Validate a Dynamic Content Ruleset
- Delete a Dynamic Content Ruleset

Reporting

- Export Raw Contact Events
- Get Report ID by Date
- Get Sent Mailings for an Org
- Get Sent Mailings for a User
- Get Sent Mailings for a Database
- Get Aggregate Tracking Metrics for a Mailing
- Get Aggregate Tracking Metrics for an Org
- Get Aggregate Tracking Metrics for a User
- Export Web Tracking Events

Miscellaneous

- List Contact Mailings

- Get Status of a Data Job
- Delete a Data Job
- Get Folder Path of a Folder or Object

Requests

The interfaces to Engage are facilitated through XML requests over Secure HTTP. You can make the request through either the POST or GET method.

Always begin the XML body format with the <Envelope> element followed by the <Body> element. For example:

```
<Envelope>
        <Body>
                <...your content...>
        </Body>
</Envelope>
```

To pass parameters, list each parameter as a child element of the operation element, as shown below:

```
<Envelope>
   <Body>
      <SendMailing>
         <MailingId>211191</MailingId>
         <RecipientEmail>somebody@domain1.com</RecipientEmail>
      </SendMailing>
   </Body>
</Envelope>
```

Submitting an API Request

Licensed API users should refer to the **Appendix Silverpop API Endpoints** document, located in the XML API Test Harness file in the Support Portal's Knowledge Base, to obtain their appropriate API endpoint.

It is recommended that XML is passed as the **Body** of a **POST**. In order to specify encoding (e.g. UTF-8) of the XML you are submitting, the XML must be part of the POST body. Submitting the XML as a URL parameter will not allow you to specify encoding. The encoding is specified as part of the Content Type in the HTTPS header (for example, Content-Type: text/xml;charset=UTF-8). If you specify an encoding attribute within the XML (for example, <?xml version="1.0" encoding="UTF-8"?>), Engage will ignore it. If you submit XML through the POST Body, but do not include the charset in Content Type, the Organization default is used. If submitting a POST where the body contains content, ensure that Content-Length is specified in the HTTPS header. If passing the XML as a URL parameter (xml=) you must URL encode the XML and specify that it is URL encoded (Content-Type: application/x-www-form-urlencoded).

API requests will pass either the OAuth based Access Token in the request's header or the legacy Java Session ID (jsessionid) as a path parameter in the URL string.

Note: Append the path parameter to the URL using a semicolon, as opposed to the question mark separator used for appending query parameters to a URL. Do **NOT** pass the legacy jsessionID as a form parameter rather than a path parameter.

API Limits

Silverpop has a robust API that handles the automation of most actions available through the Engage UI, such as adding new contacts or sending email campaigns. However, we place limits on the way our customers use our APIs to ensure that we provide the highest level of service and performance to all of our customers. Below are the limits that each Engage Organization is set to by default, however, these limits are adjustable. If you need an increased capacity, contact your Relationship/Renewal Manager:

Silverpop allows our API users to maintain up to 10 active login sessions at any given time. In order to optimize your use of active sessions, make sure that your application uses our Logout API to release an active session when you are finished interacting with our API. Doing so will prevent lingering sessions that are no longer in use from blocking the creation of new active sessions (inactive sessions that are not terminated properly with the Logout API can remain in play for several minutes before expiring).

For OAuth and requesting Access Tokens (see more in the section on "Getting Access Tokens"), our default limits allow you to request up to 10 Access Tokens per hour per client. This limit can be increased dramatically if your integration requires it. Even though our Access Tokens are valid for a fair amount of time, there are scenarios where you will need more tokens per hour to accommodate your use case. For example, if you were to use our Mobile Connector to integrate your mobile apps with our marketing platform, you will need to significantly increase the number of Access Tokens that can be granted per hour. If you feel this is necessary, contact your Relationship/Renewal Manager to request an increase.

Silverpop allows up to 5 concurrent requests to our API servers at any given time. In most cases, using a single connection to our API should be adequate to handle all of your processing needs. If you are approaching the maximum number of connections, consider using one of our asynchronous/batch APIs for moving large data sets into and out of Engage. These APIs are much more efficient at moving large data sets rapidly in and out of our system. We have several robust asynchronous APIs such as ImportList, ExportList and RawRecipientDataExport.

Responses

Successful responses are in the form of XML documents conforming to the general format specified earlier. Each response resides within **<Envelope>** and **<Body>** tags and typically includes a **RESULT** element. Some responses return nothing more than a status, while others return more complex results containing XML elements and child elements.

Example #1: (SendMailing)

```
<Envelope>
   <Body>
      <RESULT>
         <SUCCESS>TRUE</SUCCESS>
      </RESULT>
   </Body>
</Envelope>
```

Example #2: (SelectRecipientData)

```
<Envelope>
  <Body>
    <RESULT>
      <SUCCESS>TRUE</SUCCESS>
      <EMAIL>somebody@domain.com</EMAIL>
      <Email>somebody@domain.com</Email>
      <RecipientId>33439394</RecipientId>
      <EmailType>0</EmailType>
      <LastModified>6/25/04 3:29 PM</LastModified>
      <CreatedFrom>1</CreatedFrom>
      <OptedIn>6/25/04 3:29 PM</OptedIn>
      <OptedOut/>
      <COLUMNS>
        <COLUMN>
          <NAME>Fname</NAME>
          <VALUE>Somebody</VALUE>
         </COLUMN>
         <COLUMN>
          <NAME>Lname</NAME>
          <VALUE>Special</VALUE>
         </COLUMN>
      </COLUMNS>
    </RESULT>
  </Body>
</Envelope>
```

Processing Responses

When processing response envelopes, programmers should use an XML parser and algorithms that handle changes in the number of elements returned in the envelope. This will improve the reliability of scripts that process responses. The contents of a response envelope change in length depending on when you call them (and between releases of Engage). For example, the **GetJobStatus API** will return responses of varying length depending on the status of the job (including WAITING, RUNNING, COMPLETE, and ERROR).

Errors

Unsuccessful responses or errors will also adhere to the general format specified earlier. The XML elements in their bodies will vary based on the type of error. The best way to become familiar with the responses caused by specific error conditions is to recreate them using the XML API Test Harness.

If Engage is not available due to a maintenance outage, the following XML will be returned:

```
<Envelope>
	<Body>
		<RESULT>
			<SUCCESS>false</SUCCESS>
		</RESULT>
		<Fault>
			<Request/>
			<FaultCode/>
			<FaultString><![CDATA[The system is currently down for
maintenance.]]></FaultString>
			<detail>
				<error>
					<errorid>52</errorid>
					<module/>
					<class>SP.Admin</class>
					<method/>
				</error>
			</detail>
		</Fault>
	</Body>
</Envelope>
```

Authentication with the APIs

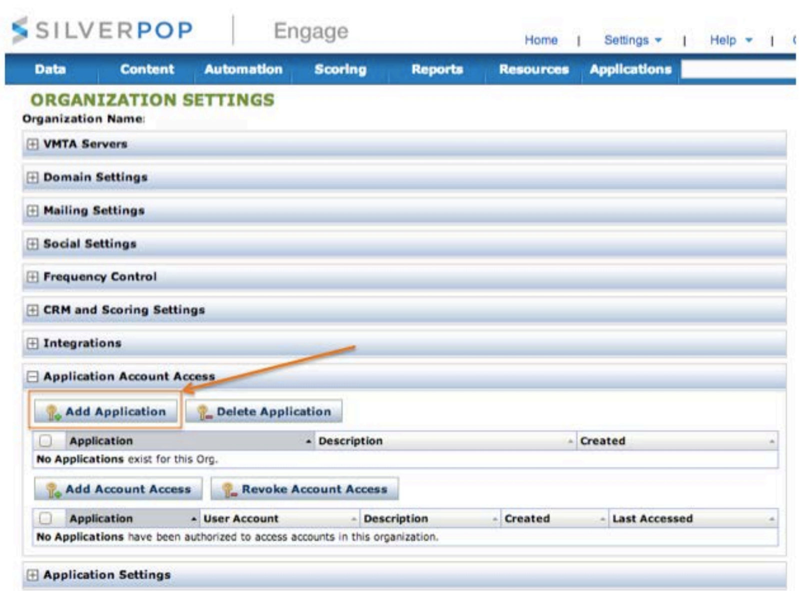
The following section defines the ways that developers can authenticate their API requests before retrieving or modifying their data. We support two types of authentication, but are encouraging our customers to use our newly released OAuth 2.0 system whenever appropriate because of its inherently increased security and implementation flexibility.

Authentication Method: OAuth 2.0 Granted Access

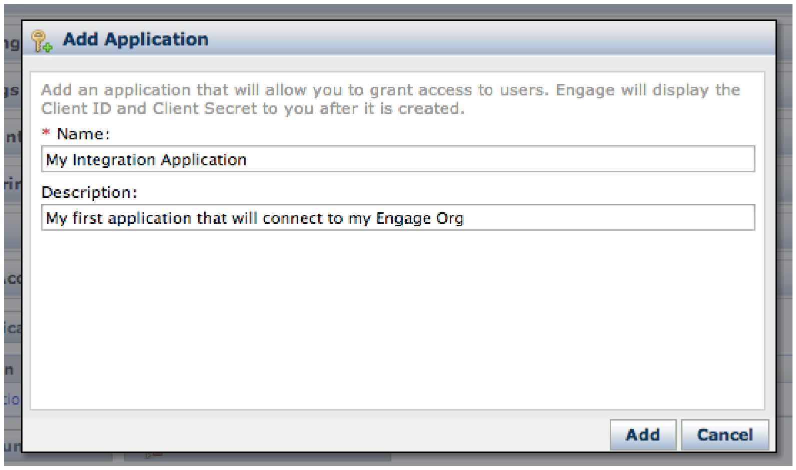
OAuth 2.0 authentication allows you to POST an HTTPS request to our server that embeds an Access Token in the response. The request will include your unique Client Id, Client Secret, and Refresh Token. All of these items are automatically provisioned to you through the Org Admin section of the Engage UI Interface. The response, as stated above, provides you with an Access Token which has a set lifetime of 4 hours. From this point forward, all requests against our API can be authenticated by adding the Access Token to the header of the HTTPS request. When the Access Token expires (or is about to expire), another one can be requested which will allow you to have longer term access when needed. You will not have to get the new Access Token nearly as often as you do with the JSESSIONID.

Getting Started with OAuth in Engage

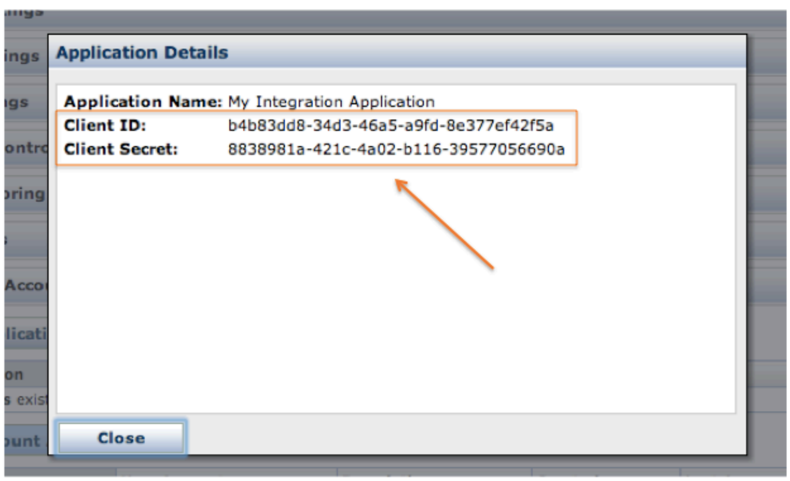
In this section, you will see how to create applications that use OAuth as their authentication mechanism and represent your integrations with Engage. To get started, navigate to **Organization Settings**, click to expand **Application Account Access**, and select **Add Application**.



The next page will prompt you for the Name of the Application you are creating access for and a brief description. Enter a name and description that will easily identify which application is connecting to your data in Engage.



After you click **Add**, you will be able to see your Client Id and Client Secret. You will want to securely embed these tokens within you application.

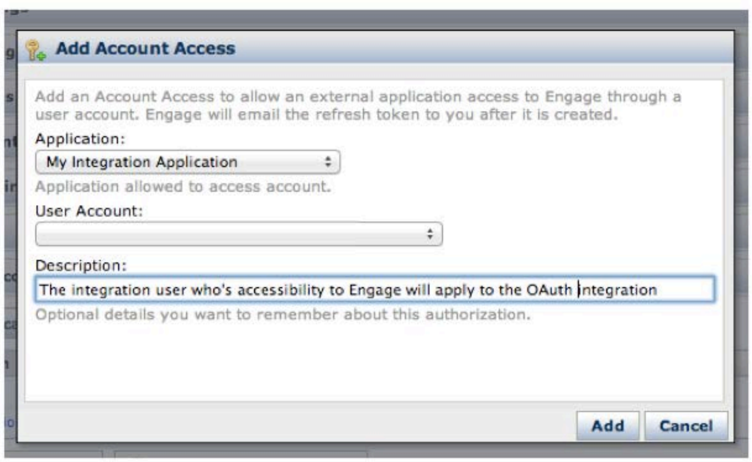


Getting Refresh Tokens

Once you set up your application and get your Client Id and Client Secret tokens, you will be ready to associate a user to that application. We connect users to the application so that we can apply our existing built-in security model as is defined for that user for the application that will be authenticating with OAuth. The user name or password does not need to be (nor should it be) stored in the external application; is only used at this stage in the process to deliver the Refresh token to the **Notification Address** of the user. Make sure you create a new Org Admin user that can be used for your integration and set their notification email address to your own so that you can receive the Refresh Token email when you grant that integration user access to your new application.

Click **Add Account Access**.

Enter the information required and click **Add**:



The notification email address specified for the user that you granted access to your new application integration will subsequently receive an email with the refresh token within it.

Getting Access Tokens

Once you obtain your Client Id, Client Secret, and Refresh Tokens, you will be able to generate your own Access Tokens. Here is how you do it:

You will be sending a HTTP POST to our Engage Server, substituting the following parameters as is appropriate for your Org within Engage:

Description	Generate Access Token
URL Endpoint	https://api[x].silverpop.com/oauth/token
Header	Content-Type: application/x-www-form-urlencoded
Body	grant_type=refresh_token client_id={silverpop generated client Id} client_secret={silverpop generated client secret} refresh_token={silverpop generated refresh token}
Method	POST

Note: Replace [x] with your Engage Pod number.

Once the Access Token is acquired, you can use this cryptic string value to gain access to any of the XML APIs described in this document. To call any of the APIs within this document using the OAuth security model, you will send a HTTP POST to our Engage Server, substituting the following parameters as is appropriate for your Org within Engage:

Description	Invoke XML API
URL Endpoint	https://api[x].silverpop.com/XMLAPI
Header	Authorization: Bearer {access token}
Body	xml=<your xml request here>
Method	POST

Note: Replace [x] with your Engage Pod number.

Legacy Authentication Method: JSESSIONID User Sessions

Our Username/Password paradigm for authenticating against both our UI and our API will continue to be supported, but is considered to be a legacy feature. This approach allows your external systems to securely store a customer-defined integration user's credentials within them to subsequently authenticate and gain access to Engage. The methods used for authenticating against Engage with JSESSIONIDs are as follows:

User Login

Login

Before calling any operation, you must obtain a Jsession ID using the Login operation as shown below.

Operation	**<Login>**		
Elements	USERNAME	Engage user's username.	
	PASSWORD	Engage user's password.	
Example	<Envelope> <Body> <Login> <USERNAME>username@company.com</USERNAME> <PASSWORD>userpassword</PASSWORD> </Login> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS	TRUE	Always set to **TRUE** for successful login.
	SESSIONID	System-generated session ID.	
	SESSION_ENCODING	Encoding of the system-generated session ID. The format for SESSION_ENCODING is as follows: *;jsessionid=SESSIONID* where ***SESSIONID*** is the system-generated session ID returned.	
	ORGANIZATION_ID	Identifies your Engage Organization.	
Example	<Envelope> <Body> <RESULT> <SUCCESS>true</SUCCESS> <SESSIONID>dc302149861088513512481</SESSIONID> <SESSION_ENCODING>;jsessionid=dc302149861088513512481		

	</SESSION_ENCODING> </RESULT> </Body> </Envelope>

User Logout

Logout

After completing the API actions, you must use a **Logout** request to close and invalidate the session, as shown below. This API call is important both from the perspective of better securing your customer data from hackers as well as allows our servers to perform optimally for your subsequent calls.

Operation	**<Logout>**		
Elements		None	
Example	<Envelope> <Body> <Logout/> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS	TRUE	To successfully log off the system, always set to **TRUE**.
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> </RESULT> </Body> </Envelope>		

Contact XML Interfaces

The following section defines the interfaces provided to manage the most common Mailing and Database management actions. We refer to these as **Contact Actions** because each relates directly to one or more Contacts.

Mailing Management Interfaces

Mailing Management interfaces allow you to invoke Engage functions directly related to particular mailings. There are two operations in this category: **SendMailing** and **ForwardToFriend**.

Send a Mailing to a Contact

SendMailing

You can automate a mailing to send after contacts successfully **Opt In** or **Edit Profile** using a Web form associated. You can also set automation for custom events, which are triggered by an API request.

The **SendMailing** operation sends a mailing to the specified contact. The SendMailing operation is specifically for the delivery of autoresponders based on a custom event. This operation will NOT send a mailing **Template**[1]**.** The Mailing ID must be for an existing, custom event autoresponder and the contact email address must be for a contact that already exists in the database associated with the mailing. See the **Add a Contact** for information on adding a contact to a database.

Each **Send** is queued for message generation and delivery. They do not always send immediately, but receive slices of processing time as determined by an algorithm. Small sends such as these are typically built and sent within a range of seconds-to-minutes, but that varies based on several factors such as concurrent system usage.

Note: There is a limit of 1,000 SendMailing API calls per day for an Organization. If you plan to send more than 1,000 transactional emails per day, please contact your Silverpop account manager to discuss alternatives.

Operation	**<SendMailing>**	
Elements	MailingId	Identifies the mailing Engage will send.
	RecipientEmail	Identifies the targeted contact's email address.
	COLUMNS	Optional XML nodes defining the column name and value for the database's key fields. **Note:** Each node must have a Name tag and a Value tag. If the database has a key that is not Email, Unique key columns must also be part of the submission with column names and values.

[1] *Templates are the building blocks for mailings. You can send, schedule, or automate a mailing but not a Template.*

Operation	**<SendMailing>**			
	Child Elements	COLUMN	XML nodes defining the column name and value. **Note:** Each node must have a **Name** tag and a **Value** tag. If the database has a key that is not Email, Unique key columns must also be part of the submission with column names and values.	
		Child Elements	Name	Defines the field name.
			Value	Contains the field value.
Example	<Envelope> <Body> <SendMailing> <MailingId>149482</MailingId> <RecipientEmail>someone@somedomain.com</RecipientEmail> </SendMailing> </Body> </Envelope>			
Response	**<RESPONSE>**			
Elements	SUCCESS	**TRUE** if successful. Upon success, the API immediately schedules a mailing to the contact.		
	ORGANIZATION_ID	Identifies your Engage Organization.		
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> </RESULT> </Body> </Envelope>			

Forward Mailing to a Friend

ForwardToFriend

This operation sends a copy of a mailing along with an additional message to the specified contacts. You must use the Mailing ID of an existing Sent Mailing.

Note: The ForwardToFriend API is **not** available for mailings associated with Non-Email-Key databases.

Operation	**<ForwardToFriend>**	
Elements	SENDER_EMAIL	Email address of message sender (forwarder).
	Rs or r	The internal ID of the contact forwarding the message. Use <rs> with an encoded ID or <r> with an un-encoded ID.
	M	Identifies the mailing to send.
	RECIPIENTS	Identifies the email address of the targeted contact. Separate multiple values with commas.
	MESSAGE	Text of additional message that will accompany the forwarded mailing.
Example	<Envelope> <Body> <ForwardToFriend> <SENDER_EMAIL>someone@somedomain.com</SENDER_EMAIL> <rs>MQS2</rs> <m>5683</m> <RECIPIENTS>someoneelse@somedomain.com, metoo@f2f.com</RECIPIENTS> <MESSAGE>Forwarded Mailing</MESSAGE> </ForwardToFriend> </Body> </Envelope>	
Response	**<FTF_RESULT>**	
Elements	FTF_RESULT	For a successful submission, FTF_RESULT will have no value. Possible error values include: 5 – Error w/ Sender recipient ID 4 – Error w/ Sender email address 3 – Error w/ mailing ID – not found 2 – Error w/ recipient ID 1 – Error w/ mailing ID – invalid format
	ORGANIZATION_ID	Identifies your Engage Organization.
Example	<Envelope> <Body> <FTF_RESULT/> </Body> </Envelope>	

Get Contact Mailing Details

GetContactMailingDetails

This interface returns details about a mailing sent to a specific contact. It is intended to be used with the SureFrom Code which can be placed in the mailing body.

Operation	**<GetContactMailingDetails>**	
Elements	SURE_FROM_CODE	An ID that can be returned in a mailing body and used to validate that a mailing is legitimate.
	ORGANIZATION_ID	**One or more** Organization Ids that will be checked for the SureFrom Code.
Example with normal usage	<Envelope> <Body> <GetContactMailingDetails> <SURE_FROM_CODE>1-8QhP4YIS-XYPxqdpK-ia6t4PP3-+/dKSxVg-uCXkKZka-X77SLMfa-oJ+R5TBj-AX3NoCTU</SURE_FROM_CODE> <ORGANIZATION_ID>1dcd49d-108b594203d-f528764d624db129b32c21fbcaocb8d6</ORGANIZATION_ID> </GetContactMailingDetails> </Body> </Envelope>	
Example 2 with multiple orgs	<Envelope> <Body> <GetContactMailingDetails> <SURE_FROM_CODE>1-8QhP4YIS-XYPxqdpK-ia6t4PP3-+/dKSxVg-uCXkKZka-X77SLMfa-oJ+R5TBj-AX3NoCTU</SURE_FROM_CODE> <ORGANIZATION_ID>1dcd49d-108b594203d-f528764d624db129b32c21fbcaocb8d6</ORGANIZATION_ID> <ORGANIZATION_ID>2edd49d-108b594203d-f528764d624db129b32c21fbcaocc9e7</ORGANIZATION_ID> </GetContactMailingDetails> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**true** if successful.
	SENT_TIMESTAMP	The date/time that the mailing was sent in GMT.
	FROM_NAME	The From Name of the mailing's sender.
	FROM_ADDRESS	The From Address (email) of the mailing's sender.
	SUBJECT	The Subject of the mailing.

	TO_ADDRESS	Optional	The email address of the recipient whose SEND_FROM_CODE was provided. Only returned if ORGANIZATION_ID is provided in the request.
	CLICK_TO_VIEW_URL	Optional	The Click To View URL of the personalized mailing body for the recipient whose SEND_FROM_CODE was provided. Only returned if ORGANIZATION_ID is provided in the request.
	ORGANIZATION_ID	Optional	The Organization Id of the Organization that sent the mailing for the provided SEND_FROM_CODE. Only returned if ORGANIZATION_ID is provided in the request.
Example 1 with org in request	<Envelope> <Body> <RESULT> <SUCCESS>true</SUCCESS> <SENT_TIMESTAMP>09/30/2010 21:08:59 GMT</SENT_TIMESTAMP> <FROM_NAME>Silverpop Support</FROM_NAME> <FROM_ADDRESS>my@domain</FROM_ADDRESS> <SUBJECT>Upcoming Engage Release</SUBJECT> <TO_ADDRESS>recipient@domain.com</TO_ADDRESS> <CLICK_TO_VIEW_URL><![CDATA[https://links.mkt012.com/servlet/MailView?ms=Mjk5OTM5NQS2&r=MjAxOTQ1NjU3OtkS1&j=OdkxNzM3NjUS1&mt=1&rt=0]]></CLICK_TO_VIEW_URL> <ORGANIZATION_ID>1dcd49d-108b594203d-f528764d624db129b32c21fbcaocb8d6</ORGANIZATION_ID> </RESULT> </Body> </Envelope>		
Example 2 without org in request	<Envelope> <Body> <RESULT> <SUCCESS>true</SUCCESS> <SENT_TIMESTAMP>09/30/2010 21:08:59 GMT</SENT_TIMESTAMP> <FROM_NAME>Silverpop Support</FROM_NAME> <FROM_ADDRESS>my@domain.com</FROM_ADDRESS> <SUBJECT>Upcoming Engage Release</SUBJECT> </RESULT> </Body> </Envelope>		
Error example	<Envelope> <Body>		

	<RESULT> <SUCCESS>false</SUCCESS> </RESULT> <Fault> <Request/> <FaultCode/> <FaultString>The value you entered could not be confirmed.</FaultString> <detail> <error> <errorid>330</errorid> <module/> <class>SP.API</class> <method/> </error> </detail> </Fault> </Body> </Envelope>

Purge Data

PurgeData

This interface deletes all records from a database, suppression, seed, test or contact list (target) based on the contacts existing in a specific database, contact list, or query (source).

If both databases share the same key definition, contacts will be matched using the key. However, if the key definition is different, contacts will be matched using email address.

It is suggested that Contacts be purged from a non-keyed database by using an associated query or contact list as the Source. Specifying different non-keyed databases for Target and Source will not result in any matches to be purged.

If a contact list is specified for the target, records are not deleted from the database; the contacts are only removed from the contact list.

A data job will be created upon successful processing of the request. The data job will perform the following:

- Remove contacts found in the target object which also exist in the source object.
- Create a new database or list containing copies of the purged records.

This operation requires exposing the existing Purge functionality to the API with the following enhancements:

- Ability to specify a source query that is owned by the target parent (UI restricts this)
- Ability to purge data from a target Contact List
- Ability to purge a target Database or Contact List using the same object as the source

The existing GetJobStatus operation may be used to determine the status of the data job.

Operation	**<PurgeData>**	
Elements	TARGET_ID	The Id of the Database, Suppression, Seed, Test, or Contact List being purged.
	SOURCE_ID	The Id of the Database, Suppression, Seed, Test, or Contact List, or Query that will be used to determine which records are deleted. ***Note***: If a Query Id is specified, it must be a Classic Query. New Queries are not supported at this time.
Example	<Envelope> <Body> <PurgeData> <TARGET_ID>87981</TARGET_ID> <SOURCE_ID>64987</SOURCE_ID> </PurgeData> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**TRUE** if successful.
	JOB_ID	Identifies the Engage Background Job created for the purge.
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <JOB_ID>98784</JOB_ID> </RESULT> </Body> </Envelope>	

Database Management Interfaces – Contact

Database Management Interfaces allow you to invoke Engage functions directly related to database contacts.

When adding or updating a contact, you can pass values for fields that contain content such as HTML, CSS, XML, and XSL. To accomplish this, you must enclose the content in a **[CDATA]** which will pass it through the XML parser without processing it. Engage will process the content differently depending on the parameters set for the call.

To display the markup as literal characters, enclose it in a **[CDATA]** tag and do not use the **ALLOW_HTML** element in the XML envelope. Engage will convert the special markup characters into their corresponding character entity references. When rendered in a user agent, they will display as the literal characters instead of interpreting the code as markup.

Option 2 – Interpreting the content to render that data

To interpret the content and render the data, enclose the content in a **[CDATA]** tag and set the **ALLOW_HTML** element in the XML envelope to **True**. Engage will pass the special markup characters through as markup and will render the content appropriately. When rendered in a user agent, it will interpret the markup characters and will display the content with the correct formatting.

Examples:

A typical name/value pair used for text values:

```
<NAME>First_Name</NAME>
<VALUE>John</VALUE>
```

A name/value pair when using HTML markup with the text value:

```
<NAME>First_Name</NAME>
<VALUE><![CDATA[<b>John</b>]]></VALUE>
```

When you include the <ALLOW_HTML>true</ALLOW_HTML> element in the envelope with the name/value pair above, the value "John" is displayed in a user agent (with bold formatting). If you leave the ALLOW_HTML element out of the envelope, the value "John" will be displayed just as it appears above in the name/value pair ("John" with the HTML code included).

When using the Engage **Send Time Optimization** feature, you can set the contact's **Send Time** by including a value in the Send Hour field of the database (using the **COLUMN** section of the request to add a contact). Ensure the **Name** element is the name of the field having a Send Hour type and that the **Value** is a number between 0 and 168. This numeric range represents the full number of hours in a week (where the number 1 represents Sunday at 12:00 AM GMT and 168 represents the following Saturday at 11:00 PM GMT). To send the message to the contact immediately, specify a **0**. Note: you will receive this value in the response when you select contact details.

Add a Contact

AddRecipient

This interface adds one new contact to an existing database. If your database has a key other than Email, you must include all unique key columns with their corresponding name/value pairs. If adding and/or updating contacts in a database that has no Unique Identifier defined, one or more Sync Fields must be specified in order to look up the contact.

When adding/updating records in your organization's CRM List, you can specify whether to sync the contact to your CRM system by passing a COLUMN element with a NAME "CRM Enable Sync" and VALUE of **Yes** or **No**. If you do not include this column, the organization's default setting will be used.

When adding/updating records in your organization's CRM List, you can specify whether the individual is a Lead or Contact by passing a COLUMN element with a NAME "CRM Contact Type" and VALUE of "Lead" or "Other". If you do not include this column, the individual will be designated a Lead.

Operation	**<AddRecipient>**		
Elements	LIST_ID	The ID of the database to which you are adding the contact.	
	CREATED_FROM	Value indicating the way in which you are adding the contact to the system. Values include: 0 – Imported from a database 1 – Added manually 2 – Opted in 3 – Created from tracking database	
	SEND_AUTOREPLY	Optional	If the database has an autoresponder associated with it and the SEND_AUTOREPLY element is set to **true**, Engage sends the confirmation when the contact is added to the database.
	UPDATE_IF_FOUND	Optional	If the UPDATE_IF_FOUND element is set to **true**, attempting to add a contact with a duplicate key will result in an update to their record. If the database has no Unique Identifier and UPDATE_IF_FOUND is set to **true**, attempting to add a contact with duplicate SYNC_FIELDS will result in an update to the record.
	ALLOW_HTML	Optional	If ALLOW_HTML is set to **true**, Engage will interpret—and properly render—HTML within a value in the text field. The default value is **false** and will convert HTML tags to character entity references (and tags display as literal

			characters in the text field).	
	VISITOR_KEY	Optional	Allows specifying a visitor key to associate a contact to a previously unknown visitor when using the Web Tracking feature.	
	CONTACT_LISTS	Optional	If included, the contact will be added to all specified contact lists in addition to a database.	
		Child Element	CONTACT_LIST_ID	The contact list id that contact will be added to.
	SYNC_FIELDS	Optional	Required if the database has no Unique Identifier and UPDATE_IF_FOUND is set to **true**. Used to look up a contact in a database that has no Unique Identifier defined. Include a SYNC_FIELD element for each database column that defines a unique contact. ***Note:*** If more than one contact is found matching the lookup columns, the oldest contact will be updated. Sync fields will be ignored if the database has a key.	

	Child Element	SYNC_FIELD	XML nodes defining the column name and value of the unique identifier fields. Each node must have a **Name** tag and a **Value** tag.	
		Child Element	NAME	Defines the field name.
			VALUE	Contains the field value.
	COLUMN	XML nodes defining the column name and value for fields being added or updated. **Note:** Each node must have a Name tag and a Value tag. If the database has a key that is not Email, Unique key columns must also be part of the submission with column names and values.		
	Child Element	Name	Defines the field name. The name of the Email column must be EMAIL. The name of the email type column must be EMAIL_TYPE	
		Value	Contains the field value	
Example with contact list ids	<Envelope> <Body> <AddRecipient> <LIST_ID>85628</LIST_ID> <CREATED_FROM>1</CREATED_FROM> <CONTACT_LISTS> <CONTACT_LIST_ID>289032</CONTACT_LIST_ID> <CONTACT_LIST_ID>12345</CONTACT_LIST_ID>			

	</CONTACT_LISTS> <COLUMN> <NAME>Customer Id</NAME> <VALUE>123-45-6789</VALUE> </COLUMN> <COLUMN> <NAME>EMAIL</NAME> <VALUE>somebody@domain.com</VALUE> </COLUMN> <COLUMN> <NAME>Fname</NAME> <VALUE>John</VALUE> </COLUMN> </AddRecipient> </Body> </Envelope>

Example 2 with Sync Fields	<Envelope> <Body> <AddRecipient> <LIST_ID>85628</LIST_ID> <CREATED_FROM>2</CREATED_FROM> <SYNC_FIELDS> <SYNC_FIELD> <NAME>EMAIL</NAME> <VALUE>somebody@domain.com</VALUE> </SYNC_FIELD> <SYNC_FIELD> <NAME>Customer Id</NAME> <VALUE>123-45-6789</VALUE> </SYNC_FIELD> </SYNC_FIELDS> <COLUMN> <NAME>Street_Address</NAME> <VALUE>123 New Street</VALUE> </COLUMN> </AddRecipient> </Body> </Envelope>

Response	**<RESULT>**		
Elements	SUCCESS	**True** if successful.	
	RecipientId	ID of added or updated contact.	
	ORGANIZATION_ID	Identifies your Engage Organization.	
	VISITOR_ASSOCIATION	Optional	Indicates whether the specified Visitor ID is valid and was associated to a profile.
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <RecipientId>33535067</RecipientId> </RESULT> </Body> </Envelope>		

Double Opt-in a Contact

DoubleOptInRecipient

This interface confirms the Opt-in for a contact in a **Double Opt-in Database**. Any fields specified will override any previously specified values set when creating the contact from the **Pre Opt-in** state. **Note:** You must specify all key fields in the COLUMN elements.

Operation	**<DoubleOptInRecipient>**		
Elements	LIST_ID	The Is of the **Double Opt-In** database to which Engage will add the contact.	
	SEND_AUTOREPLY	Optional	If **true** and database has a confirmation autoresponder, Engage will send the confirmation.
	ALLOW_HTML	Optional	If **true,** Engage will interpret and properly render HTML within a value in the text field. The default value is **false** and will convert HTML tags to character entity references (tags display as literal characters in the field).
	COLUMN	XML nodes defining the column name and value. **Note:** A node must exist for the Email column. Each node must have a **Name** tag and a **Value** tag. Unique key columns must be part of the submission with column names and values.	
	Child Elements	NAME	Defines the field name. The name of the Email column must be EMAIL. The name of the email type column must be EMAIL_TYPE.
		VALUE	Contains the field value.
Example	<Envelope> <Body> <DoubleOptInRecipient> <LIST_ID>85628</LIST_ID> <COLUMN> <NAME>Customer Id</NAME> <VALUE>123-45-6789</VALUE> </COLUMN> <COLUMN> <NAME>EMAIL</NAME> <VALUE>somebody@domain.com</VALUE> </COLUMN> </DoubleOptInRecipient> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS	**True** if successful.	
	RecipientId	ID of newly added contact.	
	ORGANIZATION_ID	Identifies your Engage Organization.	
Example	<Envelope> <Body> <RESULT>		

	<SUCCESS>TRUE</SUCCESS> <RecipientId>33535067</RecipientId> </RESULT> </Body> </Envelope>

Update a Contact

UpdateRecipient

This interface updates a contact in an existing database. Use of the optional **OPT_OUT** child element allows this operation to add a currently opted-out contact back into the database.

If your database has a key other than Email, you must include all unique key columns with their corresponding name/value pairs.

If updating contacts in a database that has no Unique Identifier defined, one or more Sync Fields must be specified in order to look up the contact.

When adding/updating records in your organization's CRM List, you can specify whether to sync the contact to your CRM system by passing a COLUMN element with a NAME "CRM Enable Sync" and VALUE of "Yes" or "No". If you do not include this column, the organization's default setting will be used.

When adding/updating records in your organization's CRM List, you can specify whether the individual is a Lead or Contact by passing a COLUMN element with a NAME "CRM Contact Type" and VALUE of "Lead" or "Other". If you do not include this column, the individual will be designated a Lead.

Key columns may only be updated if the RECIPIENT_ID or ENCODED_RECIPIENT_ID is passed to look up the contact.

Operation	**<UpdateRecipient>**		
Elements	LIST_ID	The ID of the database to which the contact resides in.	
	OLD_EMAIL	Optional	Old email address for the contact.
	RECIPIENT_ID	Optional	The Recipient Id may be provided with a LIST_ID to look up a contact. No key fields are required when passing the Recipient Id.
	ENCODED_RECIPIENT_ID	Optional	The encoded Recipient Id may be provided with a LIST_ID to look up a contact. No key fields are required when passing the Recipient Id. ***Note***: The encoded Recipient Id can be obtained from a sent mailing when %%RECIPIENT_ID%% is placed in the mailing body.
	SEND_AUTOREPLY	Optional	If the database has an autoresponder associated with it and the SEND_AUTOREPLY element is set to **true,** Engage sends the confirmation when the contact is added to the database.

	ALLOW_HTML	Optional	If **true**, Engage will interpret—and properly render—HTML markup within a value in the text field. The default value is **false**, which will convert HTML tags to character entity references (and tags will display as literal characters in the field).
	VISITOR_KEY	Optional	Allows specifying a visitor key to associate a contact to a previously unknown visitor when using the Web Tracking feature.
	SYNC_FIELDS	Optional	Required if the database has no Unique Identifier. Used to look up a contact in a database that has no Unique Identifier defined. Include a SYNC_FIELD element for each column that defines a unique contact. ***Note:*** If more than one contact is found matching the lookup columns, the oldest contact will be updated.

	Child Element	SYNC_FIELD	XML nodes defining the column name and value of the unique identifier fields. Each node must have a **Name** tag and a **Value** tag.	
		Child Element	NAME	Defines the field name.
			VALUE	Contains the field value.
	COLUMN	XML nodes defining the column name and value. **Note:** Each node must have a **Name** tag and a **Value** tag. If the database has a key that is not Email, Unique key columns must also be part of the submission with column names and values. If changing a value of one of the SYNC_FIELDS, the new value should be included as a COLUMN node. In order to subscribe a previously opted out contact, specify "OPT_OUT" in a NAME element and "false" in the corresponding VALUE element.		
	Child Element	NAME	Defines the field name.	
		VALUE	Contains the field value.	
	SNOOZE_SETTINGS	Optional	XML node defining snooze settings.	
	Child Elements	SNOOZED	If **true** snooze is activated for that recipient. For manual resume set to **false.**	
		RESUME_SEND_ DATE	Specify date to resume sending communication from Engage.	
		DAYS_TO_SNO OZE	Number of days to stop sending communication from Engage.	

Example 1 with normal usage	<Envelope> <Body> <UpdateRecipient> <LIST_ID>85628</LIST_ID> <CREATED_FROM>2</CREATED_FROM> <OLD_EMAIL>somebody@domain.com</OLD_EMAIL> <COLUMN> <NAME>Customer Id</NAME> <VALUE>123-45-6789</VALUE> </COLUMN> <COLUMN> <NAME>EMAIL</NAME> <VALUE>somebodynew@domain.com</VALUE> </COLUMN> <COLUMN> <NAME>Street_Address</NAME> <VALUE>123 New Street</VALUE> </COLUMN> <SNOOZE_SETTINGS> <SNOOZED>true</SNOOZED> <RESUME_SEND_DATE>03/15/2015</RESUME_SEND_DATE> </SNOOZE_SETTINGS> </UpdateRecipient> </Body> </Envelope>

Example 2 with resubscribe

```
<Envelope>
  <Body>
    <UpdateRecipient>
      <LIST_ID>85628</LIST_ID>
      <CREATED_FROM>2</CREATED_FROM>
      <OLD_EMAIL>somebody@domain.com</OLD_EMAIL>
      <COLUMN>
        <NAME>OPT_OUT</NAME>
        <VALUE>false</VALUE>
      </COLUMN>
      <COLUMN>
        <NAME>Customer Id</NAME>
        <VALUE>123-45-6789</VALUE>
      </COLUMN>
      <COLUMN>
        <NAME>EMAIL</NAME>
        <VALUE>somebodynew@domain.com</VALUE>
      </COLUMN>
      <COLUMN>
        <NAME>Street_Address</NAME>
        <VALUE>123 New Street</VALUE>
      </COLUMN>
    </UpdateRecipient>
  </Body>
</Envelope>
```

Example 3 with Sync Fields	<Envelope> <Body> <UpdateRecipient> <LIST_ID>85628</LIST_ID> <CREATED_FROM>2</CREATED_FROM> <SYNC_FIELDS> <SYNC_FIELD> <NAME>EMAIL</NAME> <VALUE>somebody@domain.com</VALUE> </SYNC_FIELD> <SYNC_FIELD> <NAME>Customer Id</NAME> <VALUE>123-45-6789</VALUE> </SYNC_FIELD> </SYNC_FIELDS> <COLUMN> <NAME>EMAIL</NAME> <VALUE> somebody2@domain.com</VALUE> </COLUMN> <COLUMN> <NAME>Street_Address</NAME> <VALUE>123 New Street</VALUE> </COLUMN> </UpdateRecipient> </Body> </Envelope

Response	**<RESULT>**		
Elements	SUCCESS	**True** if successful.	
	Recipient ID	ID of the updated contact. If an encoded Recipient Id is specified in the request, the Recipient Id in the response will be encoded.	
	ORGANIZATION_ID	Identifies your Engage Organization.	
	VISITOR_ASSOCIATION	Optional	Indicates whether the specified Visitor ID is valid and was associated to a profile.
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <RecipientId>33439394</RecipientId> </RESULT> </Body> </Envelope>		

Opt-out a Contact

OptOutRecipient

This interface moves a contact in a database to an opted-out state.

If you are using a custom opt-out page and wish to record opt outs against specific mailings, you must ensure that the link to your opt-out page within your mailing templates includes the following parameters:

%%MAILING_ID%%
%%RECIPIENT_ID%%
%%JOB_ID_CODE%%

Reliability Measures

SUCCESS in the XML response indicates that the system can—and will—opt out the contact. However, the opt-out may not happen immediately; Engage places all opt-out events in an event queue and processes them in the order in which the events are received. If an internal error occurs while processing the event from the queue, Engage places the event in an error queue. When the problem is resolved, it moves back into the event queue.

We recommend that you document any responses other than **SUCCESS** to assist customer support in a manual investigation, should that be necessary. The following categories of errors can result in an unsuccessful request:

- Invalid field formats
- Database does not exist in Engage
- Missing parameters when opting out for a particular mailing
- Database prohibits opting out through API
- Limited rights on the related database

To ensure opt-out reliability, you should continue to retry any opt-out requests that do not receive a response until you receive a **SUCCESS** response.

Operation	**<OptOutRecipient>**		
Elements	LIST_ID	Identifies the ID of the database from which to opt out the contact.	
	EMAIL	The contact email address to opt out. **Note:** If using a regular email key database, a node must exist for the Email column. If passing MAILING_ID, RECIPIENT_ID, and JOB_ID, Engage does not require EMAIL. You must provide each of the three elements if EMAIL is not included.	
	MAILING_ID	The internal Mailing ID passed to the opt-out page using personalized links in the mailing template. Used to track opt-outs at the mailing level.	
	RECIPIENT_ID	The internal ID of the Contact passed to the opt-out page using personalized links in the mailing template. Used to track opt-outs at the mailing level.	
	JOB_ID	The internal Job ID passed to the opt-out page using personalized links in the mailing template. Used to track opt-outs at the mailing level.	
	COLUMN	XML nodes defining the column name and value. Ignored for regular email key databases. **Note:** Unique key columns must be part of the submission with column names and values. If you only specify the email address, the system will opt out all contacts with that email address.	
	Child Elements	NAME	Specifies the field name of the unique key for the database.
		VALUE	Specifies the value of the field name for the unique key in the database

Example 1 with Standard Opt Out

```
<Envelope>
    <Body>
        <OptOutRecipient>
            <LIST_ID>45654</LIST_ID>
            <EMAIL>someone@adomain.com</EMAIL>
            <COLUMN>
              <NAME>Customer Id</NAME>
              <VALUE>123-45-6789</VALUE>
            </COLUMN>
        </OptOutRecipient>
    </Body>
</Envelope>
```

Example 2 with Opt Out Tracked at Mailing Level

```
<Envelope>
        <Body>
                <OptOutRecipient>
                        <LIST_ID>45035</LIST_ID>
                        <MAILING_ID>132767</MAILING_ID>
                        <RECIPIENT_ID>NTM3NjIoNTg3S0</RECIPIENT_ID>
                        <JOB_ID>NTQ5NTk5MwS2</JOB_ID>
                </OptOutRecipient>
        </Body>
</Envelope>
```

Response	**<RESULT>**	
Elements	SUCCESS	**SUCCESS** if successful.
	ORGANIZATION_ID	Identifies your Engage Organization.
Example	<Envelope> <Body> <RESULT> <SUCCESS>SUCCESS</SUCCESS> </RESULT> </Body> </Envelope>	

Select Contact Details

SelectRecipientData

This interface retrieves the information about a contact in a database.

If your database has a key other than Email, you must include all unique key columns with their corresponding name/value pairs.

If your database has no Unique Identifier defined, one or more columns must be specified in order to look up the contact.

If both Recipient Id and Visitor Key are provided, Recipient Id is used to lookup a contact.

Operation	**<SelectRecipientData>**		
Elements	LIST_ID	ID of the database to which the contact belongs. If a Contact List is specified instead of a database, the system will determine whether the contact is in the Contact List. If the contact is found, their database columns will be returned.	
	EMAIL	The contact email address to look up. **Note:** When using a Non-Email Key or No-Key database, this element is not required	
	RECIPIENT_ID	Optional	The Recipient Id may be provided with a LIST_ID to look up a contact. No key fields are required when passing the Recipient Id.
	ENCODED_RECIPIENT_ID	Optional	The encoded Recipient Id may be provided with a LIST_ID to look up a contact. No key fields are required when passing the encoded Recipient Id.
	VISITOR_KEY	Optional	The Silverpop Web Tracking visitor key may be provided with a LIST_ID to look up a contact. No key fields are required when passing the Silverpop Web Tracking Visitor key. The Silverpop Web Tracking Visitor key is available in the Web Tracking cookie stored in a contact's browser settings.

	RETURN_CONTACT_LISTS	Optional	Indicates whether to return all Contact Lists which the contact belongs to. Valid values are: • true • false
	COLUMN	Optional	Required if the database has a key other than Email or no Unique Identifier. XML nodes defining the column name and value used to look up a contact in a database. Ignored for regular email key databases. **Note:** Unique key columns must be part of the submission with column names and values. ***Note:*** If more than one contact is found matching the lookup columns, the oldest contact will be returned.
	Child Elements	Name	Specifies the field name of the unique key for the database.
		Value	Specifies the field value for the field name of the unique key for the database.

Example

```
<Envelope>
	<Body>
		<SelectRecipientData>
			<LIST_ID>45654</LIST_ID>
			<EMAIL>someone@adomain.com</EMAIL>
				<COLUMN>
					<NAME>Customer Id</NAME>
					<VALUE>123-45-6789</VALUE>
				</COLUMN>
		</SelectRecipientData>
	</Body>
</Envelope>
```

Response	**<RESULT>**	
Elements	SUCCESS	**True** if successful.
	EMAIL	Email address for the contact.
	Email	Email address for the contact. **Note:** At the time of this writing, email is included twice in the response.
	RecipientId	Internal unique ID for each contact. If an encoded Recipient Id is specified in the request, the Recipient Id in the response will be encoded.
	EmailType	Value indicating the preferred email type for the selected contact. 0 – HTML 1 – Text 2 – AOL

	LastModified	Last time the contact record was modified.	
	CreatedFrom	Value indicating the way in which Engage added the selected contact to the system. 0 – Imported from a database 1 – Manually added 2 – Opted in 3 – Created from tracking database	
	OptedIn	Contact's opt-in date, if applicable.	
	OptedOut	Contact's opt-out date, if applicable.	
	ResumeSendDate	The date contact will be opted back in if they are presently in a Snoozed state.	
	ORGANIZATION_ID	Identifies your Engage Organization.	
	CONTACT_LISTS	If the RETURN_CONTACT_LISTS is 'true' in the request, this parent node will be included in the response. It will contain a child element for each Contact List the contact belongs to.	
	Child Elements	CONTACT_LIST_ID	Id of a Contact List which the contact belongs to.
	COLUMNS	XML nodes defining the user-created column name and value.	
Child Elements	COLUMN		
	Child Element	Name	Specifies the field name.
		Value	Specifies the field value.

Example	```<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS>```

```
<Envelope>
	<Body>
		<RESULT>
			<SUCCESS>TRUE</SUCCESS>
			<EMAIL>somebody@domain.com</EMAIL>
			<Email>somebody@domain.com</Email>
			<RecipientId>33439394</RecipientId>
			<EmailType>0</EmailType>
			<LastModified>6/25/04 3:29 PM</LastModified>
			<CreatedFrom>1</CreatedFrom>
			<OptedIn>6/25/04 3:29 PM</OptedIn>
			<OptedOut/>
			<COLUMNS>
				<COLUMN>
					<NAME>Fname</NAME>
					<VALUE>Somebody</VALUE>
				</COLUMN>
				<COLUMN>
					<NAME>Lname</NAME>
					<VALUE>Special</VALUE>
				</COLUMN>
			</COLUMNS>
		</RESULT>
	</Body>
</Envelope>
```

User XML Interfaces

This section defines the automation interfaces for importing and/or updating databases from external files. We refer to these as **User Actions**.

Database Management Interfaces – User

Import to a Database

ImportList

This interface allows importing a batch file containing new, modified, or opted out contacts. Contacts can also be added to an existing Contact List.

Use the steps below to perform an import through the API:

1. Upload the **source file** to the Engage FTP server.
2. Upload the **Definition and Column Mapping** file to the Engage FTP server.
3. Authenticate the user with a **Login API** call.
4. Initiate the database import using an **ImportList** call.
5. Determine status of the background import job by making a Get**JobStatus API** call (optional).
6. Log off from the Engage API using a **Logout API** call.

You must place all files in the **upload** directory on the FTP server or the database import service will not "find" them.

Note: Contact your Relationship Manager to obtain FTP logon information.

Below is the syntax of the request and response XML:

Operation	**<ImportList>**		
Elements	MAP_FILE	The name of the **Mapping** file in the **upload** directory of the FTP server to use for the import.	
	SOURCE_FILE	The name of the file containing the contact information to use in the import. This file must reside in the **upload** directory of the FTP Server.	
	EMAIL	Optional	If specified, this email address receives notification when the job is complete.
	FILE_ENCODING	Optional	Defines the encoding of the source file. Supported values are: • UTF-8 • ISO-8859-1 If not specified, Engage will use the Org default encoding.
Example	<Envelope> <Body> <ImportList> <MAP_FILE>list_import_map.xml</MAP_FILE> <SOURCE_FILE>list_create.csv</SOURCE_FILE> </ImportList> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS	**TRUE** if successful.	
	JOB_ID	Identifies the Engage Background Job created and scheduled for this import.	
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <JOB_ID>108518</JOB_ID> </RESULT> </Body> </Envelope>		

Database Definition and Column Mapping Format

Mapping is the process of associating columns in the import file with the corresponding fields in the database. For example, **Column 1** in the mapping file corresponds to **Name** in the database. You must describe the details of the database including its columns as well as the format of the source file (which contains the actual contact records). You will place this information in an XML mapping file and upload it to a specified directory on an FTP server.

Add the **<KEY_COLUMN>** element (set to **True**) in a field declaration in the mapping file to set a field as a unique key for the database. You can also define more than one unique key field for a database. You must define the **EMAIL** field in the mapping file and cannot leave it out for an NEK database. The column TYPE should always be set to **9** and the **IS_REQUIRED** element should always be set to **True**. However, you must specify **KEY_COLUMN** as **True** in order to list it as a unique key. If not specified, it will be a regular field in the database.

Engage returns errors in the XML submission as part of the XML response. However, Engage does not detect errors in the mapping file or the source file until it runs the data job. You can view **Data Job** errors on the **Data Jobs** screen of the Engage user interface.

The XML file contains three main sections:

> **LIST_INFO**: defines information about the database you are creating (or updating), as well as actions to perform on this database.
>
> **COLUMNS**: if creating a new database, this section consists of one or more **COLUMN** elements. Each COLUMN element contains child elements to define the attributes for that column (for example, name, type, and key column). If the action is not **CREATE** (and you are not adding new columns to the database), you can exclude the COLUMNS section from the mapping file.
>
> **MAPPING**: this section also consists of COLUMN elements. Each COLUMN element contains child elements to define the attributes for mapping that column (for example, name and index).
>
> **CONTACT_LISTS**: this optional section is used to specify one or more Contact Lists that all contacts will be added to in addition to the database. This section may be used with all actions except OPT_OUT.

The Import Data Job will handle imports of contacts to a database having No Unique Identifier as follows:

- If the ACTION is CREATE or ADD_ONLY and SYNC_FIELDS are NOT specified, each row in the source file will result in a new contact.
- If the ACTION is CREATE and SYNC_FIELDS are specified, rows having identical SYNC_FIELDS within the source file will be rejected as duplicates.
- If the ACTION is ADD_ONLY and SYNC_FIELDS are specified, rows having identical SYNC_FIELDS within the source file or in the database will be rejected as duplicates.
- If the ACTION is UPDATE_ONLY, contacts matching the SYNC_FIELDS will be updated; all other contacts will be ignored. If more than one contact is found matching the lookup columns, all contacts will be updated.
- If the ACTION is ADD_AND_UPDATE, contacts matching the SYNC_FIELDS will be updated; all other contacts will be added. If more than one contact is found matching the lookup columns, all contacts will be updated.
- If the ACTION is OPT_OUT, contacts matching the SYNC_FIELDS will be opted out; all other contacts will be ignored. If more than one contact is found matching the lookup columns, the all matching contacts will be opted out.

Note: All column names are case-sensitive, meaning you must ensure the names in the mapping file and source file exactly match the field names in the Engage database.

The table below defines the sections. You should contain each of these sections within the **LIST_IMPORT** element:

Section	<LIST_INFO>	The LIST_INFO section defines the structure of the database you are creating or updating, as well as the action to perform on this database.
Elements	ACTION	Defines the type of import you are performing. The following is a list of valid values and how Engage interprets them: • CREATE – create a new database. • ADD_ONLY – only add new contacts to the database. Ignore existing contacts when found in the source file. • UPDATE_ONLY – only update the existing contacts in the database. Ignore contacts who exist in the source file but not in the database. • ADD_AND_UPDATE – process all contacts in the source file. If they already exist in the database, update their values. If they do not exist, create a new record in the database for the contact. • OPT_OUT – opt out any contact in the source file who is already in the database. Ignore contacts who exist in the source file but not the database. All contacts matching the provided Email Address will be opted out unless the database has no unique identifier and SYNC_FIELDS are provided.

	LIST_TYPE	Defines the type of database. Only specified if the ACTION is CREATE. Supported values are: 0 – Database 6 – Seed list 13 – Suppression list	
	LIST_NAME	Defines the name of the new database if the ACTION is CREATE (names must be unique).	
	LIST_ID	Unique ID of the database in the Engage system. You must provide this for all ACTION values (except CREATE).	
	LIST_VISIBILITY	Required. Defines the visibility of the newly created database. 0 – private 1 – shared	
	PARENT_FOLDER_PATH	Used with the CREATE ACTION to specify the folder to place the new database in. Use the VISIBILITY element to specify Private/Shared.	
	FILE_TYPE	Defines the formatting of the source file. Supported values are: 0 – CSV file 1 – Tab-separated file 2 – Pipe-separated file	
	HASHEADERS	The HASHEADERS element is set to **true** if the first line in the source file contains column definitions.	
	LIST_DATE_FORMAT	Optional	Used to specify the date format and date fields in your file if your date format differs from "mm/dd/yyyy" (the month, day, and year can be in any order you choose). Valid values for Month are: • mm (e.g. 01) • m (e.g. 1) • mon (e.g. Jan) • month (e.g. January) Valid values for Day are: • dd (e.g. 02) • d (e.g. 2) Valid values for Year are: • yyyy (e.g. 1999) • yy (e.g. 99) Separators may be up to two characters in length and can consist of periods, commas, question marks, spaces, and forward slashes (/). **Examples:** • For "Jan 2, 1975" your LIST_DATE_FORMAT would be "mon d, yyyy" • For "1975/09/02" your LIST_DATE_FORMAT would be "yyyy/mm/dd"
	DOUBLE_OPT_IN	Optional	If this element accompanies the CREATE action, Engage creates a **Double Opt-In** database. If you do not specify the element, the Organization default is used. **Note:** The database security settings that allow you to use the database import operation are, by default, disabled for all Double Opt-In databases.
	ENCODED_AS_MD5	Optional	If this element is set to **true,** email addresses in the Email column will be MD5 encoded.
	SYNC_FIELDS	Optional	Required if the database has no Unique Identifier and action is UPDATE_ONLY,

			ADD_AND_UPDATE, or OPT_OUT. Used to specify what columns are used to look up a contact in a database that has no Unique Identifier defined. Include a SYNC_FIELD element for each database column that defines a unique contact. ***Note:*** If more than one contact is found matching the lookup columns, all matching contacts will be updated. If the ACTION is opt out, all matching contacts will be opted out.
	Child Element	SYNC_FIELD	XML nodes defining the column Name of the unique identifier fields.
		Child Element	NAME — Defines the field name.
Section	**<COLUMNS>**	This section defines how many columns to write to the database (0 or more). Use a separate COLUMN element to define each database column. If the ACTION is not CREATE and you are not adding new columns to the database, you can exclude the COLUMNS section from the mapping file.	
Elements	COLUMN		
	Child Elements	NAME	Defines the name of the column.
		TYPE	Defines what type of column to create. The following is a list of valid values : 0 – Text column 1 – YES/No column 2 – Numeric column 3 – Date column 4 – Time column 5 – Country column 6 – Select one 8 – Segmenting 9 – System (used for defining EMAIL field only) 13 – SMS Opt In 14 – SMS Opted Out Date 15 – SMS Phone Number 16 – Phone Number 17 – Timestamp 20 – Multi-Select **Note:** The import process does not validate the data in the source file against these column types.
		IS_REQUIRED	Defines whether to create the column as "required" when contacts are providing information through a Web form. Valid values are **True** and **False**. **Note:** The import process does not use this information to validate the data.
		KEY_COLUMN	Added to field definition and defines a field as a unique key for the database when set to **True**. You can define more than one unique field for each database. **Note**: When creating a non-keyed database and not indicating any key fields, RECIPIENT_ID column must be defined as follows: <COLUMN> <NAME>RECIPIENT_ID</NAME> <TYPE>9</TYPE> <IS_REQUIRED>True</IS_REQUIRED> <KEY_COLUMN>True</KEY_COLUMN> <DEFAULT_VALUE/> </COLUMN>

		DEFAULT_VALUE	Defines the default value of the column used by the **Send** process if the contact does not contain a value for this column.
		SELECTION_VALUE	Defines values in a drop-down list. You must provide at least one element for each value. Used in conjunction with **TYPE 6**.
Section	**<MAPPING>**	Use the MAPPING section to define which columns in the source file map to which columns in the database. Define each column in its own COLUMN element.	
Elements	COLUMN		
	Child Elements	INDEX	The INDEX element defines the order in which the source file defines this column.
		NAME	The name of the database column to which Engage maps the source file column. You can use the following system-defined names in the mapping file: • EMAIL – the email address. • EMAIL_TYPE – the type of contact body to send. • CREATED_FROM – An integer defining where Engage collected the contact's information. • OPTED_IN – the date and time the contact was opted into the database. Engage assumes all times to be GMT. If there is no opt-in date specified, the system will insert the current date and time (in GMT). • OPT_IN_DETAILS – any information (up to 250 characters) that defines where Engage obtained the contact's information. • OPTED_OUT – the date and time the contact opted out of the database (for OPT_OUT actions only). All system-type field names must be in uppercase.
		INCLUDE	Defines whether to include the element in the database import. You do not need to include all source file columns in the import. This element allows you to specify which columns to skip during the Import process. Valid values are **True** and **False**.
Section	**<CONTACT LISTS>**	Use the CONTACT_LISTS section if you want to specify one or more Contact Lists that all contacts will be added to in addition to the database. This section may be used with all actions except OPT_OUT.	
	Child Elements	CONTACT_LIST_ID	Zero to many elements defining the Id of an **existing** Contact List.

Example Mapping Files

Example 1

This example shows a simple mapping file with the unique key set as **CustID**. Email is **not** a unique key in this example.

```
<LIST_IMPORT>
	<LIST_INFO>
		<ACTION>CREATE</ACTION>
		<LIST_NAME>Premier Accts</LIST_NAME>
		<LIST_VISIBILITY>0</LIST_VISIBILITY>
		<FILE_TYPE>0</FILE_TYPE>
		<HASHEADERS>true</HASHEADERS>
	</LIST_INFO>
	<COLUMNS>
		<COLUMN>
			<NAME>EMAIL</NAME>
			<TYPE>9</TYPE>
			<IS_REQUIRED>true</IS_REQUIRED>
		</COLUMN>
		<COLUMN>
			<NAME>CustID</NAME>
			<TYPE>0</TYPE>
			<IS_REQUIRED>true</IS_REQUIRED>
			<KEY_COLUMN>true</KEY_COLUMN>
		</COLUMN>
	</COLUMNS>
	<MAPPING>
		<COLUMN>
			<INDEX>1</INDEX>
			<NAME>EMAIL</NAME>
			<INCLUDE>true</INCLUDE>
		</COLUMN>
		<COLUMN>
			<INDEX>2</INDEX>
			<NAME>CustID</NAME>
			<INCLUDE>true</INCLUDE>
		</COLUMN>
	</MAPPING>
</LIST_IMPORT>
```

Example 2

This example shows a more complicated mapping file with custom column definitions. Both EMAIL and CustID are unique keys. Contacts are also being added to existing Contact Lists.

```
<LIST_IMPORT>
        <LIST_INFO>
               <ACTION>ADD_AND_UPDATE</ACTION>
                <LIST_ID>50194</LIST_ID>
                <FILE_TYPE>0</FILE_TYPE>
                <HASHEADERS>true</HASHEADERS>
        </LIST_INFO>
        <COLUMNS>
                <COLUMN>
                        <NAME>EMAIL</NAME>
                        <TYPE>9</TYPE>
                        <IS_REQUIRED>true</IS_REQUIRED>
                        <KEY_COLUMN>true</KEY_COLUMN>
                </COLUMN>
                <COLUMN>
                        <NAME>CustID</NAME>
                        <TYPE>0</TYPE>
                        <IS_REQUIRED>true</IS_REQUIRED>
                        <KEY_COLUMN>true</KEY_COLUMN>
                </COLUMN>
                <COLUMN>
                        <NAME>Att1</NAME>
                        <TYPE>0</TYPE>
                        <IS_REQUIRED>false</IS_REQUIRED>
                        <DEFAULT_VALUE/>
                </COLUMN>
                <COLUMN>
                        <NAME>Att2</NAME>
                        <TYPE>0</TYPE>
                        <IS_REQUIRED>false</IS_REQUIRED>
                        <DEFAULT_VALUE/>
                </COLUMN>
                <COLUMN>
                        <NAME>CountryField1</NAME>
                        <TYPE>0</TYPE>
                        <IS_REQUIRED>false</IS_REQUIRED>
                        <DEFAULT_VALUE/>
                </COLUMN>
                <COLUMN>
                        <NAME>CountryField2</NAME>
                        <TYPE>0</TYPE>
```

	(XML continued below)


```
                        <IS_REQUIRED>false</IS_REQUIRED>
                        <DEFAULT_VALUE/>
                </COLUMN>
                <COLUMN>
                        <NAME>DateField1</NAME>
                        <TYPE>0</TYPE>

                        <IS_REQUIRED>false</IS_REQUIRED>
                        <DEFAULT_VALUE/>
                </COLUMN>
        </COLUMNS>
        <MAPPING>
                <COLUMN>
                        <INDEX>1</INDEX>
                        <NAME>EMAIL</NAME>
                        <INCLUDE>true</INCLUDE>
                </COLUMN>
                <COLUMN>
                        <INDEX>2</INDEX>
                                <NAME>EMAIL_TYPE</NAME>
                        <INCLUDE>true</INCLUDE>
                </COLUMN>
                <COLUMN>
                        <INDEX>3</INDEX>
                        <NAME>CustID</NAME>
                        <INCLUDE>true</INCLUDE>
                </COLUMN>
                <COLUMN>
                        <INDEX>4</INDEX>
                        <NAME>DateField1</NAME>
                        <INCLUDE>true</INCLUDE>
                </COLUMN>
        </MAPPING>
        <CONTACT_LISTS>
                <CONTACT_LIST_ID>31279</CONTACT_LIST_ID>
                <CONTACT_LIST_ID>54564</CONTACT_LIST_ID>
        </CONTACT_LISTS>
</LIST_IMPORT>
```

Example 3	This example shows a mapping file for a database with no unique key where the columns EMAIL and Customer Id are being used to update contacts.

```
<LIST_IMPORT>
        <LIST_INFO>
                <ACTION>UPDATE_ONLY</ACTION>
                <LIST_NAME>Premier Accts</LIST_NAME>
                <LIST_VISIBILITY>0</LIST_VISIBILITY>
                <FILE_TYPE>0</FILE_TYPE>
                <HASHEADERS>true</HASHEADERS>
        </LIST_INFO>
        <SYNC_FIELDS>
                <SYNC_FIELD>
                        <NAME>EMAIL</NAME>
                </SYNC_FIELD>
                <SYNC_FIELD>
                        <NAME>Customer Id</NAME>
                </SYNC_FIELD>
        </SYNC_FIELDS>
        <MAPPING>
                <COLUMN>
                        <INDEX>1</INDEX>
                        <NAME>EMAIL</NAME>
                        <INCLUDE>true</INCLUDE>
                </COLUMN>
                <COLUMN>
                        <INDEX>2</INDEX>
                        <NAME>Customer Id</NAME>
                        <INCLUDE>true</INCLUDE>
                </COLUMN>
                <COLUMN>
                        <INDEX>3</INDEX>
                        <NAME>First_Name</NAME>
                        <INCLUDE>true</INCLUDE>
                </COLUMN>
        </MAPPING>
</LIST_IMPORT>
```

Export from a Database

ExportList

This interface exports contact data from a database, query, or contact list. Engage exports the results to a CSV file, then adds that file to the FTP account associated with the current session. You can copy the results file to **Stored Files** in Engage.

Note: If the database is a Double Opt-in database, Engage only exports confirmed contacts. In addition, the database security settings that allow you to use this operation are, by default, disabled for all **Double Opt-In** databases.

Operation	**<ExportList>**		
Elements	LIST_ID	Unique identifier for the database, query, or contact list Engage is exporting.	
	EMAIL	Optional	If specified, this email address receives notification when the job is complete.
	EXPORT_TYPE	Specifies which contacts to export. Valid values are: • ALL – export entire database. • OPT_IN – export only currently opted-in contacts. • OPT_OUT – export only currently opted-out contacts. • UNDELIVERABLE – export only contacts who are currently marked as undeliverable.	
	EXPORT_FORMAT	Specifies the format (file type) for the exported data. Valid values are: • CSV – create a comma-separated values file • TAB – create a tab-separated values file • PIPE – create a pipe-separated values file	
	FILE_ENCODING	Optional	Defines the encoding of the exported file. Supported values are: • utf-8 • iso-8859-1 If not specified, Engage uses the Organization default encoding.
	ADD_TO_STORED_FILES	Optional	Use the ADD_TO_STORED_FILES parameter to write the output to the **Stored Files** folder within Engage. If you omit the ADD_TO_STORED_FILES parameter, Engage will move exported files to the **download** directory of the user's FTP space.
	DATE_START	Optional	Specifies the beginning boundary of information to export (relative to the last modified date). If time is included, it must be in 24-hour format.
	DATE_END	Optional	Specifies the ending boundary of information to export (relative to the last modified date). If time is included, it must be in 24-hour format.
	USE_CREATED_DATE	Optional	If included, the DATE_START and DATE_END range will be relative to the contact create date rather than last modified date.

	INCLUDE_LEAD_SOURCE	Optional	Specifies whether to include the Lead Source column in the resulting file. Supported values are: • true • false If the element is not included in the request, the Lead Source column will not appear in the generated file. ***Note***: This element is only used when not using EXPORT_COLUMNS to export specific database columns.
	LIST_DATE_FORMAT	Optional	Used to specify the date format of the date fields in your exported file if date format differs from "mm/dd/yyyy" (month, day, and year can be in any order you choose). Valid values for Month are: • mm (e.g. 01) • m (e.g. 1) • mon (e.g. Jan) • month (e.g. January) Valid values for Day are: • dd (e.g. 02) • d (e.g. 2) Valid values for Year are: • yyyy (e.g. 1999) • yy (e.g. 99) Separators may be up to two characters in length and can consist of periods, commas, question marks, spaces, and forward slashes (/). **Examples:** • If dates in your file are formatted as "Jan 2, 1975" your LIST_DATE_FORMAT would be "mon d, yyyy". • If dates in your file are formatted as "1975/09/02" your LIST_DATE_FORMAT would be "yyyy/mm/dd".
	EXPORT_COLUMNS	Optional	XML node used to request specific custom database columns to export for each contact. If EXPORT_COLUMNS is not specified, all database columns will be exported.
		COLUMN	The name of a column to export. All custom columns may be specified using their name. The following System columns are also available: • CRM Account ID • CRM Contact Type • CRM Enable Sync • CRM Lead Source • CRM Sync ID • Email • Email Type • Opt In Date • Opt In Details • Opt Out Details • Opted Out

		• Opted Out Date • RECIPIENT_ID • Last Modified Date
Example	<Envelope> <Body> <ExportList> <LIST_ID>59294</LIST_ID> <EXPORT_TYPE>ALL</EXPORT_TYPE> <EXPORT_FORMAT>CSV</EXPORT_FORMAT> <ADD_TO_STORED_FILES/> <DATE_START>07/25/2011 12:12:11</DATE_START> <DATE_END>09/30/2011 14:14:11</DATE_END> <EXPORT_COLUMNS> <COLUMN>FIRST_NAME</COLUMN> <COLUMN>INITIAL</COLUMN> <COLUMN>LAST_NAME</COLUMN> </EXPORT_COLUMNS> </ExportList> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**TRUE** if successful
	JOB_ID	Identifies the Engage Background Job created and scheduled for this import.
	FILE_PATH	Identifies the file name of the exported file.
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <JOB_ID>499600</JOB_ID> <FILE_PATH> /download/Leads – All – Oct 11 2011 03-58- 17PM.CSV </FILE_PATH> </RESULT> </Body> </Envelope>	

Add a Column to a Database

AddListColumn

This interface allows adding a column to an existing Engage Database.

Operation	**<AddListColumn>**			
Elements	LIST_ID	The ID of the Engage database the column is being added to.		
	COLUMN_NAME	The name of the column being added.		
	COLUMN_TYPE	Defines what type of column to create. The following is a list of valid values : 0 – Text column 1 – YES/No column 2 – Numeric column 3 – Date column 4 – Time column 5 – Country column 6 – Select one 8 – Segmenting 13 – SMS Opt In 14 – SMS Opted Out Date 15 – SMS Phone Number 16 – Phone Number 17 – Timestamp 20 – Multi-Select		
	DEFAULT	Specifies the default value for the new column. For a multi-select column, the default value will be a semi-colon delimited list of values.		
	Optional	SELECTION_VALUES	Used to define possible values for 'Select one' and 'Multi'Select' type columns.	
		Child Elements	VALUE	A possible value for a selection column.
Example	<Envelope> <Body> <AddListColumn> <LIST_ID>60700</LIST_ID> <COLUMN_NAME>Interested In</COLUMN_NAME> <COLUMN_TYPE>20</COLUMN_TYPE> <DEFAULT>Sales;Coupons</DEFAULT> <SELECTION_VALUES> <VALUE>Sales</VALUE> <VALUE>Coupons</VALUE> <VALUE>Product News</VALUE> <VALUE>Corporate News</VALUE> </SELECTION_VALUES> </AddListColumn> </Body> </Envelope>			
Response	**<RESULT>**			
Elements	SUCCESS	**TRUE** if successful.		
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> </RESULT> </Body> </Envelope>			

Get Database Details

GetListMetaData

This interface returns the description of a Database, Query, or Relational table.

Operation	**<GetListMetaData>**	
Elements	LIST_ID	Numeric identifier for the entity Engage is exporting.
Example	<Envelope> <Body> <GetListMetaData> <LIST_ID>59294</LIST_ID> </GetListMetaData> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**TRUE** if successful.
	ID	Identifies the Database ID.
	NAME	Returns the name of the database.
	TYPE	Returns the type of database.
	SIZE	Returns the number of contacts in the database.
	NUM_OPT_OUTS	Returns the number contacts opted-out from the database.
	NUM_UNDELIVERABLE	Returns the number of undeliverable contacts.
	LAST_MODIFIED	Returns the date Engage last modified the database.
	LAST_CONFIGURED	Returns the date Engage last configured the database.
	CREATED	Returns the creation date for the database.
	VISIBILITY	Returns the visibility of the database. Valid values are: 0 – Private 1 – Shared
	USER_ID	Returns the ID of the user who created the database.
	ORGANIZATION_ID	Returns the ID for the Organization within which the database was created.
	OPT_IN_FORM_DEFINED	Specifies whether an **Opt-In** form exists for the database. Values are **True** or **False**.
	OPT_OUT_FORM_DEFINED	Specifies whether an **Opt-Out** form exists for the database. Values are **True** or **False**.
	PROFILE_FORM_DEFINED	Specifies whether an **Edit Profile** form exists for the database. Values are **True** or **False**.
	OPT_IN_AUTOREPLY_DEFINED	Specifies whether an **Opt-In Auto-reply** form exists for the database. Values are **True** or **False**.
	PROFILE_AUTOREPLY_DEFINED	Specifies whether an **Edit Profile Auto-reply** form exists for the database. Values are **True** or **False**.
	SMS_KEYWORD	Returns only for SMS databases
	KEY_COLUMNS	Returns the Key Column(s) of the database

Optional Child Elements	COLUMN	XML nodes returning the column name of system and custom fields in the database.		
	Child Elements	NAME	Specifies the field name.	
		DEFAULT_VALUE	Specifies the default value for custom fields. For a multi-select column, the default value will be a semi-colon delimited list of values.	
		TYPE	Specifies the type of custom field. The following is a list of valid values : 0 – Text column 1 – YES/No column 2 – Numeric column 3 – Date column 4 – Time column 5 – Country column 6 – Select one 8 – Segmenting 9 – System (used for defining EMAIL field only) 13 – SMS Opt In 14 – SMS Opted Out Date 15 – SMS Phone Number 16 – Phone Number 17 – Timestamp 20 – Multi-Select	
		SELECTION_VALUES	Optional	A list of possible values when column type is Selection or Multi-Select.
		Child Elements	VALUE	A single value for a Selection or Multi-Select column.

Example

```
<Envelope>
   <Body>
      <RESULT>
         <SUCCESS>TRUE</SUCCESS>
         <ID>108220</ID>
         <NAME>Test3</NAME>
         <TYPE>0</TYPE>
         <SIZE>12</SIZE>
         <NUM_OPT_OUTS>0</NUM_OPT_OUTS>
         <NUM_UNDELIVERABLE>0</NUM_UNDELIVERABLE>
         <LAST_MODIFIED>02/02/06 04:51 PM</LAST_MODIFIED>
         <LAST_CONFIGURED>02/02/06 04:51 PM</LAST_CONFIGURED>
         <CREATED>02/02/06 04:51 PM</CREATED>
         <VISIBILITY>0</VISIBILITY>
       <SMS_KEYWORD/>
         <USER_ID>12c734c-108b610e402-f528764d624db129b32c21fbcaocb8d6</USER_ID>
<ORGANIZATION_ID>113cf49-fc61243bob-f528764d624db129b32c21fbcaocb8d6 </ORGANIZATION_ID>
         <OPT_IN_FORM_DEFINED>false</OPT_IN_FORM_DEFINED>
         <OPT_OUT_FORM_DEFINED>true</OPT_OUT_FORM_DEFINED>
         <PROFILE_FORM_DEFINED>false</PROFILE_FORM_DEFINED>
         <OPT_IN_AUTOREPLY_DEFINED>false</OPT_IN_AUTOREPLY_DEFINED>
         <PROFILE_AUTOREPLY_DEFINED>false</PROFILE_AUTOREPLY_DEFINED>
                 <COLUMNS>
                    <COLUMN>
                        <NAME>LIST_ID</NAME>
                    </COLUMN>
                    <COLUMN>
                        <NAME>MAILING_ID</NAME>
                    </COLUMN>
                    <COLUMN>
                        <NAME>RECIPIENT_ID</NAME>
                    </COLUMN>
```

```
                                <COLUMN>
                                        <NAME>EMAIL</NAME>
                                </COLUMN>
                                <COLUMN>
                                        <NAME>CRM Lead Source</NAME>
                                </COLUMN>
                                <COLUMN>
                                        <NAME>CAR_TYPE</NAME>
                                        <DEFAULT_VALUE>Hybrid</DEFAULT_VALUE>
                                        <TYPE>0</TYPE>
                                </COLUMN>
                        </COLUMNS>
            <KEY_COLUMNS>
                <COLUMN>
                        <NAME>Email</NAME>
                </COLUMN>
            </KEY_COLUMNS>
                </RESULT>
     </Body>
</Envelope>
```

List Contact Mailings

ListRecipientMailings

This interface returns a list of mailings for a specified database (or query) and contact as well as metrics associated with those mailings. **Note:** This operation only returns mailings with associated events (for example, Open, Clickthrough, and Bounce).

Operation	**<ListRecipientMailings>**		
Elements	LIST_ID		Unique ID of the database or query whose mailings you are requesting.
	RECIPIENT_ID		Unique ID of the contact for whom to look up mailings.
Example	<Envelope> <Body> <ListRecipientMailings> <LIST_ID>45654</LIST_ID> <RECIPIENT_ID>360916458</RECIPIENT_ID> </ListRecipientMailings> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS		**True** if successful.
	Mailing		Zero or more XML nodes containing the definition of each mailing. Each node must contain all of the following elements.
	Child Element	MailingName	Name of the mailing (enclosed in CDATA tags).
		MailingId	Unique ID of the mailing.
		SentTS	Time Engage sent the mailing.
		TotalOpens	Total messages opened.
		TotalClickstreams	Total clickstreams clicked.
		TotalClicks	Total links clicked.
		TotalConversions	Total link conversions.
		TotalAttachments	Total attachments opened.

		TotalForwards	Total times message was forwarded.
		TotalMediaPlays	Total times media was played.
		TotalBounces	Total bounces.
		TotalOptOuts	Total times opted out of mailings.
Example			

```
<Envelope>
  <Body>
    <RESULT>
      <SUCCESS>TRUE</SUCCESS>
      <Mailing>
        <MailingId>5644</MailingId>
        <MailingName><![CDATA[Digital Engage (1)]]></MailingName>
        <SentDateTime>6/22/04 10:12 AM</SentDateTime>
        <TotalOpens>7</TotalOpens>
        <TotalClickstreams>0</TotalClickstreams>
        <TotalClicks>5</TotalClicks>
        <TotalConversions>0</TotalConversions>
        <TotalAttachments>0</TotalAttachments>
        <TotalForwards>0</TotalForwards>
        <TotalMediaPlays>0</TotalMediaPlays>
        <TotalBounces>0</TotalBounces>
        <TotalOptOuts>0</TotalOptOuts>
      </Mailing>
    </RESULT>
  </Body>
</Envelope>
```

Remove a Contact

RemoveRecipient

Use this interface to remove a contact from a database or Contact List. If your database has a key other than Email, you must include all unique key columns with their corresponding name/value pairs. If your database has no Unique Identifier defined, one or more Sync Fields must be specified in order to look up the contact.

Operation	**<RemoveRecipient>**		
Elements	LIST_ID	Unique ID of the database or Contact List from which to remove contact.	
	EMAIL	Email address of the contact to remove.	
	COLUMN	Optional	Required if the database has a key other than Email or no Unique Identifier. XML nodes defining the column name and value used to look up a contact in a database or Contact List. You must include all unique key columns with their corresponding name/value pairs when using this operation on a database having a key other than EMAIL. ***Note:*** If more than one contact is found matching the lookup columns, all contacts will be removed.
	Child Element	Name	Specifies the field name of a field which is part of the unique key for the database.
		Value	Specifies the value for the field.
Example			

```
<Envelope>
  <Body>
    <RemoveRecipient>
```

	<LIST_ID>10000</LIST_ID> <EMAIL>somebody@domain.com</EMAIL> <COLUMN> <NAME>Customer Id</NAME> <VALUE>123-45-6789</VALUE> </COLUMN> </RemoveRecipient> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**True** if successful.
	ORGANIZATION_ID	Identifies your Engage Organization.
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> </RESULT> <ORGANIZATION_ID> 1dcd49d-108b594203d-f528764d648fb129b32c21fbca0cb8d6 </ORGANIZATION_ID> </Body> </Envelope>	
Error Codes	133	Contact does not exist in the database or Contact List.
	108	The Database or Contact List ID provided does not exist.
	140	Permission denied to the provided Database or Contact List.
	145	User session is invalid or has expired.

Get a List of Databases

GetLists

This interface extracts a list of databases for an organization.

Operation	**<GetLists>**		
Elements	VISIBILITY	Required. Defines the visibility of the databases to return. 0 – Private 1 – Shared	
	LIST_TYPE	Defines the type of entity to return. Supported values are: 0 – Databases 1 – Queries 2 – Databases, Contact Lists and Queries 5 – Test Lists 6 – Seed Lists 13 – Suppression Lists 15 – Relational Tables 18 – Contact Lists	
	FOLDER_ID	Optional parameter to specify a particular folder from which to return databases.	
	INCLUDE_ALL_LISTS	Optional parameter to specify that Engage return all databases within subfolders. Valid values are **True** and **False**. **Note:** Engage will ignore the FOLDER_ID element if set to **True**.	
	INCLUDE_TAGS	Optional parameter to return all Tags associated with the database.	
Example	<Envelope> <Body> <GetLists> <VISIBILITY>0</VISIBILITY> <LIST_TYPE>2</LIST_TYPE> </GetLists> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS	**True** if successful.	
	LIST	XML nodes defining each database.	
	Child Element	ID	Returns the unique ID of the entity.
		NAME	Returns the name of the entity.
		TYPE	Returns the type. Possible values are: 0 – Databases 1 – Queries 2 – Both Databases and Queries 5 – Test Lists 6 – Seed Lists 13 – Suppression Lists 15 – Relational Tables

		SIZE	Returns the number of contacts in the database.	
		NUM_OPT_OUTS	Returns the number of opted-out contacts in the database.	
		NUM_UNDELIVERABLE	Returns the number of undeliverable contacts in the database.	
		LAST_MODIFIED	Returns the date and time when the database was last modified.	
		VISIBILITY	Returns the visibility of the database (for example, Shared or Private).	
		PARENT_NAME	Returns the name of the associated parent database.	
		USER_ID	Returns the User ID for the database owner.	
		PARENT_FOLDER_ID	Returns the ID of the folder within which the database resides.	
		IS_FOLDER	Returns **True** if the entity is a folder rather than a database, query, or relational table.	
		FLAGGED_FOR_BACKUP	Indicates whether the Database is flagged to be exported using the Automated Backup functionality.	
		SUPPRESSION_LIST_ID	The ID of the associated Suppression List. The element will be empty if there is no associated Suppression List.	
		Tags	If INCLUDE_TAGS is specified in the request, returns any tags associated with the database.	
			Tag	XML nodes containing the Tags associated with the database.

Example

```
<Envelope>
  <Body>
   <RESULT>
        <SUCCESS>TRUE</SUCCESS>
         <LIST>
              <ID>365333</ID>
              <NAME>Folder One</NAME>
              <TYPE>0</TYPE>
              <SIZE>0</SIZE>
              <NUM_OPT_OUTS>0</NUM_OPT_OUTS>
              <NUM_UNDELIVERABLE>0</NUM_UNDELIVERABLE>
              <LAST_MODIFIED>08/23/07 04:18 PM</LAST_MODIFIED>
              <VISIBILITY>0</VISIBILITY>
              <PARENT_NAME/>
              <USER_ID>8c3747-111fae2b32c21fbca0cb8d6</USER_ID>
              <PARENT_FOLDER_ID>285607</PARENT_FOLDER_ID>
              <IS_FOLDER>true</IS_FOLDER>
        </LIST>
        <LIST>
              <ID>323543</ID>
              <NAME>List One</NAME>
              <TYPE>0</TYPE>
              <SIZE>1</SIZE>
              <NUM_OPT_OUTS>0</NUM_OPT_OUTS>
              <NUM_UNDELIVERABLE>0</NUM_UNDELIVERABLE>
              <LAST_MODIFIED>09/26/07 10:31 AM</LAST_MODIFIED>
              <VISIBILITY>0</VISIBILITY>
              <PARENT_NAME/>
              <USER_ID>8c3747-111fae23972-f520cb8d6</USER_ID>
              <PARENT_FOLDER_ID>285607</PARENT_FOLDER_ID>
              <IS_FOLDER>false</IS_FOLDER>
```

	</LIST> <LIST> <ID>390254</ID> <NAME>List two</NAME> <TYPE>0</TYPE> <SIZE>65</SIZE> <NUM_OPT_OUTS>0</NUM_OPT_OUTS> <NUM_UNDELIVERABLE>0</NUM_UNDELIVERABLE> <LAST_MODIFIED>09/26/07 10:31 AM</LAST_MODIFIED> <VISIBILITY>0</VISIBILITY> <PARENT_NAME/> <USER_ID>8c3747-111ae23d624dbca0cb8d6</USER_ID> <PARENT_FOLDER_ID>285607</PARENT_FOLDER_ID> <IS_FOLDER>false</IS_FOLDER> </LIST> </RESULT> </Body> </Envelope>

Create a Relational Table

CreateTable

This interface creates a Relational Table in Engage.

Operation	<CreateTable>				
Elements	TABLE_NAME	Defines the name of the new table.			
	COLUMNS	XML nodes defining the table columns you are inserting/updating. **Note:** Unique key columns must be part of the submission with column names and values.			
	Elements	COLUMN			
		Child Elements	NAME	Defines the name of the column.	
			TYPE	Defines what type of column to create. The following is a list of valid values : • TEXT • YESNO • NUMERIC • DATE • TIME • COUNTRY • SELECTION • EMAIL • DATE_TIME • SYNC_ID	
			IS_REQUIRED	Defines whether to create the column as "required". Valid values are **True** and **False**.	
			KEY_COLUMN	Added to field definition and defines a field as a unique key for the table when set to **True**. You can define more than one unique field for each table.	
			DEFAULT_VALUE	Defines the default value of the column if the row does not contain a value for this column.	
			SELECTION_VALUES	Defines values in a drop-down list. You must provide at least one VALUE element for when **TYPE = SELECTION**.	
				VALUE	A single value in a drop-down list.

Example

```xml
<Envelope>
    <Body>
        <CreateTable>
            <TABLE_NAME>Purchases</TABLE_NAME>
            <COLUMNS>
                <COLUMN>
                    <NAME>Record Id</NAME>
                    <TYPE>NUMERIC</TYPE>
                    <IS_REQUIRED>true</IS_REQUIRED>
                    <KEY_COLUMN>true</KEY_COLUMN>
                </COLUMN>
                <COLUMN>
                    <NAME>Purchase Date</NAME>
                    <TYPE>DATE</TYPE>
                    <IS_REQUIRED>true</IS_REQUIRED>
    </COLUMN>
                <COLUMN>
                    <NAME>Product Id</NAME>
                    <TYPE>NUMERIC</TYPE>
                    <IS_REQUIRED>true</IS_REQUIRED>
                </COLUMN>
            </COLUMNS>
        </CreateTable>
    </Body>
</Envelope>
```

Response	**<RESULT>**	
Elements	SUCCESS	**TRUE** if successful
	TABLE_ID	Identifies the Id of the relational table created in Engage.

Example

```xml
<Envelope>
    <Body>
        <RESULT>
            <SUCCESS>TRUE</SUCCESS>
            <TABLE_ID>499600</TABLE_ID>
        </RESULT>
    </Body>
</Envelope>
```

Associate Relational Data with Contacts in a Database

JoinTable

This interface supports associating a Database with a Relational Table. You must specify a table (name or ID) and database (name or ID) along with the contact columns making the association. Below is the syntax for the API request and response:

Operation	**<JoinTable>**		
Elements	TABLE_NAME	Optional parameter to specify the folder path and name of the Relational Table you are associating. Either TABLE_NAME or TABLE_ID is required. **Note:** If you specify TABLE_VISIBILITY, Engage will ignore Private / Shared in TABLE_NAME.	
	TABLE_ID	Optional parameter to specify the ID of the Relational Table you are joining. Either TABLE_NAME or TABLE_ID is required. **Note:** Private/Shared is not specified as part of the path.	
	TABLE_VISIBILITY	Optional field defining the visibility of the table you are referencing. Required if not specified in TABLE_NAME. 0 – Private 1 – Shared	
	LIST_NAME	Optional parameter specifying the folder path and name of the database you are associating. Either LIST_NAME or LIST_ID is required. **Note:** If you specify LIST_VISIBILITY, Engage will ignore Private / Shared in LIST_NAME.	
	LIST_ID	Optional parameter to specify the ID of the database you are associating. Either LIST_NAME or LIST_ID is required.	
	LIST_VISIBILITY	Optional field defining the visibility of the database you are referencing. Required if not specified in LIST_NAME. 0 – Private 1 – Shared	
	REMOVE	Optional Boolean element used to remove the relationship between a database and a Relational Table. If specified, Engage ignores any MAP_FIELD elements.	
	EMAIL	Optional parameter to specify an email address for notification of job completion or errors.	
Child Element	MAP_FIELD		
	Child Elements	LIST_FIELD	Contains the field name in the Engage database.
		TABLE_FIELD	Contains the field name in the Relational Table.

Example	<Envelope> <Body> <JoinTable> <TABLE_NAME>Shared/Web Analytics/Purchases</TABLE_NAME> <LIST_ID>65745</LIST_ID> <MAP_FIELD> <LIST_FIELD>ItemID</LIST_FIELD> <TABLE_FIELD>Item ID</TABLE_FIELD> </MAP_FIELD> <MAP_FIELD> <LIST_FIELD>PurchPrice</LIST_FIELD> <TABLE_FIELD>Purchase Price</TABLE_FIELD> </MAP_FIELD> </JoinTable> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**TRUE** if successful
	JOB_ID	Identifies the Engage Background Job created and scheduled for this process.
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <JOB_ID>108518</JOB_ID> </RESULT> </Body> </Envelope>	

Insert and Update Records in a Relational Table

InsertUpdateRelationalTable

This interface inserts or updates relational data.

For each Row that is passed in:

- If a row is found having the same key as the passed in row, update the record.
- If no matching row is found, insert a new row setting the column values to those passed in the request.

When rows are inserted or updated in the relational table, all Column values are set based on the values passed in the COLUMN elements.

Note: Only one hundred rows may be passed in a single InsertUpdateRelationalTable call.

Operation	**<InsertUpdateRelationalTable>**				
Elements	TABLE_ID	The Id of the Engage Table that will be updated.			
	ROWS	XML node containing ROW elements for each row being inserted or updated.			
	Child Elements	ROW	XML nodes defining a relational table row and its columns.		
		Child Elements	COLUMN	The value of the Relational Table Column being inserted or updated.	
			Attributes	name	The name of the Relational Table Column being inserted or updated.
Example					

```
<Envelope>
    <Body>
        <InsertUpdateRelationalTable>
            <TABLE_ID>86767</TABLE_ID>
            <ROWS>
                <ROW>
                    <COLUMN name="Record Id"><![CDATA[GHbjh73643hsdiy]]></COLUMN>
                    <COLUMN name="Purchase Date"><![CDATA[01/09/1975]]></COLUMN>
                    <COLUMN name="Product Id"><![CDATA[123454]]></COLUMN>
                </ROW>
                <ROW>
                    <COLUMN name="Record Id"><![CDATA[WStfh73643hsdgw]]></COLUMN>
                    <COLUMN name="Purchase Date"><![CDATA[02/11/1980]]></COLUMN>
                    <COLUMN name="Product Id"><![CDATA[45789]]></COLUMN>
                </ROW>
                <ROW>
                    <COLUMN name=" Record Id"><![CDATA[Yuhbh73643hsfgh]]></COLUMN>
                    <COLUMN name="Purchase Date"><![CDATA[05/10/1980]]></COLUMN>
                    <COLUMN name="Product Id"><![CDATA[4766454]]></COLUMN>
                </ROW>
            </ROWS>
        </InsertUpdateRelationalTable>
    </Body>
</Envelope>
```

Response	**<RESULT>**			
Elements	SUCCESS	**true** if the request is successfully received and processed.		
	FAILURES	XML node containing FAILURE elements for each row that could NOT be inserted or updated. This element and its children will not be included if no failures are encountered.		
	Child Elements	FAILURE	Provides a type and description of each failure.	
	Attributes	failure_type	Identifies whether the row could not be inserted or updated for a reason that is temporary. Valid values are: • transient • permanent	
		description	The description of the error.	
		Child Elements	COLUMN	The value of the Relational Table Column being inserted or updated. A COLUMN will be returned for each field of the record that could not be inserted.
		Attributes	name	The name of the Relational Table Column being inserted or updated that was not successful.
Example	<Envelope> <Body> <RESULT> <SUCCESS>true</SUCCESS> <FAILURES> <FAILURE failure_type=”permanent” description=”Invalid Date Value in column: Purchase Date”> <COLUMN name=”Record Id”>WStfh73643hsdgw</COLUMN> </FAILURE> <FAILURE failure_type=”permanent” description=”Invalid Number in column: Price”> <COLUMN name=”Record Id”>GHbjh73643hsdiy</COLUMN> </FAILURE> </FAILURES> </RESULT> </Body> </Envelope>			

Delete Records From a Relational Table

DeleteRelationalTableData

This interface deletes records from a relational table.

Operation	**<DeleteRelationalTableData>**				
Elements	TABLE_ID	The Id of the Engage Table that will be updated.			
	ROWS	XML node containing ROW elements for each row being deleted.			
	Child Elements	ROW	XML nodes defining a relational table row and its key columns.		
		Child Elements	KEY_COLUMN	The value of a Relational Table Key Column for the row being deleted. If the table has multiple columns defined as key columns, each ROW will have multiple COLUMN elements. All Key Columns must be provided for each ROW.	
			Attributes	name	The name of a Relational Table Key Column for the row being deleted.
Example	<Envelope> <Body> <DeleteRelationalTableData> <TABLE_ID>56753246</TABLE_ID> <ROWS> <ROW> <KEY_COLUMN name="Record Id"><![CDATA[GHbjh73643hsdiy]]></KEY_COLUMN> </ROW> <ROW> <KEY_COLUMN name="Record Id"><![CDATA[WStfh73643hsdgw]]></KEY_COLUMN> </ROW> </ROWS> </DeleteRelationalTableData> </Body> </Envelope>				

Response	**<RESULT>**			
Elements	SUCCESS	**true** if any rows are deleted.		
	FAILURES	XML node containing FAILURE elements for each row that could NOT be deleted.		
	Child Elements	FAILURE	Provides a description of each failure.	
	Attributes	failure_type	Identifies whether the row could not be deleted for a reason that is temporary. Valid values are: • transient • permanent	
		description	The description of the error.	
		Child Elements	KEY_COLUMN	The value of a Relational Table Key Column for the row being deleted. A KEY_COLUMN will be returned for each key field of the record that could not be inserted.
		Attributes	name	The name of the Relational Table Key Column being deleted that was not successful.
Example	<Envelope> <Body> <RESULT> <SUCCESS>true</SUCCESS> <FAILURES> <FAILURE failure_type="transient" description="Database Deadlock"> <KEY_COLUMN name="Record Id">WStfh73643hsdgw</KEY_COLUMN> </FAILURE> <FAILURE failure_type="permanent" description="Row Not Found"> <KEY_COLUMN name="Record Id">GHbjh73643hsdiy</KEY_COLUMN> </FAILURE> </FAILURES> </RESULT> </Body> </Envelope>			

Import to a Relational Table

ImportTable

This interface is used for programmatically creating or updating a Relational Table in Engage. This operation requires a mapping file and source file stored on the FTP server related to the Engage account used to Login. Once you upload the Relational Table source and mapping files to the FTP server, you can make an ImportTable API call to launch the job.

Notes: This operation does not associate the relational table to a database. See "Associate Relational Data to Contacts in a Database" for creating the association.

The table below shows the syntax for the API request and response.

Operation	**<ImportTable>**	
Elements	MAP_FILE	The name of the Mapping file in the **upload** directory on the FTP server to use for the import.
	SOURCE_FILE	The name of the file containing the Relational Table data to use in the import. This file must reside in the **upload** directory on the FTP server.
Example	<Envelope> <Body> <ImportTable> <MAP_FILE>table_import_map.xml</MAP_FILE> <SOURCE_FILE>table_create.csv</SOURCE_FILE> </ImportTable> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**True** if successful.
	JOB_ID	Identifies the Engage Background Job created and scheduled for this import.
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <JOB_ID>108518</JOB_ID> </RESULT> </Body> </Envelope>	

Relational Table Definition and Column Mapping Format

Mapping is the process of associating columns in the import file with the corresponding fields in the Relational Table. For example, **Column 1** in the file corresponds to **Purchase Date** in the **Purchases Relational Table**.

The Relational Table Mapping XML file is broken into three main sections:

TABLE_INFO – defines the information about the table you are creating or updating, as well as the action to perform on this table.

COLUMNS – consists of one or more COLUMN elements. Each COLUMN element contains child elements to define the attributes for that column (for example, name, type, and key column).

MAPPING – also consists of COLUMN elements. Each COLUMN element contains child elements to define the attributes for mapping that column (for example, name or index).

Note: All column names are case-sensitive. The Mapping file and Relational Data file field names must exactly match the case and content in the existing Relational Table.

The table below describes each section to include with a TABLE_IMPORT element:

Section	**<TABLE_INFO>**	The TABLE_INFO section defines information about the table you are creating or updating, as well as the action to perform on this table.
Elements	ACTION	Defines the type of Relational Table import you are executing. Valid values (and the actions they perform) are: CREATE – create a new Relational Table;. ADD_ONLY – only add new records to the table. Engage ignores existing (duplicate) records in the source file. UPDATE_ONLY – only update existing records in the table. Engage ignores records found in the source file but not in the Relational Table. ADD_AND_UPDATE – process all records in the source file. If the item already exists in the Relational Table, update specified field values. If the item does not exist, create a new record in the Relational Table. REPLACE – deletes all records from the Relational Table and adds new records to the table.
	TABLE_NAME	Defines the name of the new Relational Table if the ACTION is CREATE (table names must be unique).
	TABLE_ID	Defines the Id of the Relational Table for actions other than CREATE.
	TABLE_VISIBILITY	Defines the visibility of the table. If not included, Engage will set to Private. 0 – Private 1 – Shared
	FILE_TYPE	Defines the formatting of the source file. Supported values are: 0 – CSV file 1 – Tab-separated file 2 – Pipe-separated file
	HASHEADERS	The HASHEADERS element is set to **True** if the first line in the source file contains column definitions.
Section	**<COLUMNS>**	The COLUMNS section defines 0 or more columns to write to within the Relational Table. Define each table column in its own COLUMN element.

Elements	COLUMN		
	Child Elements	NAME	Defines the name of the column.
		TYPE	Defines what type of column to create. The valid values are: 0 – Text column 1 – Yes/No column 2 – Numeric column 3 – Date column 4 – Time column 5 – Country column 6 – Select One 7 – Email 16 – Phone Number 17 – Date/Time 19 – Sync ID
		IS_REQUIRED	Defines whether to create the column as **Required** when contacts are providing information through a form. Valid values are **True** and **False**. **Note:** The import process does use this to validate the data.
		KEY_COLUMN	Added to field definition and set to **True** to define a field as a unique key for the Relational Table. You can define more than one unique field for the table to create a composite key.
		DEFAULT_VALUE	Defines the default value of the column used by the **Send** process if the record does not contain a value for this column.
Section	**<MAPPING>**		The MAPPING section tells the Relational Table Import process which columns in the source file map to which columns in the Engage Relational Table. Define each table column in its own COLUMN element.
Elements	COLUMN		
	Child Elements	INDEX	The INDEX element defines the order within which to define this column in the source file.
		NAME	The name of the TABLE column to which to map the column in the source file.
		INCLUDE	Defines if this element is included in the table import. You do not need to import all columns from the source file. This element allows you to direct the Relational Table Import process to skip the columns you don't want to include. Valid values are **True** and **False**.

Example Mapping File

Example	This example shows a mapping file for creating a new Relational Table related to an existing list. Customer ID and Purchase Date make up the key for the new table:

```
<TABLE_IMPORT>
	<TABLE_INFO>
		<ACTION>CREATE</ACTION>
		<TABLE_NAME>Purchases</TABLE_NAME>
		<FILE_TYPE>0</FILE_TYPE>
		<HASHEADERS>true</HASHEADERS>
	</TABLE_INFO>
	<COLUMNS>
		<COLUMN>
			<NAME>Customer ID</NAME>
			<TYPE>0</TYPE>
			<IS_REQUIRED>true</IS_REQUIRED>
			<KEY_COLUMN>true</KEY_COLUMN>
		</COLUMN>
		<COLUMN>
			<NAME>Purchase Date</NAME>
			<TYPE>3</TYPE>
			<IS_REQUIRED>true</IS_REQUIRED>
			<KEY_COLUMN>true</KEY_COLUMN>
		</COLUMN>
		<COLUMN>
			<NAME>Product ID</NAME>
			<TYPE>0</TYPE>
			<IS_REQUIRED>true</IS_REQUIRED>
			<DEFAULT_VALUE/>
		</COLUMN>
	</COLUMNS>
	<MAPPING>
		<COLUMN>
			<INDEX>1</INDEX>
			<NAME>Customer ID</NAME>
			<INCLUDE>true</INCLUDE>
		</COLUMN>
		<COLUMN>
			<INDEX>2</INDEX>
			<NAME>Purchase Date</NAME>
			<INCLUDE>true</INCLUDE>
		</COLUMN>
		<COLUMN>
			<INDEX>3</INDEX>
			<NAME>Product ID</NAME>
			<INCLUDE>true</INCLUDE>
		</COLUMN>
	</MAPPING>
</TABLE_IMPORT>
```

Export from a Relational Table

ExportTable

This interface supports programmatically exporting Relational Table data from Engage into a CSV file, which Engage uploads to the FTP account or to the Stored Files directory associated with the session.

Below is the syntax for the API request and response:

Operation	**<ExportTable>**		
Elements	TABLE_NAME	Optional parameter specifying the folder path and name of the Relational Table you are exporting. Either TABLE_NAME or TABLE_ID is required. **Note:** If you specify TABLE_VISIBILITY, Engage ignores Private / Shared in TABLE_NAME.	
	TABLE_ID	Optional parameter to specify the ID of the Relational Table you are exporting. Either TABLE_NAME or TABLE_ID is required. **Note:** Engage does not specify Private or Shared as part of the path. ***Note***: Providing the ID of a Relational Table query will export all rows in the table returned by specified query.	
	TABLE_VISIBILITY	Optional field defining the visibility of the table you are referencing. Required if not specified in TABLE_NAME. 0 – Private 1 – Shared	
	EMAIL	Optional	If specified, Engage sends a notification email to the provided email address when the job is complete.
	EXPORT_FORMAT	Specifies the file format for the exported data. Valid values are: • CSV – create a comma-separated values file • TAB – create a tab-separated values file • PIPE – create a pipe-separated values file	
	FILE_ENCODING	Optional	Defines the encoding of the exported file. Supported values are: • utf-8 • iso-8859-1 If not specified, Engage uses the Organization default encoding.
	ADD_TO_STORED_FILES	Optional	Use the ADD_TO_STORED_FILES parameter to write output to the **Stored Files** folder in Engage. If you omit this parameter, Engage moves exported files to the **download** directory of your FTP account.
	DATE_START	Optional	Specifies the beginning boundary of information to export (relative to the last modified date of the contact). If time is included, it must be in 24-hour format.
	DATE_END	Optional	Specifies the ending boundary of information to export (relative to the last modified date of the contact). If time is included, it must be in 24-hour format.

Example	<Envelope> <Body> <ExportTable> <TABLE_NAME>59294</TABLE_NAME> <EXPORT_FORMAT>CSV</EXPORT_FORMAT> <ADD_TO_STORED_FILES/> <DATE_START>07/25/2011 12:12:11</DATE_START> <DATE_END>09/30/2011 14:14:11</DATE_END> </ExportTable> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**TRUE** if successful
	JOB_ID	Identifies the Engage Background Job created and scheduled for this export.
	FILE_PATH	Returns the file name of the export file.
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <JOB_ID>499600</JOB_ID> <FILE_PATH> /download/Purchases – Oct 11 2011 03-58- 17PM.CSV </FILE_PATH> </RESULT> </Body> </Envelope>	

Purge Data from a Relational Table

PurgeTable

This interface supports programmatically deleting Relational Table data from Engage. You can purge all data (or specify a date range using "Delete Before"). The table below shows the **Purge Table API** syntax.

Operation	**<PurgeTable>**		
Elements	TABLE_NAME	Parameter to specify the folder path and name of the Relational Table you are deleting. **Note:** Requires either **TABLE_NAME** or **TABLE_ID**. If you specify TABLE_VISIBILITY, Engage ignores Private/Shared in TABLE_NAME.	
	TABLE_ID	Parameter specifying the ID of the Relational Table you are deleting. **Note:** Requires either **TABLE_NAME** or **TABLE_ID**. Engage does not specify Private/Shared as part of the path.	
	TABLE_VISIBILITY	Parameter defining the visibility of the table you are referencing. Required if not specified in TABLE_NAME. 0 – Private 1 – Shared	
	DELETE_BEFORE	Optional	If included, only data last modified before the specified date/time will be purged.
	EMAIL	Optional	If specified, Engage sends notification to the provided email address when the job is complete.
Example	<Envelope> <Body> <PurgeTable> <TABLE_NAME>Web Analytics Purchases</TABLE_NAME> <TABLE_VISIBILITY>0</TABLE_VISIBILITY> <DELETE_BEFORE>07/25/2011 12:12:11</DELETE_BEFORE> <EMAIL>user@silverpop.com</EMAIL> </PurgeTable> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS	**TRUE** if successful	
	JOB_ID	Identifies the Engage Background Job created and scheduled for this table purge.	
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <JOB_ID>499600</JOB_ID> </RESULT> </Body> </Envelope>		

Delete a Relational Table

DeleteTable

The Delete Table API supports programmatically deleting Relational Table data from Engage. You cannot delete the Relational Table in any of the following cases:

- When the table is in use by active Group of Automated Messages.
- If the table is in use by an active Autoresponder.
- If the table is in use by an active Sending mailing.
- When queries exist that reference the table, but are owned by an associated database.

You can delete the following:

- Table data
- The table entity/structure
- Database/Table associations (joins)
- Relational Table queries

Below is the syntax for the API request and response.

Operation	**<DeleteTable>**		
Elements	TABLE_NAME	Optional parameter to specify the folder path and name of the Relational Table you are deleting. Either TABLE_NAME or TABLE_ID is required. **Note:** If you specify TABLE_VISIBILITY, Engage ignores Private/Shared in TABLE_NAME.	
	TABLE_ID	Optional parameter specifying the ID of the Relational Table you are deleting. Either TABLE_NAME or TABLE_ID is required. **Note:** Engage does not specify Private/Shared as part of the path.	
	TABLE_VISIBILITY	Optional field defining the visibility of the table you are referencing. Required if not specified in TABLE_NAME. 0 – Private 1 – Shared	
	EMAIL	Optional	If specified, Engage sends notification to the provided email address when the job is complete.
Example	<Envelope> <Body> <DeleteTable> <TABLE_NAME>Web Analytics Purchases</TABLE_NAME> <TABLE_VISIBILITY>0</TABLE_VISIBILITY> <EMAIL>user@silverpop.com</EMAIL>		

	</DeleteTable> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**TRUE** if successful
	JOB_ID	Identifies the Engage Background Job created and scheduled for this table deletion.
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <JOB_ID>499600</JOB_ID> </RESULT> </Body> </Envelope>	

Create a Contact List

CreateContactList

This interface creates a new Contact List in Engage.

Operation	**<CreateContactList>**		
Elements	DATABASE_ID	The Id of the database the new Contact List will be associated with.	
	CONTACT_LIST_NAME	The name of the Contact List to be created.	
	VISIBILITY	Defines the visibility of the Contact List being created. 0 – Private 1 – Shared	
	PARENT_FOLDER_ID	Optional	Used to specify the ID of the Contact List folder you wish to place the Contact List in. The specified folder must already exist in the Contact List structure and the user must have access to the folder.
	PARENT_FOLDER_PATH	Optional	Used to specify the path of the Contact List folder you wish to place the Contact List in. The specified folder must already exist in the Contact List structure and the user must have access to the folder.
Example	<Envelope> <Body> <CreateContactList> <DATABASE_ID>565576</DATABASE_ID> <CONTACT_LIST_NAME>Re-marketing Contacts</CONTACT_LIST_NAME> <VISIBILITY>0</VISIBILITY> <PARENT_FOLDER_PATH>Main Folder/Sub Folder</PARENT_FOLDER_PATH> </CreateContactList> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS	**True** if successful.	
	CONTACT_LIST_ID	The Id of the Contact List that was created.	
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <CONTACT_LIST_ID>33535067</CONTACT_LIST_ID> </RESULT> </Body> </Envelope>		

Add a Contact to a Contact List

AddContactToContactList

This interface adds one new contact to a Contact List. If the contact is not yet in your database, they may be added using the AddRecipient operation (see the "Add a Contact" section).

The contact may be added to the Contact List by providing either a Contact Id or the key fields from the Database. The Contact Id can be obtained from the RecipientId element in the response of the following operations: AddRecipient, UpdateRecipient, and SelectRecipientData.

If providing key fields and your database has a key other than Email, you must include all unique key columns with their corresponding name/value pairs. If using a database that has no Unique Identifier defined, one or more columns must be specified in order to look up the contact.

Upon looking up the contact in the database using the Contact Id or key fields, Engage will process the contact as follows:

- If the contact is found and does not yet exist in the Contact List, they will be added.
- If the contact is found and already exists in the Contact List, they will be not be added.
- If more than one contact is found matching the Sync Fields for a non-keyed list, an error message will be returned.
- If the contact is not found, an error message will be returned.

Operation	**<AddContactToContactList>**		
Elements	CONTACT_LIST_ID	The ID of the Contact List to which you are adding the contact.	
	CONTACT_ID	Optional	The ID of the contact being added to the Contact List. Either a CONTACT_ID or COLUMN elements must be provided. If CONTACT_ID is provided, any COLUMN elements will be ignored.
	COLUMN	Optional	Required if the database has a key other than Email or no Unique Identifier and Contact Id is not provided. XML nodes defining the column name and value used to look up a contact in a database. Ignored for regular email key databases. **Note:** Unique key columns must be part of the submission with column names and values.
	Child Element	NAME	Contains the field name.
		VALUE	Contains the field value
Example with Contact Id	<Envelope> <Body> <AddContactToContactList>		

	<CONTACT_LIST_ID>85628</CONTACT_LIST_ID> <CONTACT_ID>15674222</CONTACT_ID> </AddContactToContactList> </Body> </Envelope>

Example with COLUMN elements	<Envelope> <Body> <AddContactToContactList> <CONTACT_LIST_ID>85628</CONTACT_LIST_ID> <COLUMN> <NAME>Customer Id</NAME> <VALUE>123-45-6789</VALUE> </COLUMN> <COLUMN> <NAME>First_Name</NAME> <VALUE>Joe</VALUE> </COLUMN> <COLUMN> <NAME>Last_Name</NAME> <VALUE>Smith</VALUE> </COLUMN> </AddContactToContactList> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**True** if successful.
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> </RESULT> </Body> </Envelope>	

Add a Contact to a Program

AddContactToProgram

This interface is used to add a Contact to a Program. The Contact will be added to the beginning of the Program.

This operation will typically be used in conjunction with either the AddRecipient or UpdateRecipient operation. After adding or updating a Contact using those operations, the RecipientId in the response XML can be used in the CONTACT_ID element in a subsequent AddContactToProgram API request.

The contact will not be added if any of the following scenarios exist:

- The specified CONTACT_ID is already an active participant in the Program.
- The specified CONTACT_ID is NOT in the database associated with the Program.
- The specified PROGRAM_ID is Pending and not accepting new Contacts.
- The specified PROGRAM_ID is Inactive and not accepting new Contacts.
- The specified PROGRAM_ID is Completed and not accepting new Contacts.
- The specified PROGRAM_ID has a Last Contact Add Date in the past.
- The specified PROGRAM_ID does not exist.

Operation	**<AddContactToProgram>**	
Elements	PROGRAM_ID	Unique ID of the program.
	CONTACT_ID	Unique ID of the contact being added to the program.
Example	<Envelope> <Body> <AddContactToProgram> <PROGRAM_ID>56753246</PROGRAM_ID> <CONTACT_ID>7657657</CONTACT_ID> </AddContactToProgram> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**True** if successful.

Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> </RESULT> </Body> </Envelope>

Create a Query of a Database

CreateQuery

This interface supports programmatically creating a query of an Engage database. A query can search for values within database columns, relational table columns, and also filter by mailing activity.

Operation	**<CreateQuery>**		
Elements	QUERY_NAME	The name of the new query.	
	PARENT_LIST_ID	The ID of the database being queried.	
	VISIBILITY	Defining the visibility of the new query. Valid values are: 0 – Private 1 – Shared	
	PARENT_FOLDER_ID	Optional	The ID of the folder within which to save the query. If not specified, it will be stored in the root of Shared or Private based on VISIBILITY.
	SELECT_COLUMNS	Optional	A pipe-separated list of fields available for personalization.
	ALLOW_FIELD_CHANGE	Optional	Defines whether to allow users to change the fields available for personalization when editing or creating a new query from this query. If not specified, the default will be 1 (Allow Changes).Valid values are: 0 – Do not allow changes 1 – Allow Changes

Child Element	CRITERIA		Describes the expressions to perform on one or more columns in the database.		
	Child Element	TYPE	Optional	Specifies if criteria are locked or editable. Valid values are: • Locked • Editable If not specified, the default will be marked **Editable**.	
		EXPRESSION	Describes a single expression to perform on a column. One or more EXPRESSION elements can exist within the CRITERIA element.		
		Child Element	TYPE	Optional	The type of evaluation. Valid values are: NE – Numeric DE – Date/Time TE – Text***Note***: TYPE is required when defining database expressions but not specified for relational table expressions.
			COLUMN_NAME	Optional	The name of the database field. ***Note***: COLUMN_NAME is required when defining database expressions but not specified for relational table expressions.
			OPERATORS	Optional	The operator used for comparison. Valid values are: != < > = IS NOT null IS null LIKE NOT like IN NOT in BETWEEN ***Note***: OPERATORS is required when defining database expressions but not specified for relational table expressions.
			VALUES	Optional	The value being compared to the Engage database field. This can include text or a numeric or date value. It can also include a list of values or the name of a Column. If specifying a Database Column, surround with square brackets, such as: [My Text Column]. If specifying a list of values, use the pipe character (|) to separate values in a list. Required for all operators other than **IS NOT null** and **IS null**.

					Note: VALUES is not specified for relational table expressions.	
			TABLE_ID	Optional	The ID of the Relational Table when one or more relational table expressions are being defined.	
			LEFT_PARENS	Optional	Used to specify one or more left parentheses when relating more than one expression. For example, two left parentheses would be indicated as: **((**	
			RIGHT_PARENS	Optional	Used to specify one or more right parentheses when relating more than one expression. For example, two right parentheses would be indicated as: **))**	
			AND_OR	Optional	Used to specify **AND** or **OR** when relating multiple expressions.	
			RT_EXPRESSION	Optional	Describes a single expression to perform on a relational table column. One or more RT_EXPRESSION elements can exist within the EXPRESSION element. ***Note***: The RT_EXPRESSION child elements are the same for database criteria and relational table criteria. See EXPRESSION section within the CRITERIA section for definition.	
			Child Element	TYPE	The type of evaluation. Valid values are: NE – Numeric DE – Date/Time TE – Text	
				COLUMN_ NAME	The name of the relational table field.	
				OPERATO RS	The operator used for comparison. Valid values are: != < > = IS NOT null IS null LIKE NOT like IN NOT in BETWEEN	
				VALUES	Optional	The value being compared to the Engage relational table field. This can include text or a numeric or date value. It can also include a list of

						values or the name of a Column. If specifying a Column, surround with square brackets, such as: [My Text Column]. If specifying a list of values, use the pipe character (|) to separate values in a list. Required for all operators other than **IS NOT null** and **IS null**.
				LEFT_PARENS	Optional	Used to specify one or more left parentheses when relating more than one expression. For example, two left parentheses would be indicated as: **((**
				RIGHT_PARENS	Optional	Used to specify one or more right parentheses when relating more than one expression. For example, two right parentheses would be indicated as: **))**
				AND_OR	Optional	Used to specify **AND** or **OR** when relating multiple expressions.
Child Element	BEHAVIOR	Optional	Optional; defines the **Behavior** section of the query. Filters mailing contacts by their activity.			
	Child Element	OPTION_OPERATOR	Specifies the operation or activity for which you are searching. Valid values are: 100 – Opened 101 – Clicked 102 – Bounced 103 – No Activity			
		TYPE_OPERATOR	Specifies whether to search a particular mailing (or all mailings) for the database. Valid values are: 110 – For Any Mailing (only available if query is on the database designated to store Automated Behavior Updates) 111 – For One Mailing			
		MAILING_ID	Optional	The ID of the mailing within which to search for contact activity.		
		REPORT_ID	Optional	The Report ID that identifies the mailing instance within which to search for contact activity.		
		LINK_NAME	Optional	The name of a link in a mailing. Used to include only contacts who clicked on a particular link in a mailing.		
		WHERE_OPERATOR	Optional	Specifies whether to compare to an activity count, date of activity, or number of days since activity. Valid values are: 120 – Days Since Activity 121 – Activity Date 122 – Activity Count Only used when you specify **For One Mailing** and you do NOT include a **No Activity** parameter.		
		CRITERIA_OPER	Optional	The operator used for comparison. Required if		

		ATOR		WHERE_OPERATOR is included. Valid values are: != < > = >= <= BETWEEN
		VALUES	Optional	The value you are comparing to the contact activity count, date of activity, or number of days since activity. Could be numeric or date value. If operator is BETWEEN, two values must be provided delimited by a pipe. Required if WHERE_OPERATOR is included.

Example

```
<Envelope>
	<Body>
		<CreateQuery>
			<QUERY_NAME>Has_Mobile_30_Zip</QUERY_NAME>
			<PARENT_LIST_ID>17607</PARENT_LIST_ID>
			<VISIBILITY>0</VISIBILITY>
			<PARENT_FOLDER_ID>9512</PARENT_FOLDER_ID>
			<ALLOW_FIELD_CHANGE>0</ALLOW_FIELD_CHANGE>
			<SELECT_COLUMNS>Last_Name|First_Name</SELECT_COLUMNS>
			<CRITERIA>
				<TYPE>editable</TYPE>
				<EXPRESSION>
					<TYPE>TE</TYPE>
					<COLUMN_NAME>Zip</COLUMN_NAME>
					<OPERATORS><![CDATA[BETWEEN]]></OPERATORS>
					<VALUES><![CDATA[30000|30999]]></VALUES>
					<LEFT_PARENS>(</LEFT_PARENS>
				</EXPRESSION>
				<EXPRESSION>
					<AND_OR>AND</AND_OR>
					<TYPE>TE</TYPE>
					<COLUMN_NAME>Mobile</COLUMN_NAME>
					<OPERATORS><![CDATA[IS NOT null]]></OPERATORS>
					<RIGHT_PARENS>)</RIGHT_PARENS>
				</EXPRESSION>
				<EXPRESSION>
					<TABLE_ID>4645</TABLE_ID>
					<AND_OR>AND</AND_OR>
					<LEFT_PARENS>(</LEFT_PARENS>
					<RT_EXPRESSION>
						<TYPE>TE</TYPE>
						<COLUMN_NAME>Purchase Type</COLUMN_NAME>
						<OPERATORS><![CDATA[=]]></OPERATORS>
						<VALUES><![CDATA[New]]></VALUES>
						<LEFT_PARENS>(</LEFT_PARENS>
					</RT_EXPRESSION>
					<RT_EXPRESSION>
						<AND_OR>AND</AND_OR>
						<TYPE>DE</TYPE>
						<COLUMN_NAME>Purchase Date</COLUMN_NAME>
						<OPERATORS><![CDATA[>]]></OPERATORS>
						<VALUES><![CDATA[[Current Date]-3]]></VALUES>
						<RIGHT_PARENS>)</RIGHT_PARENS>
					</RT_EXPRESSION>
				</EXPRESSION>
				<EXPRESSION>
					<TABLE_ID>8581</TABLE_ID>
					<AND_OR>OR</AND_OR>
					<RIGHT_PARENS>)</RIGHT_PARENS>
```

	<RT_EXPRESSION> <TYPE>TE</TYPE> <COLUMN_NAME>Expired</COLUMN_NAME> <OPERATORS><![CDATA[=]]></OPERATORS> <VALUES><![CDATA[true]]></VALUES> </RT_EXPRESSION> </EXPRESSION> </CRITERIA> <BEHAVIOR> <OPTION_OPERATOR>103</OPTION_OPERATOR> <TYPE_OPERATOR>111</TYPE_OPERATOR> <MAILING_ID>669614</MAILING_ID> <REPORT_ID>502993</REPORT_ID> </BEHAVIOR> </CreateQuery> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**TRUE** if successful
	ListId	The ID of the query that was created.
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <ListId>25874</ListId> </RESULT> </Body> </Envelope>	

Calculate the Current Contacts for a Query

CalculateQuery

This interface supports programmatically calculating the number of contacts for a query. A data job is submitted to calculate the query and GetJobStatus must be used to determine whether the data job is complete. You may only call the Calculate Query data job for a particular query if you have not calculated the query size in the last 12 hours.

Below is the syntax for the API request and response:

Operation	**< CalculateQuery >**		
Elements	QUERY_ID	The ID of the Query to be calculated.	
	EMAIL	Optional	If specified, Engage sends a notification email to the provided email address when the job is complete.
Example	<Envelope> <Body> <CalculateQuery> <QUERY_ID>59294</QUERY_ID> <EMAIL>notify@silverpop.com</EMAIL> </CalculateQuery> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS	**TRUE** if successful	
	JOB_ID	Identifies the Engage Background Job created and scheduled for this export.	
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <JOB_ID>499600</JOB_ID> </RESULT> </Body> </Envelope>		

Set a Column Value

SetColumnValue

This interface updates a column value for all records in a database, query, or contact list.

If a query or contact list is specified, contacts in the parent database will be updated if they are in the query or contact list.

Operation	**<SetColumnValue>**		
Elements	LIST_ID	The Id of the database, query, or contact list used to determine what contacts to update.	
	COLUMN_NAME	The name of the database column being updated.	
	COLUMN_VALUE	Optional	The value the database column will be set to. This element is required when the ACTION is Update.
	ACTION	The action to take on the database column. Possible values for ACTION are: • 0 – Reset (sets the value to 'null' or 0 depending on type) • 1 – Update (set to the VALUE passed in) • 2 – Increment (increment by 1)	
	EMAIL	Optional	If specified, this email address receives notification when the job is complete.
Example	<Envelope> <Body> <SetColumnValue> <LIST_ID>140146</LIST_ID> <COLUMN_NAME>State</COLUMN_NAME> <COLUMN_VALUE>GA</COLUMN_VALUE> <ACTION>1</ACTION> <EMAIL>somebody@domain.com</EMAIL> </SetColumnValue> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS	**TRUE** if successful.	
	JOB_ID	Identifies the Engage Background Job created to set column values.	
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <JOB_ID>98784</JOB_ID> </RESULT> </Body> </Envelope>		

Automation - User

Engage supports two great Programs APIs that allow you to discover what Programs are available to a given database and also how an individual contact has interacted with a program.

Get a List of Programs

GetPrograms

This interface extracts a list of Programs for an organization. Standard permissions are applied to this API so that only those programs will be returned that the authenticated user has access to see.

Note:

Operation	**<GetPrograms>**		
Elements	INCLUDE_ACTIVE	Required	Specify to return the Programs that are in Running and Scheduled state. Valid values are: • True • False
	INCLUDE_INACTIVE	Required	Specify to return the Programs that are only in inactive state. Valid values are: • True • False
	CREATED_DATE_RANGE	Optional	Specify to return Programs that are created within specific date range. This date is based on Program create date.
	Child Element	BEGIN_DATE	Specify start of the date range in the format "mm/dd/yyyy"
		END_DATE	Specify end of the date range in the format "mm/dd/yyyy"
	LIST_ID	Optional	Parameter that makes the function only returns Programs for the specified list (Database or Query). The Database List will return all programs tied to that list through its associated queries.
	APPROVED_FOR_SALES	Optional	Specify to return programs that are only Approved for Sales. Note: These are the programs that are available through contact insight.
	INCLUDE_TAGS	Optional	Specify to return programs associated to specific tag(s)
	Child Element	TAG	Specify to return a program associated to a specific tag
Example	<Envelope> <Body> <GetPrograms> <INCLUDE_ACTIVE>True</INCLUDE_ACTIVE> <INCLUDE_INACTIVE>False</INCLUDE_INACTIVE> <CREATED_DATE_RANGE> <BEGIN_DATE>01/01/2012</BEGIN_DATE> <END_DATE>12/12/2012</END_DATE> </CREATED _DATE_RANGE>		

	<INCLUDE_TAGS> <TAG>Lead Gen</TAG> <TAG>January Campaigns</TAG> </INCLUDE_TAGS> <APPROVED_FOR_SALES/> </GetPrograms> </Body> </Envelope>				
Response	**<RESULT>**				
Elements	SUCCESS	**True** if successful.			
	PROGRAMS	XML node defining all returned programs.			
	Child Element	PROGRAM	XML nodes defining each program being returned		
		Child Element	ID	Returns the unique ID of the Program.	
			NAME	Returns the name of the Program.	
			LIST_ID	The Contact Source List Id tied to the Program (database or query)	
			NOTES	Returns the Program notes/description	
			STATE	Returns the status of the Program. Possible values: • Running • Scheduled • Completed • Inactive	
			TIME_ZONE	Returns the program's time zone	
			CREATED	Returns the date when the Program was created	
			LAST_MODIFIED	Returns the date and time when the Program was last modified. (user's time zone)	
			START_DATE	Returns the Program's start date. If the program does not have a start date, this will be null.	
			END_DATE	Returns the Program's end date. If the program does not have a start date, this will be null.	
			LAST_CONTACT_DATE	Returns the Program's date for which it's last contact can be added. This is not always available and may be null.	
			TAGS	Returns TAGS associated with the Program.	
			Child Element	TAG	XML nodes containing the Tags associated with the Program.

Success Example

```xml
<Envelope>
   <Body>
     <RESULT>
          <SUCCESS>TRUE</SUCCESS>
            <PROGRAMS>
        <PROGRAM>
                    <ID>5435</ID>
                    <NAME>Hot Leads Program</NAME>
            <LIST_ID>423126</LIST_ID>
                    <NOTES>This program contains all the leads that are considered actionable</NOTES>
                    <STATE>Running</STATE>
            <CREATED>06/12/11 10:15 PM</CREATED>
                    <LAST_MODIFIED>10/23/11 04:18 PM</LAST_MODIFIED>
                    <START_DATE>10/10/2011</START_DATE>
                    <END_DATE/>
                    <LAST_CONTACT_DATE/>
                    <TAGS>
                    <TAG>January Campaign</TAG>
                    </TAGS>
        </PROGRAM>
          </PROGRAMS>
    </RESULT>
   </Body>
</Envelope>
```

Error Message Response Example

```xml
<Envelope>
    <Body>
                <RESULT>
                <SUCCESS>false</SUCCESS>
                </RESULT>
                <Fault>
                        <Request/>
                        <FaultCode/>
                        <FaultString> The Database or Contact List ID provided does not exist.
</FaultString>
                        <detail>
                                <error>
                                        <errorid>133</errorid>
                                        <module/>
                                        <class>SP.API</class>
                                        <method/>
                                </error>
                        </detail>
                </Fault>
        </Body>
</Envelope>
```

Error Codes	100	Missing required parameter INCLUDE_ACTIVE.
	100	Missing required parameter INCLUDE_INACTIVE.

	106	Invalid LIST_ID.
	150	You do not have permission to use this API.
	191	Invalid START_DATE format.
	191	Invalid END_DATE format.
	655	Invalid CREATED_DATE_RANGE.
	656	Permission denied to the provided Database, Contact List or Query.
	804	Invalid value for required parameter INCLUDE_ACTIVE.
	804	Invalid value for required parameter INCLUDE_INACTIVE.

Get a Contact's Relationships to Programs

GetProgramsByContact

This interface extracts a list of Programs for a given contact or visitor. By default, it returns the information for the contact that is currently active in that program.

Note: Either CONTACT_ID or VISITOR_KEY must be passed to the method to get a valid response. If both are provided, the CONTACT_ID will be used to lookup the Contact. Standard permissions are applied to this API so that only those programs will be returned that the authenticated user has access to see.

Operation	**<GetProgramsByContact>**		
Elements	LIST_ID	Required	The source database or query to which the returned Programs are tied
	INCLUDE_ACTIVE	Required	Specify to return the Programs that are in Running, Scheduled and Completed state. Valid values are • True • False
	INCLUDE_INACTIVE	Required	Specify to return the Programs that are only in inactive state. Valid values are • True • False
	CONTACT_ID	Optional	The Contact Id of the Contact in the Engage DB. Pass this parameter to get all programs associated with a specific contact. **Note**: Either CONTACT_ID or VISITOR_KEY must be passed. If both passed, CONTACT_ID takes precedence.
	VISITOR_KEY	Optional	The Silverpop web tracking visitor key. Pass this parameter to get all programs associated with a specific visitor.

			Note: Either CONTACT_ID or VISITOR_KEY must be passed. If both passed, CONTACT_ID takes precedence.
	INCLUDE_HISTORY	Optional	By default, it returns the information for the contact that is currently active in that program. Include this element if you want to receive the history for contacts been through the program multiple times. Note: there will be a separate response for each trip through the program.
	CREATED_DATE_RANGE	Optional	Specify to return Programs that are created within specific date range. This date is based on Program create date.
	Child Element	BEGIN_DATE	Specify start of the date range in the format "mm/dd/yyyy"
		END_DATE	Specify end of the date range in the format "mm/dd/yyyy"
	APPROVED_FOR_SALES	Optional	Specify to return programs that are only Approved for Sales. **Note**: These are the programs that are available through contact insight.
	INCLUDE_TAGS	Optional	Specify to return programs associated to specific tag(s)
	Child Element	TAG	Specify to return a program associated to a specific tag. **Note:** This will return any program containing the tag. If you include multiple tags in your API call we will return any program containing at least one of the tags. Multiple tags will be treated as or statements.
	INCLUDE_TRACK	Optional	If specified, it will return the Program Track information the participant is currently in.
	INCLUDE_STEP	Optional	If specified, it will return the Program Step information the participant is currently in.
Example	<Envelope> <Body> <GetProgramsByContact> <INCLUDE_ACTIVE>True</INCLUDE_ACTIVE> <INCLUDE_INACTIVE>False</INCLUDE_INACTIVE> <VISITOR_KEY>125632414</VISITOR_KEY> <INCLUDE_HISTORY> <APPROVED_FOR_SALES/> <INCLUDE_TAGS> <TAG>January Campaign</TAG> <TAG>Feb Campaign</TAG> </INCLUDE_TAGS> <INCLUDE_TRACK/> <INCLUDE_STEP/> </GetProgramsByContact> </Body>		

	</Envelope>				
Response	**<RESULT>**				
Elements	SUCCESS	**True** if successful.			
	PROGRAMS	XML nodes defining each program.			
	Child Element	PROGRAM			
		Child Element	PROGRAM_ID	Returns the unique ID of the Program.	
			STATE	Returns the status of the Program. Current Possible values: • Running • Scheduled • Completed • Inactive **Note:** New values could be added in the future	
			NOT_ACTIVE	This element is included in the response IF the contact is not currently active in the program.	
			ENTERED_DATE	This element represents the date that the contact entered the program in the format "mm/dd/yyyy".	
			EXITED_DATE	This element represents the date that the contact exited the program in the format "mm/dd/yyyy". **Note**: • If end track is not used in the program, the element will not have a date value. • If the contact is currently active in the program, the element will not have a date value	
			TRACK	XML node defining the Track of a Program	
			Attribute	Type	Type of a Track • **start** - Defines Start Track • **end** - Defines End Track • **other** – Defines other than Start or End Track
			Child Element	NAME	Name of the Track
				ID	ID of the Track
			STEP	XML node defining the Step of a Program	
			Attribute	Type	Type of a step • **email** • **direct mail** • **telesales** **Note:** New values could be added in the future
			Child Element	NAME	Name of the Step

				ID	ID of the Step

Example

```
<Envelope>
	<Body>
		<RESULT>
			<SUCCESS>TRUE</SUCCESS>
			<PROGRAMS>
				<PROGRAM>
					<PROGRAM_ID>5435</PROGRAM_ID>
					<STATE>Running</STATE>
					<ENTERED_DATE>01/01/2012</ENTERED_DATE>
					<EXITED_DATE/>
					<TRACK type="start">
						<NAME><![CDATA[Initialize]]></NAME>
						<ID>555</ID>
					</TRACK>
					<STEP type="email">
						<NAME><![CDATA[Welcome Email]]></NAME>
						<ID>101</ID>
					</STEP>
				</PROGRAM>
				<PROGRAM>
					<PROGRAM_ID>5436</PROGRAM_ID>
					<STATE>Completed</STATE>
					<NOT_ACTIVE />
					<ENTERED_DATE>01/01/2012</ENTERED_DATE>
					<EXITED_DATE>01/31/2012</EXITED_DATE>
					<TRACK type="end">
						<NAME><![CDATA[End Track]]></NAME>
						<ID>557</ID>
					</TRACK>
				</PROGRAM>
				<PROGRAM>
					<PROGRAM_ID>5437</PROGRAM_ID>
					<STATE>Running</STATE>
					<ENTERED_DATE>01/01/2012</ENTERED_DATE>
					<EXITED_DATE/>
					<TRACK type="other">
						<NAME><![CDATA[Third Track]]></NAME>
						<ID>558</ID>
					</TRACK>
					<STEP type="direct mail">
						<NAME><![CDATA[Customer Event]]></NAME>
						<ID>102</ID>
					</STEP>
				</PROGRAM>
				<PROGRAM>
					<PROGRAM_ID>5437</PROGRAM_ID>
					<STATE>Inactive</STATE>
					<NOT_ACTIVE />
					<ENTERED_DATE>01/01/2012</ENTERED_DATE>
					<EXITED_DATE/>
					<TRACK type="other">
```

<table>
<tr><td></td><td colspan="2">

```
                        <NAME><![CDATA[Track-4]]></NAME>
                        <ID>560</ID>
                    </TRACK>
                    <STEP type="email">
                        <NAME><![CDATA[Renewal Email]]></NAME>
                        <ID>107</ID>
                    </STEP>
                </PROGRAM>
            </PROGRAMS>
        </RESULT>
    </Body>
</Envelope>
```

</td></tr>
<tr><td>Error Message Response Example</td><td colspan="2">

```
<Envelope>
  <Body>
        <RESULT>
        <SUCCESS>false</SUCCESS>
        </RESULT>
        <Fault>
            <Request/>
            <FaultCode/>
            <FaultString>Contact does not exist in the database or Contact
List</FaultString>
            <detail>
                <error>
                    <errorid>133</errorid>
                    <module/>
                    <class>SP.API</class>
                    <method/>
                </error>
            </detail>
        </Fault>
    </Body>
</Envelope>
```

</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>Error Codes</td><td>100</td><td>Missing required parameter INCLUDE_ACTIVE.</td></tr>
<tr><td></td><td>100</td><td>Missing required parameter INCLUDE_INACTIVE.</td></tr>
<tr><td></td><td>100</td><td>Missing required parameter LIST_ID.</td></tr>
<tr><td></td><td>100</td><td>You must provide either the CONTACT_ID or the VISITOR_KEY.</td></tr>
<tr><td></td><td>106</td><td>Invalid LIST_ID.</td></tr>
<tr><td></td><td>124</td><td>Invalid CONTACT_ID.</td></tr>
<tr><td></td><td>128</td><td>Contact does not exist in the Database, Contact List or Query.</td></tr>
<tr><td></td><td>150</td><td>You do not have permission to use this API.</td></tr>
<tr><td></td><td>191</td><td>Invalid START_DATE format.</td></tr>
<tr><td></td><td>191</td><td>Invalid END_DATE format.</td></tr>
</table>

	655	Invalid CREATED_DATE_RANGE.
	656	Permission denied to the provided Database, Contact List or Query.
	804	Invalid value for required parameter INCLUDE_ACTIVE.
	804	Invalid value for required parameter INCLUDE_INACTIVE.
	10000	Visitor Key does not exist in the Database, Contact List or Query.

Scoring Models - User

Engage supports two great Scoring Model APIs that allow you to discover what Scoring Models are available to a given database and also what scores an individual contact has within that Scoring Model.

Get a List of Scoring Models

GetScoringModels

This interface returns Scoring Models for a specific Organization. Standard permissions are applied to this API so that only those Scoring Models will be returned that the authenticated user has access to see. Only Score Components that have been defined for the Model returned will be included in the response. For example, if the Rank section of a Scoring Model has not been defined (as is a possible scenario), then it would not be listed as one of the Score Components that define that returned Scoring Model.

Operation	<GetScoringModels>						
Example	<Envelope> <Body> <GetScoringModels> </GetScoringModels> </Body> </Envelope>						
Response	**<RESULT>**						
Elements	SUCCESS	**TRUE** if successful.					
	MODELS	XML Node defining Scoring Models					
	Child Element	MODEL	XML Node defining each Scoring Model				
		Child Elements	ID	Id of the Scoring Model			
			NAME	Name of the Scoring Model			
			DESCRIPTION	Description of the Scoring Model as described by the user in the UI			
			CREATED	Created Date of the Scoring Model			
			LAST_MODIFIED	Last Modified Date of the Scoring Model			
			SCORE_COMPONENT_TYPES	The components (or sections) that define the Scoring Model.			
			Child Elements	SCORE_COMPONENT_TYPE	XML Node defining each defined component in the model		
				Child Elements	KEY	The name of the component	
					DESCRIPTION	The definition of the score component (Engage's definition, not something the user specifies)	
					VALUES	XML Node defining the possible values that the specified component could return	
					Child	VALUE	A defined

					Elements		value that can be returned from the score component

Example

```
<Envelope>
  <Body>
   <RESULT>
          <SUCCESS>TRUE</SUCCESS>
          <MODELS>
                    <MODEL>
                              <ID>5435</ID>
                              <NAME>LeadStatusScoringModel</NAME>
                              <DESCRIPTION>This model is used to quality leads into SFDC promotions</DESCRIPTION>
                              <CREATED>10/12/11 10:15 PM</CREATED>
                              <LAST_MODIFIED>10/23/11 04:18 PM</LAST_MODIFIED>
                              <SCORE_COMPONENT_TYPES>
                                        <SCORE_COMPONENT_TYPE>
                              <KEY>totalScore</KEY>
                                                  <DESCRIPTION>This is the overall score for the contact</ DESCRIPTION>
                                        </SCORE_COMPONENT_TYPE>
                                        <SCORE_COMPONENT_TYPE>
                        <KEY>behaviorScore</KEY>
                                                  <DESCRIPTION>This is the behavior score for the contact</ DESCRIPTION>
                                        </SCORE_COMPONENT_TYPE>
                                        <SCORE_COMPONENT_TYPE>
                        <KEY>totalRank</KEY>
                                                  <DESCRIPTION>This is the rank for the contact based on their total score</
          DESCRIPTION>
                                                  <VALUES>
                                                            <VALUE>Hot</VALUE>
                                                            <VALUE>Warm</VALUE>
                                                            <VALUE>Cold</VALUE>
                                                  </VALUES>
                                        </SCORE_COMPONENT_TYPE>
                              </SCORE_COMPONENT_TYPES>
                    </MODEL>
            <MODEL>
                              <ID>5435</ID>
                              <NAME>CustomerValueScoringModel</NAME>
                              <DESCRIPTION>This model is used to quality leads into SFDC promotions</DESCRIPTION>
                              <CREATED>10/12/11 10:15 PM</CREATED>
                              <LAST_MODIFIED>10/23/11 04:18 PM</LAST_MODIFIED>
                              <SCORE_COMPONENT_TYPES>
                                        <SCORE_COMPONENT_TYPE>
                              <KEY>totalScore</KEY>
                                                  <DESCRIPTION>This is the overall score for the contact</ DESCRIPTION>
                                        </SCORE_COMPONENT_TYPE>
                                        <SCORE_COMPONENT_TYPE>
                        <KEY>behaviorScore</KEY>
                                                  <DESCRIPTION>This is the behavior score for the contact</ DESCRIPTION>
                                        </SCORE_COMPONENT_TYPE>
                                        <SCORE_COMPONENT_TYPE>
                              <KEY>totalRank</KEY>
                                                  <DESCRIPTION>This is the rank for the contact based on their total score</
          DESCRIPTION>
                                                  <VALUES>
                                                            <VALUE>High</VALUE>
```

		<VALUE>Average</VALUE> <VALUE>Low</VALUE> </VALUES> </SCORE_COMPONENT_TYPE> </SCORE_COMPONENT_TYPES> </MODEL> </MODELS> </RESULT> </Body> </Envelope>
Error Codes	145	User session is invalid or has expired.
	6001	Scoring Models are not available for this ORG
	6002	Scoring Model database is not set for this ORG
	6003	Scoring Models are not available for this user
	6004	There are no Scoring Models defined for this ORG
Error Message Response Example	<Envelope> <Body> <RESULT> <SUCCESS>false</SUCCESS> </RESULT> <Fault> <Request/> <FaultCode/> <FaultString>User session is invalid or has expired.</FaultString> <detail> <error> <errorid>145</errorid> <module/> <class>SP.API</class> <method/> </error> </detail> </Fault> </Body> </Envelope>	

Get a Contact's Scoring Model Scores

GetContactScores

This interface returns the Contact's Scoring Model's values for a particular Contact. Either CONTACT_ID or VISITOR_KEY must be passed to the method to get a valid response. If both are provided, the CONTACT_ID will be used to lookup the Contact. Standard permissions are applied to this API so that only those Scoring Models for a Contact will be returned that the authenticated user has access to see. If the contact has not been scored and has null values in any of the returned Score Components, then the value will be returned as <value/> (see example 2 below).

Operation	<GetContactScores>					
Elements	CONTACT_ID	**Optional**	The Contact Id of the Contact in the Engage DB. Pass this parameter to get their score values.			
	VISITOR_KEY	**Optional**	The Silverpop web tracking visitor key. Pass this parameter to get their score values.			
	MODEL_ID	**Optional**	If specified, the method will return only the score values for the model, otherwise, the method will return all Scoring Models for the specified user			
Example	<Envelope> <Body> <GetContactScores> <CONTACT_ID>3920341</CONTACT_ID> </GetContactScores> </Body> </Envelope>					
Response	**<RESULT>**					
Elements	SUCCESS	**TRUE** if successful.				
	MODELS	XML Node defining Scoring Models				
	Child Element	MODEL	XML Node defining each Scoring Model			
		Child Elements	ID	Id of the Scoring Model		
			NAME	Name of the Scoring Model		
			SCORE_COMPONENTS	The components (or sections) that define the scoring model.		
			Child Elements	SCORE_COMPONENT	XML Node defining each defined Score Component in the model	
				Child Elements	KEY	The name of the Score Component
					VALUE	The value/score of the Score Component
Example 1: Contact with all values populated	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <MODELS> <MODEL> <ID>5435</ID> <NAME>LeadStatusScoringModel</NAME> <SCORE_COMPONENTS> <SCORE_COMPONENT>					

	<KEY>totalScore</KEY> <VALUE>15</VALUE> </SCORE_COMPONENT> <SCORE_COMPONENT> <KEY>behaviorScore</KEY> <VALUE>45</VALUE> </SCORE_COMPONENT> <SCORE_COMPONENT> <KEY>totalRank</KEY> <VALUE>Hot</VALUE> </SCORE_COMPONENT> </SCORE_COMPONENTS> </MODEL> <MODEL> <ID>5435</ID> <NAME>CustomerValueScoringModel</NAME> <DESC>This model is used to quality leads into SFDC promotions</DESC> <SCORE_COMPONENTS> <SCORE_COMPONENT> <KEY>totalScore</KEY> <VALUE>5</VALUE> </SCORE_COMPONENT> <SCORE_COMPONENT> <KEY>behaviorScore</KEY> <VALUE>0</VALUE> </SCORE_COMPONENT> <SCORE_COMPONENT> <KEY>totalRank</KEY> <VALUE>Low</VALUE> </SCORE_COMPONENT> </SCORE_COMPONENTS> </MODEL> </MODELS> </RESULT> </Body> </Envelope>
Example 2: Contact that has some null values	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <MODELS> <MODEL> <ID>5435</ID> <NAME>LeadStatusScoringModel</NAME> <SCORE_COMPONENTS> <SCORE_COMPONENT> <KEY>totalScore</KEY> <VALUE/> </SCORE_COMPONENT> <SCORE_COMPONENT> <KEY>behaviorScore</KEY> <VALUE/> </SCORE_COMPONENT> <SCORE_COMPONENT> <KEY>totalRank</KEY> <VALUE/> </SCORE_COMPONENT> </SCORE_COMPONENTS> </MODEL>

	</MODELS> </RESULT> </Body> </Envelope>	
	51	Invalid XML Request
Error Codes	133	Contact does not exist in the database or Contact List.
	145	User session is invalid or has expired.
	208	<inputValueDescription> must be numeric, but is "<rawInputValue>"
	6004	No scoring model was found for 1
	10001	You must provide either the CONTACT_ID or the VISITOR_KEY
Error Message Response Example	<Envelope> <Body> <RESULT> <SUCCESS>false</SUCCESS> </RESULT> <Fault> <Request/> <FaultCode/> <FaultString>Contact does not exist in the database or Contact List.</FaultString> <detail> <error> <errorid>133</errorid> <module/> <class>SP.API</class> <method/> </error> </detail> </Fault> </Body> </Envelope>	

Reporting Interfaces – User

Engage archives mailing tracking data after 450 days by default. You can modify this setting to reduce the number of days before archiving, but 450 is the maximum setting. For Autoresponders, if the SEND or ACTION date ranges specified are greater than 450 days (or the number of days configured for archiving), you will receive an error message (returned from the call) specifying that the mailing was archived. If the date range spans both archived and available data for the mailing, Engage will return only the data available and will not return archived data.

If your Organization will frequently requests data older than 30-60 days, we recommend leaving the archive setting at 450 days. As a best practice, try to refrain from requesting data older than one year to avoid including archived data in an export date range.

The following glossary clarifies Engage terms, particularly the parameters used by the RawRecipientDataExport and GetAggregateTracking operations:

Term	Definition
Sent	Contact was in the database or query and was NOT suppressed.
Soft Bounces	All soft bounce retries exhausted for the selected mailing.
Hard Bounces	Invalid Inbox at a valid domain.
Total Bounces	Sum of soft bounces and hard bounces.
Received	Email received (as evidenced by a tracked open, click, media play, or other interaction).
Opened	Opens detected in HTML message. Other interaction detected for TEXT or AOL message.
Clicked	Count of Engage links followed.
Clickstreams	Web pages visited after following an Engage link labeled as a Clickstream link.
Unsubscribed	Contacts requesting opt-out through the Engage Opt Out function.
Subscribed	Contacts who opted in using an Engage Opt-in link in the mailing.
Forwarded	Contacts who forwarded the message using Engage's Forward-to-a-Friend (F2F) feature.
Played Media	Contacts who played Flash or video within the message.
Downloaded	Contacts who downloaded or viewed attachments (not embedded).

Term	Definition
Attachment	
Conversions	Online conversions tracked after following an Engage link labeled as a Clickstream link.
Suppressed Recipients	Contacts who were in the database, but to whom mail was not sent due to various suppression reasons.
Reply Abuse Complaint	Contacts who complained through their ISP (for example, AOL's "This is Spam").
Reply Change Address	Replies from contacts requesting an email address change. Tracked but not acted upon.
Reply Mail Blocks	Messages blocked by an ISP.
Reply Mailing Restriction	Messages rejected based on content (likely due to spam rules).
Reply Other	Reply categorized as something other than one of the following: out of office reply, opt out request, abuse complaint, or change of address request.
All Metrics Combined	Export all available metrics into a single .zip file.
All Metrics Combined	Forwards Only – export all available metrics into a single .zip file, but only for F2F contacts.
Aggregate Metrics	Mailing level counts only, no contact email addresses.
Results over time	Export metrics with time interval data (based on Engage's "Results over Time" graph).
Metrics by Top Domain	Exported Top Domains report data.
Dynamic Content Sent	Distribution of content versions based on Dynamic Content rules.
Dynamic Content Click	Tracked clicks on links defined in Dynamic Content blocks (versus links defined in the template).
Inbox Monitoring Details	Received, Bulked, or Not Received in 19 domains tracked by Pivotal Veracity.
Aggregate Clicks	Mailing-level click count, no contact email addresses.
Aggregate Clickstream	Mailing-level clickstream count, no contact email addresses.

Term	Definition
Aggregate Conversions	Mailing-level conversion count, no contact email addresses.
Aggregate Documents	Mailing-level attachment downloads or views, no contact email addresses.
Aggregate Media Plays	Mailing-level media plays, no contact email addresses.

Export Raw Contact Events

RawRecipientDataExport

This interface allows exporting unique contact-level events and creates a .zip file containing a single flat file with all metrics. You can request all (or a subset) of the Event Types. The API provides the ability to specify one of the following:

- One or more mailings
- One or more Mailing/Report ID combinations (for Autoresponders)
- A specific Database (optional: include related queries)
- A specific Group of Automated Messages
- An Event Date Range
- A Mailing Date Range

If Private mailings—not owned by the user calling the API—are explicitly specified or are determined based on a specified Database, Group of Automated Messages, or Date Range, the events associated with that mailing will not be included in the resulting file. No error will be returned and all other mailing events will be included in the file.

When exporting events within a date range, you can filter them by Mailing Type (for example, Automated Messages). This API provides the ability to export all events not yet exported. This allows exporting on a recurring basis without specifying date ranges. Engage returns any event not previously exported by the user.

The following table shows Input XML definitions.

Operation	**<RawRecipientDataExport>**		
Elements	MAILING_ID	Optional	ID of the mailing for which to export events. You can specify more than one mailing by surrounding each Mailing ID with <MAILING> parent element tags. For Example, if you want to specify more than one mailing to extract data from use the following as a guide: <MAILING> <MAILING_ID>5262609</MAILING_ID> </MAILING> <MAILING> <MAILING_ID>4667603</MAILING_ID> </MAILING>
	REPORT_ID	Optional	Often used with a Group of Automated Messages to gather metrics for a certain instance of the mailing. You can specify more than one Mailing or Report ID by surrounding each pair with a <MAILING> element. For Example, if you want to specify more than one mailing to extract data from use the following as a guide: <MAILING> <REPORT_ID>5262609</REPORT _ID> </MAILING> <MAILING> <REPORT _ID>4667603</REPORT _ID> </MAILING> **NOTE:** Depending on the type of mailing, you can assign Report IDs in a number of ways. For Autoresponders, a single Report ID is associated with every mailing for a day. For a recurring Automated Message, a single Report ID is associated with each occurrence of the mailing. For a standard mailing, there is a one-to-one relationship between a Report ID and Mailing ID.
	CAMPAIGN_ID	Optional	ID for the Group of Automated Messages for which to export events. When specified, Engage will return data for all mailings associated with the Automated Message . Do not use the group's ID and Mailing ID at the same time.
	LIST_ID	Optional	Used as an alternative to Mailing ID or Group of Automated Messages. When specified, Engage will return data for all mailings associated with the Database ID or Query ID. Do not use Database ID, Mailing ID, or Group of Automated Messages at the same time.
	INCLUDE_CHILDREN	Optional	If you provide a Database ID, this element will allow retrieving mailings for queries and contact lists based on the specified database.
	ALL_NON_EXPORTED	Optional	Only includes the events not exported by the user calling the API. Can use with a particular Mailing, Group of Automated Messages, or Database and/or in conjunction with a date range. If you do not specify a date range, Engage will use the last 30 days to filter the results.

			Note: When this element is used, Engage will flag exported events for future exports. Exports initiated without this element will not flag exported events. A single user should not use this function with more than one set of event types. Use multiple user accounts if setting up multiple scheduled jobs that request different sets of event types with this function (e.g. One job for OPENS and CLICKS; another job for OPTOUTS).
	EVENT_DATE_START	Optional	Specifies the beginning boundary of activity for information to receive. If not specified, the Event Date Start will default to the Send Date Start. If you do not specify Event Date Range or Send Date Range, the system will default Event Date Range to the last 30 days.
	EVENT_DATE_END	Optional	Specifies the ending boundary of activity for information to receive. If you do not specify Event Date Range, the Event Date End will default to the Send Date End plus the number of days Engage is tracking the Organization's mailings. If you do not specify Event Date Range or Send Date Range, the system will default the Event Date Range to the last 30 days.
	SEND_DATE_START	Optional	Specifies the beginning "Send" boundary for information to receive. If you do not specify Send Date Range, the Send Date Start will default to Event Date Start minus the number of days Engage is tracking the Organization's mailings.
	SEND_DATE_END	Optional	Specifies the ending "Send" boundary for information to receive. If you do not specify Send Date Range, the Send Date End will default to the Event Date End.
Elements	EXPORT_FORMAT	Optional	Defines the formatting of the source file. Supported values are: 0 – CSV file 1 – Pipe-separated file 2 – Tab-separated file If you do not specify, Engage will use the default format (CSV).
	FILE_ENCODING	Optional	Defines the encoding of the exported file. Supported values are: • utf-8 • iso-8859-1 If not specified, Engage uses the Organization's default encoding.
	EXPORT_FILE_NAME	Optional	Specifies the output file name when submitting API request. This feature is used to avoid getting multiple files with the same name when submitting jobs at or near same time. If specified, the value will be used to replace 'Raw Recipient Data Export' in the file name. The date and time will still be appended to the file name.
	EMAIL	Optional	If specified, the provided email address receives notification when the job is complete.
	MOVE_TO_FTP	Optional	Use the MOVE_TO_FTP parameter to retrieve the output file programmatically. If specified, Engage moves the files to the download directory of the user's FTP space. If you omit the MOVE_TO_FTP parameter, Engage will place exported files in

			the Export Files area of Engage.
	PRIVATE	Optional	Parameter to retrieve private mailings. If the API does not receive a Private or Shared designation, Engage will return both private and shared mailings.
	SHARED	Optional	Parameter to retrieve shared mailings.
	SENT_MAILINGS	Optional	Mailing Type parameter to retrieve sent mailings.
	SENDING	Optional	Mailing Type parameter to retrieve mailings in the process of sending.
	OPTIN_CONFIRMATION	Optional	Mailing Type parameter to retrieve Opt–In Autoresponder mailings.
	PROFILE_CONFIRMATION	Optional	Mailing Type parameter to retrieve Edit Profile Autoresponder mailings.
	AUTOMATED	Optional	Mailing Type parameter to retrieve Custom Autoresponder mailings.
	CAMPAIGN_ACTIVE	Optional	Mailing Type parameter to retrieve active Groups of Automated Messages.
	CAMPAIGN_COMPLETED	Optional	Mailing Type parameter to retrieve completed Groups of Automated Messages.
	CAMPAIGN_CANCELLED	Optional	Mailing Type parameter to retrieve canceled Groups of Automated Messages.
	CAMPAIGN_SCRAPE_TEMPLATE	Optional	Mailing Type parameter to retrieve mailings that use content retrieval.
	INCLUDE_TEST_MAILINGS	Optional	Specify to include Test Mailings. If you do not provide this element, Engage will not return any test mailings.
	ALL_EVENT_TYPES	Optional	Specify to receive all events regardless of Event Type. If ALL_EVENT_TYPES is used, **do not** specify any of the individual metrics parameters.
	SENT	Optional	Specify to receive Sent events. **Note**: Suppressed contacts are not included. If a mailing is in the process of sending (for example, Throttle Mailings) and you invoke the ALL_NON_EXPORTED feature, Engage will not include Sent events until it sends to all contacts.
Elements	SUPPRESSED	Optional	Specify to receive Suppressed events. **Note**: Suppressed contacts are not included. If a mailing is in the process of sending (for example, Throttle Mailings) and you invoke the ALL_NON_EXPORTED feature, Engage will not include Suppressed events until it sends to all contacts.
	OPENS	Optional	Specify to receive Open events.
	CLICKS	Optional	Specify to receive Clickthrough events.
	OPTINS	Optional	Specify to receive Opt In events.
	OPTOUTS	Optional	Specify to receive Opt Out events.
	FORWARDS	Optional	Specify to receive Forwarded events.
	ATTACHMENTS	Optional	Specify to receive Attachment events.

	CONVERSIONS	Optional	Specify to receive Conversion events.		
	CLICKSTREAMS	Optional	Specify to receive Clickstream events.		
	HARD_BOUNCES	Optional	Specify to receive Hard Bounce events.		
	SOFT_BOUNCES	Optional	Specify to receive Soft Bounce events.		
	REPLY_ABUSE	Optional	Use to receive Reply – Abuse events.		
	REPLY_COA	Optional	Use to receive Reply – Change of Address events.		
	REPLY_OTHER	Optional	Use to receive Reply – Other events.		
	MAIL_BLOCKS	Optional	Use to receive Mail Block events.		
	MAILING_RESTRICTIONS	Optional	Use to receive Mail Restricted events.		
	SMS_ERROR	Optional	Use to receive SMS Error Events		
	SMS_REJECT	Optional	Use to receive SMS Reject Events		
	SMS_OPTOUT	Optional	Use to receive SMS Opt Out Events		
	INCLUDE_SEEDS	Optional	Specify to include Seed contacts. **Note**: If you delete a Seed contact from the Seed List, Engage will not export their associated events.		
	INCLUDE_FORWARDS	Optional	Use to include Forwarded contacts.		
	INCLUDE_INBOX_MONITORIN G	Optional	Use to include Inbox Monitoring contacts.		
	CODED_TYPE_FIELDS	Optional	Use to return numeric values rather than strings in the following fields: Contact Type, Event Type, Body Type, and Suppression Reason.		
	EXCLUDE_DELETED	Optional	Use to exclude events for contacts who were deleted/purged from their database. **Note:** including this element can **greatly decrease the time** to generate the metrics file and is recommended whenever metrics for deleted contacts are not required.		
	FORWARDS_ONLY	Optional	Used to retrieve only events performed by forwarded recipients.		
	RETURN_MAILING_NAME	Optional	Specify to include Mailing Name in the generated file. If not specified, 'false' will be assumed.		
	RETURN_SUBJECT	Optional	Specify to include Mailing Subject in the generated file. If not specified, 'false' will be assumed.		
	RETURN_CRM_CAMPAIGN_ID	Optional	Specify to include the CRM Campaign Id (when populated) in the generated file. If not specified 'false' will be assumed.		
Optional Child Elements	COLUMNS	Optional	XML node used to request list columns to export for each contact.		
	Child Element	COLUMN			
		Child Element	NAME	Specifies the field name.	
Example	<Envelope>				

	<Body> <RawRecipientDataExport> <EVENT_DATE_START>12/01/2011 00:00:00</EVENT_DATE_START> <EVENT_DATE_END>12/02/2011 23:59:00</EVENT_DATE_END> <MOVE_TO_FTP/> <EXPORT_FORMAT>0</EXPORT_FORMAT> <EMAIL>admin@yourorg.com</EMAIL> <ALL_EVENT_TYPES/> <INCLUDE_INBOX_MONITORING/> <COLUMNS> <COLUMN> <NAME>CustomerID</NAME> </COLUMN> <COLUMN> <NAME>Address</NAME> </COLUMN> </COLUMNS> </RawRecipientDataExport> </Body> </Envelope>

Response	<RESULT>		
Elements	SUCCESS	**True** if successful.	
	MAILING	XML nodes defining the user-created column name and value.	
	Child Element	JOB_ID	Specifies resulting Background Job ID for the export. You can use this value with the GET_JOB_STATUS and/or DELETE_JOB APIs
		FILE_PATH	Returns the file name of the export file.

Example

```
<Envelope>
  <Body>
    <RESULT>
      <SUCCESS>TRUE</SUCCESS>
      <MAILING>
        <JOB_ID>72649</JOB_ID>
        <FILE_PATH>15167_20041213100410_track.zip</FILE_PATH>
      </MAILING>
    </RESULT>
  </Body>
</Envelope>
```

A single file exports with the following columns populated for each event:

Column	Description
Recipient ID	The ID of the contact associated with the event.
Recipient Type	The type of contact to whom Engage sent the mailing. Valid values are: • Regular – 0 • Forward – 1 • Seed – 3 • Inbox Monitoring – 4
Mailing ID	The ID of the Sent Mailing associated with the event.
Report ID	Depending on the type of mailing, you can assign Report IDs in a number of ways. For event-driven Autoresponders, a single Report ID is associated with every mailing for a day. For a recurring Automated Message, a single Report ID is associated with each occurrence of the mailing. For a standard mailing, there is a one-to-one relationship between a Report ID and Mailing ID.
Campaign ID	The ID of the Group of Automated Messages associated with the event.
Email	The contact's email address.

Event Type	The type of contact event. Valid values are: • Open – 0 • Click Through – 1 • Clickstream – 2 • Conversion – 3 • Attachment – 4 • Media – 5 • Forward – 6 • Opt In – 7 • Opt Out – 8 • Reply Abuse – 10 • Reply Change Address – 11 • Reply Mail Block – 12 • Reply Mail Restriction – 13 • Reply Other – 14 • Suppressed – 15 • Sent – 16 • Soft Bounce – 98 • Hard Bounce – 99 • SMS Error – 18 • SMS Reject – 19 • SMS Opt Out – 20
Event Timestamp	The date and time of the event.

You can (optionally) populate the following columns based on the Event Type:

Column	Description	Event Types
Body Type	The body type the contact received. Valid values are: • HTML – 0 • AOL – 1 • TEXT – 2 • WEB – 3 (Click-to-View)	Clickthrough, Open
Content ID	The user-specified identifier of the attachment.	Attachments
Click Name	The user-specified name of the link or Clickstream.	Clickthrough, Clickstream
URL	The hyperlink of a Clickthrough or Clickstream.	Clickthrough, Clickstream
Conversion Action	The user-specified action of a conversion.	Conversion
Conversion Detail	The user-specified description of a conversion.	Conversion
Conversion Amount	The dollar amount of a conversion.	Conversion
Suppression Reason	The reason a contact was suppressed. Valid values are: • Invalid System Email Domain – 1 • Invalid System Email Local – 2 • Invalid Organization Email Domain – 3 • Organization Suppression List – 4 • Global Suppression – 5 • Invalid Organization Email Local – 6 • Frequency Control – 7 • Database Level Suppression – 8 • Query Level Suppression – 9 • Mailing Level Suppression – 10	Suppressed

You can optionally populate the following columns by requesting them in the request (they will be returned between the Suppression Reason and the requested database columns):

Column	Description
Mailing Name	The Mailing Name of the mailing associated with the event. This column is only present if RETURN_MAILING_NAME is specified in the request.
Mailing Subject	The Subject Line of the mailing associated with the event. This column is only present if RETURN_SUBJECT is specified in the request.
CRM Campaign Id	The CRM Campaign Id linked to the mailing associated with the event. This column is only present if RETURN_CRM_CAMPAIGN_ID is specified in the request. ***Note***: This column will only be populated for mailings associated with a Campaign in a CRM system.

Export Web Tracking Events

WebTrackingDataExport

This interface allows exporting unique Web Tracking events and creates a .zip file containing a single flat file with all events. You can request all (or a subset) of the Event Types. The API provides the ability to specify the following:

- One or more Landing Pages Site
- One or more Custom Domains
- A specific Database
- An Event Date Range

If a Site, Domain, or Database is not specified, all events for the Event Date Range will be returned for the Organization of the calling user.

A file is generated based on the request and placed in the Stored Files or FTP account associated with the Engage user calling the API.

The following table shows Input XML definitions.

Operation	**<WebTrackingDataExport>**		
Elements	EVENT_DATE_START	Optional	Specifies the beginning boundary of activity for events to receive. ***Note***: Events within the range of the Organizations retention setting will be returned although when the date range includes days older than the retention setting all available web tracking data within the specified range will be provided.
	EVENT_DATE_END	Optional	Specifies the ending boundary of activity for events to receive. Required if ALL_NON_EXPORTED element is not provided.
	DOMAINS	Optional	Used to specify one or more Custom Domains for which to export events.
	Child Element	DOMAIN_ID	Indicates a Custom Domain for which to export events.
	SITES	Optional	Used to specify one or more Landing Pages Sites for which to export events.
	Child Element	SITE_ID	Indicates a Site for which to export events.
	DATABASE_ID	Optional	The Database ID is optional based on recipient database field columns being requested.
	EXPORT_FORMAT	Optional	Defines the formatting of the source file. Supported values are: 0 – CSV file 1 – Pipe-separated file 2 – Tab-separated file If you do not specify, Engage will use the default format (CSV).

	EXPORT_FILE_NAME	Optional	If specified, the value will be used to replace 'Web Tracking Export' in the file name. The date and time will still be appended to the file name.
	FILE_ENCODING	Optional	Defines the encoding of the exported file. Supported values are: • utf-8 • iso-8859-1 If not specified, Engage uses the Organization's default encoding.
	EMAIL	Optional	If specified, the provided email address receives notification when the job is complete.
	MOVE_TO_FTP	Optional	Use the MOVE_TO_FTP parameter to retrieve the output file programmatically. If specified, Engage moves the files to the download directory of the user's FTP space. If you omit the MOVE_TO_FTP parameter, Engage will place exported files in the Export Files area of Engage.
	ALL_EVENT_TYPES	Optional	Specify to receive all events regardless of Event Type. If ALL_EVENT_TYPES is used, **do not** specify any of the individual event types. At least one event type must be specified when ALL_EVENT_TYPES is not included.
	INCLUDE_SITE_VISIT_EVENTS	Optional	Specify to receive Site Visit events.
	INCLUDE_PAGE_VIEW_EVENTS	Optional	Specify to receive Page View events.
	INCLUDE_CLICK_EVENTS	Optional	Specify to receive Click events.
	INCLUDE_FORM_SUBMIT_EVENTS	Optional	Specify to receive Form Submit events.
	INCLUDE_DOWNLOAD_EVENTS	Optional	Specify to receive Download events.
	INCLUDE_MEDIA_EVENTS	Optional	Specify to receive Played Media events.
	INCLUDE_SHARE_TO_SOCIAL_EVENTS	Optional	Specify to receive Share to Social events.
	INCLUDE_CUSTOM_EVENTS	Optional	Specify to receive Custom events.
	COLUMNS	Optional	XML node used to request database columns to export for each contact. The generated file will contain columns for each of the column names specified in the request.
	Child Element	COLUMN	
		Child Element	NAME — Specifies the field name.
Example	<Envelope> <Body> <WebTrackingDataExport> <EVENT_DATE_START>12/01/2011 00:00:00</EVENT_DATE_START> <EVENT_DATE_END>12/02/2011 23:59:00</EVENT_DATE_END> <DATABASE_ID>123456</DATABASE_ID> <DOMAINS> <DOMAIN_ID>122412</DOMAIN_ID>		

	<DOMAIN_ID>234523</DOMAIN_ID> </DOMAINS> <SITES> <SITE_ID>2389423</SITE_ID> </SITES> <MOVE_TO_FTP/> <EXPORT_FORMAT>0</EXPORT_FORMAT> <EMAIL>admin@yourorg.com</EMAIL> <ALL_EVENT_TYPES/> <COLUMNS> <COLUMN> <NAME>CustomerID</NAME> </COLUMN> <COLUMN> <NAME>Address</NAME> </COLUMN> </COLUMNS> </WebTrackingDataExport > </Body> </Envelope>

Response	<RESULT>	
Elements	SUCCESS	**True** if successful.
	JOB_ID	Specifies resulting Background Job ID for the export. You can use this value with the GET_JOB_STATUS and/or DELETE_JOB APIs
	FILE_PATH	Returns the file name of the export file. Example: "Web Tracking Export Jan 11 2011 08-30-55 AM 1254.zip" If EXPORT_FILE_NAME is specified in the request, the value will replace 'Web Tracking Export'. ***Note***: The string following the time is a unique identifier used to ensure files are not overwritten if multiple files are generated at the same time..
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <JOB_ID>72649</JOB_ID> <FILE_PATH> Web Tracking Export Jan 11 2011 08-30-55 AM 1254.zip </FILE_PATH> </RESULT> </Body> </Envelope>	

A single file is generated with the following columns populated for each event:

Column	Description
DATABASE_ID	The ID of the database the contact belongs to.
RECIPIENT_ID	The ID of the contact associated with the event.
VISITOR_ID	The Web Tracking Visitor ID associated with the event.
SESSION_ID	The Session ID associated with the visitor's session.
SESSION_START_TS	The Start Time associated with the visitor's session.
SESSION_LEAD_SOURCE	The Lead Source ID associated with the visitor's session.
REFERRER_TYPE	Indicates the identified referrer type that led the visitor to your site. Valid values are: • Mailing • Search Engine • Website • Landing Page Site • Social Network • Direct)
REFERRER_NAME	Indicates the recognized referrer name, based on type that led the visitor to your site. Valid

Column	Description
	values are: • [Search Engine Name] • Website • Landing Page Site • Social Network • Direct
REFERRER_URL	Indicates the full referring URL that led the visitor to your site. Valid values are: • Full path URL
REFERRER_MAILING_ID	If the referrer was a mailing, this value indicates the Mailing ID that led the visitor to your site. Valid values are: • Mailing ID
REFERRER_KEYWORDS	If the referrer was a type of Search Engine, this value provides the visitor's Search Term that led the visitor to your site. Valid values are: • The visitor's Search Term
EVENT_ID	Indicates the unique Event ID associated with the web tracking event raised by the visitor: • EVENT ID
EVENT_TS	The Event Time Stamp associated with the web tracking event raised by the visitor.
EVENT_TYPE_CODE	The numeric identifier associated with the event.
EVENT_TYPE_NAME	The type of Web Tracking event. Valid values are: • Site Visit • Page View • Click • Form Submit • Download • Played Media • Share to Social • Custom Event Type
EVENT_HYPERLINK_ID	The Hyperlink passed with the associated web tracked events for Click and Share to Social.
PAGE_ID	The Page ID associated with a page in your landing pages site where the event originated.
EVENT_NAME	The ~~Page~~ Event Name associated with a page in your site where.
PAGE_PARENT_ID	The form container page ID associated with the form contained in your landing page site.
PAGE_PARENT_NAME	The form page container type name associated with your form. • std – Opt-in Standard Form • Confirmation – Conformation Page

Column	Description
	• Opt-out – Opt-out form • Form
EVENT_URL	The full URL to the page where the event occurred. • If the event type code is Page View the URL points to a page URL. • If the event type code is a Click or Download the URL displays the friendly name or Resource Name
SITE_DOMAIN_ID	The Domain ID associated with the landing page site or custom domain used for your external domain.
SITE_NAME	The name of your landing page site.
SITE_TYPE_CODE	Indicates whether a Site is Silverpop Hosted or External. Valid values are: • 0 – Silverpop Hosted • 1 – External
SITE_NAME	The name of the landing page site
SITE_URL	The full path to the Landing Pages site.

The following columns will be optionally populated based on the Event Type:

Column	Description	Event Types
SESSION_LEAD_SOURCE	The name of the lead source associated with the referrer (If specified)	Fully Supported: • Site Visit • Page View • Click • Form Submit Silverpop Hosted (Exclusive) • Share to Social Silverpop Hosted (Exclusive) • Custom Event Types • Download • Played Media
REFERRER_URL	The full URL passed by the referring website, social network or search engine.	All
REFERRER_MAILING_ID	The mailing ID associated with the referrer, if the visit was referred from a mailing.	All
REFERRER_KEYWORDS	The search term associated with the referred visit from a search engine.	All

Column	Description	Event Types
EVENT_HYPERLINK_ID	The Hyperlink passed with the associated web tracked events for Click and Share to Social.	All
PAGE_PARENT_ID	The form container page ID associated with the form contained in your landing page site.	All
PAGE_PARENT_NAME	The form page container type name associated with a form. • std – Standard Form • Confirmation – Conformation Page • Opt out – Opt-out form Form	IF REFERRER_TYPE = "Mailing"
EVENT_URL	The full URL to the page where the event occurred.	If REFERRER_TYPE = "Search Engine" | "Website" | "Landing Page Site" | "Social Network"
SITE_DOMAIN_ID	The Domain ID associated with the landing page site or custom domain used for your external domain.	IF REFERRER_TYPE = Search
SITE_NAME	The name of your landing page site.	IF present
SITE_TYPE_CODE	Indicates whether a Site is Silverpop Hosted or External. Valid values are: • 0 – Silverpop Hosted • 1 – External	All

Get a List of Report IDs for a Mailing

GetReportIdByDate

This interface extracts the Report IDs for a particular mailing sent between specified dates. For standard mailings, this will be a single ID, but for Autoresponders and Automated Messages, this will be one ID per day.

Operation	<GetReportIdByDate>	
Elements	MAILING_ID	ID of the mailing for which to export metrics. The account used to log on to the current session must own this mailing.
	DATE_START	Specifies the beginning boundary of information to receive. If time is included, it must be in 24-hour format.
	DATE_END	Specifies the ending boundary of information to receive. If time is included, it must be in 24-hour format.
Example	<Envelope> <Body>	

Operation	**<GetReportIdByDate>**		
	<GetReportIdByDate> <MAILING_ID>1</MAILING_ID> <DATE_START>07/25/2011 00:00:00</DATE_START> <DATE_END>09/30/2011 23:59:59</DATE_END> </GetReportIdByDate> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS	**True** if successful.	
	Mailing	XML nodes defining the instances of the sent mailing.	
	Child Element	ReportId	Specifies the Report ID for the mailing on the date specified in "SentTS"
		SentTS	The date and time for the mailing, returned in the following format: "mm/dd/yy hh:mm AMPM"
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <Mailing> <ReportId>91570</ReportId> <SentTS>1/17/05 7:00 PM</SentTS> </Mailing> </RESULT> </Body> </Envelope>		

Get a List of Sent Mailings for an Org

GetSentMailingsForOrg

This interface extracts a listing of mailings sent for an organization for a specified date range.

Note: The Engage user account calling the API must have Organization Administration rights.

Operation	**<GetSentMailingsForOrg>**	
Elements	DATE_START	Required Starting Date in the format "mm/dd/yyyy hh:mm:ss"
	DATE_END	Required Ending Date in the format "mm/dd/yyyy hh:mm:ss"
	PRIVATE	Optional parameter to retrieve private mailings. If the API does not receive a Private or Shared parameter, Engage will return both private and shared mailings.
	SHARED	Optional parameter to retrieve shared mailings.
	SCHEDULED	Optional Mailing Type parameter to retrieve scheduled mailings. If the API does not receive a mailing type, Engage will return mailings of all types. Engage uses the various mailing type parameters to limit the list to only the specified types.
	SENT	Optional Mailing Type parameter to retrieve sent mailings.
	SENDING	Optional Mailing Type parameter to retrieve mailings in the process of sending. The SCHEDULED parameter will also include mailings in SENDING status.
Elements	OPTIN_CONFIRMATION	Optional Mailing Type parameter to retrieve Opt-In Autoresponder mailings.
	PROFILE_CONFIRMATION	Optional Mailing Type parameter to retrieve Edit Profile Autoresponder mailings.
	AUTOMATED	Optional Mailing Type parameter to retrieve Custom Autoresponder mailings.
	CAMPAIGN_ACTIVE	Optional Mailing Type parameter to retrieve active Groups of Automated Messages.
	CAMPAIGN_COMPLETED	Optional Mailing Type parameter to retrieve completed Groups of Automated Messages.
	CAMPAIGN_CANCELLED	Optional Mailing Type parameter to retrieve canceled Groups of Automated Messages.
	CAMPAIGN_SCRAPE_TEMPLATE	Optional Mailing Type parameter to retrieve mailings that use content retrieval.
	INCLUDE_TAGS	Optional parameter to return all Tags associated with the Sent mailing.
	EXCLUDE_ZERO_SENT	Optional parameter to exclude mailings with no contacts.
	MAILING_COUNT_ONLY	Optional parameter to return only the count of sent mailings for a specific date range.
	EXCLUDE_TEST_MAILINGS	Optional parameter requesting to exclude Test Mailings. If you do not provide this element, Engage will include all Test Mailings.

Example	```<Envelope><Body><GetSentMailingsForOrg><PRIVATE/><SENT/><DATE_START>07/25/2011 00:00:00</DATE_START><DATE_END>09/30/2011 23:59:59</DATE_END></GetSentMailingsForOrg></Body></Envelope>```			
Response	**<RESULT>**			
Elements	SUCCESS	**True** if successful.		
	Mailing	XML nodes defining the instances of each mailing.		
	Child Element	MailingId	Returns the mailing ID.	
		ReportId	Returns the report ID for a particular instance of an automated mailing.	
		ScheduledTS	Returns the scheduled date and time in the format, "mm/dd/yy hh:mm AMPM"	
		MailingName	Returns the mailing name.	
		ListName	Returns the database name.	
		ListId	Returns the ID of the Database, Query ID, or Contact List that the mailing was sent to.	
		ParentListId	Returns the ID of the parent database when the mailing was sent to a Query or Contact List. If the mailing was not sent to a Query or Contact List, this element will not be included.	
		UserName	Returns the mailing owner's user name.	
		SentTS	Returns the date and time Engage sent the mailing.	
		NumSent	Returns the number of contacts to whom Engage sent a mailing.	
		SentMailingsCount	Returns the number of sent mailings.	
		Subject	Returns the mailing subject.	
		Visibility	Returns visibility information about the mailing. Values are **Private** or **Shared**.	
		Tags	If INCLUDE_TAGS is specified in the request, returns any tags associated with the Sent mailing.	
			Tag	XML nodes containing the Tags associated with the Sent mailing.

Example 1 with standard usage	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <Mailing> <MailingId>5758</MailingId> <ReportId>109118</ReportId> <ScheduledTS>2011-06-14 10:54:06.0</ScheduledTS> <MailingName><![CDATA[This is the mailing name]]></MailingName> <ListName><![CDATA[This is the database name]]></ListName> <ListId>4615</ListId> <UserName> John Doe </UserName> <SentTS/> <NumSent>0</NumSent> <Subject><![CDATA[Summer Sale]]></Subject> <Visibility>Private</Visibility> </Mailing> <Mailing> <MailingId>5758</MailingId> <ReportId>109118</ReportId> <ScheduledTS>2011-06-14 10:54:06.0</ScheduledTS> <MailingName><![CDATA[My Mailing 2]]></MailingName> <ListName><![CDATA[This is the database name 2]]></ListName> <ListId>4616</ListId> <UserName>John Doe</UserName> <SentTS/> <NumSent>0</NumSent> <Subject><![CDATA[Summer Sale]]></Subject> <Visibility>Private</Visibility> </Mailing> </RESULT> </Body> </Envelope>

Example 2 when MAILING_COUNT_ONLY parameter is used	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <SentMailingsCount>5758</SentMailingsCount> </RESULT> </Body> </Envelope>

Get a List of Sent Mailings for a User

GetSentMailingsForUser

This interface extracts a listing of mailings sent for the logged on user for a specified date range.

If the calling user is an Org Admin, all users' mailings will be returned. If using the OPTIONALUSER parameter, an Org Admin can specify a particular username to retrieve only mailings sent by that user.

Operation	**<GetSentMailingsForUser>**	
Elements	DATE_START	Required Starting Date in the format "mm/dd/yyyy hh:mm:ss"
	DATE_END	Required Ending Date in the format "mm/dd/yyyy hh:mm:ss"
	OPTIONALUSER	If the calling user is an Org Admin, all users' mailings will be returned. If using the OPTIONALUSER parameter, an Org Admin can specify a particular username to retrieve only mailings sent by that user.
	PRIVATE	Optional parameter to retrieve private mailings. If the API does not receive a private or shared designation, Engage will return both private and shared mailings.
	SHARED	Optional parameter to retrieve shared mailings.
	SCHEDULED	Optional Mailing Type parameter to retrieve scheduled mailings. Engage returns mailings of all types when the API does not receive a mailing type designation.
	SENT	Optional Mailing Type parameter to retrieve sent mailings.
	SENDING	Optional Mailing Type parameter to retrieve mailings in the process of sending.
	OPTIN_CONFIRMATION	Optional Mailing Type parameter to retrieve Opt-in Autoresponder mailings.
	PROFILE_CONFIRMATION	Optional Mailing Type parameter to retrieve Edit Profile Autoresponder mailings.
	AUTOMATED	Optional Mailing Type parameter to retrieve Custom Autoresponder mailings.
	CAMPAIGN_ACTIVE	Optional Mailing Type parameter to retrieve active Groups of Automated Messages.
	CAMPAIGN_COMPLETED	Optional Mailing Type parameter to retrieve completed Groups of Automated

		Messages.
	CAMPAIGN_CANCELLED	Optional Mailing Type parameter to retrieve canceled Groups of Automated Messages.
	CAMPAIGN_SCRAPE_TEMPLATE	Optional Mailing Type parameter to retrieve Campaign Scrape Template mailings.
	INCLUDE_TAGS	Optional parameter to return all Tags associated with the Sent mailing.
	EXCLUDE_ZERO_SENT	Optional parameter to exclude mailings with no contacts.
	MAILING_COUNT_ONLY	Optional parameter to return only the count of sent mailings for a specific date range.
	EXCLUDE_TEST_MAILINGS	Optional parameter requesting to exclude Test Mailings. If you do not provide this element, Engage will include all Test Mailings.
Example	<Envelope> <Body> <GetSentMailingsForUser> <PRIVATE/> <SENT/> <DATE_START>07/25/2011 00:00:00</DATE_START> <DATE_END>09/30/2011 23:59:59</DATE_END> <OPTIONALUSER>name@domain.com</OPTIONALUSER> </GetSentMailingsForUser> </Body> </Envelope>	

Response	**<RESULT>**			
Elements	SUCCESS	**True** if successful.		
	Mailing	XML nodes defining the instances of the sent mailing.		
	Child Element	MailingId	Returns the Mailing ID.	
		ReportId	Returns the report ID for a particular instance of an automated mailing.	
		ScheduledTS	Returns the scheduled date and time in the following format: "mm/dd/yy hh:mm AMPM"	
		MailingName	Returns the mailing name.	
		ListName	Returns the database name.	
		ListId	Returns the ID of the database or query that the mailing was sent to.	
		ParentListId	Returns the ID of the parent database when the mailing was sent to a query. If the mailing was not sent to a query, this element will be empty.	
		UserName	Returns the user name for the mailing owner.	
		SentTS	Returns the date/time when Engage sent the mailing.	
		NumSent	Returns the number of contacts to whom Engage sent the mailing.	
		SentMailingsCount	Returns the number of sent mailings.	
		Subject	Returns the mailing subject.	
		Visibility	Returns visibility information about the mailing. Values are **Private** or **Shared**.	
		Tags	If INCLUDE_TAGS is specified in the request, returns any tags associated with the Sent mailing.	
			Tag	XML nodes containing the Tags associated with the Sent mailing.

Example 1 with standard usage

```
<Envelope>
  <Body>
    <RESULT>
      <SUCCESS>TRUE</SUCCESS>
      <Mailing>
        <MailingId>5758</MailingId>
        <ReportId>109118</ReportId>
        <ScheduledTS>201-06-14 10:54:06.0</ScheduledTS>
        <MailingName><![CDATA[This is the mailing name]]></MailingName>
        <ListName><![CDATA[This is the database name]]></ListName>
        <ListId>4615</ListId>
        <UserName> John Doe </UserName>
        <SentTS/>
        <NumSent>0</NumSent>
        <Subject><![CDATA[Summer Sale]]></Subject>
        <Visibility>Private</Visibility>
      </Mailing>
      <Mailing>
        <MailingId>5758</MailingId>
        <ReportId>109118</ReportId>
        <ScheduledTS>2011-06-14 10:54:06.0</ScheduledTS>
        <MailingName><![CDATA[My Mailing 2]]></MailingName>
```

	<ListName><![CDATA[This is the database name 2]]></ListName> <ListId>4616</ListId> <UserName>John Doe</UserName> <SentTS/> <NumSent>0</NumSent> <Subject><![CDATA[Summer Sale]]></Subject> <Visibility>Private</Visibility> </Mailing> </RESULT> </Body> </Envelope>
Example 2 when MAILING_COUNT_ONLY parameter is used	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <Mailing> <SentMailingsCount>5758</SentMailingsCount> </Mailing> </RESULT> </Body> </Envelope>

Get a List of Sent Mailings for a Database

GetSentMailingsForList

This interface extracts a listing of mailings sent for a particular database and specified date range. It allows specification of a Database or Query ID as well as a flag to include "children." **Note:** The Engage user account calling the API must have Organization Administration rights.

Operation	**<GetSentMailingsForList>**	
Elements	LIST_ID	Required ID of the Database or Query for which to retrieve mailings.
	INCLUDE_CHILDREN	Optional parameter to retrieve mailings for queries based on the specified Database ID.
	DATE_START	Required Starting Date in the format "mm/dd/yyyy hh:mm:ss"
	DATE_END	Required Ending Date in the format "mm/dd/yyyy hh:mm:ss"
	PRIVATE	Optional parameter to retrieve private mailings. If the API does not receive private or shared designation, Engage will return both private and shared mailings.
	SHARED	Optional parameter to retrieve shared mailings.
	SCHEDULED	Optional mailing type parameter to retrieve scheduled mailings. If you do not specify a mailing type in the API, Engage will return mailings of all types; the various mailing type parameters are used to limit the list to only the specified types.
	SENT	Optional Mailing Type parameter to retrieve sent mailings.
	SENDING	Optional Mailing Type parameter to retrieve mailings in the process of sending.
	OPTIN_CONFIRMATION	Optional Mailing Type parameter to retrieve Opt-in Autoresponder mailings.
	PROFILE_CONFIRMATION	Optional Mailing Type parameter to retrieve Edit Profile Autoresponder mailings.
	AUTOMATED	Optional Mailing Type parameter to retrieve Custom Autoresponder mailings.
	CAMPAIGN_ACTIVE	Optional Mailing Type parameter to retrieve active Groups of Automated Messages.
	CAMPAIGN_COMPLETED	Optional Mailing Type parameter to retrieve completed Groups of Automated Messages.
	CAMPAIGN_CANCELLED	Optional Mailing Type parameter to retrieve canceled Groups of Automated Messages.
	CAMPAIGN_SCRAPE_TEMPLATE	Optional Mailing Type parameter to retrieve Campaign Scrape Template mailings.
	INCLUDE_TAGS	Optional parameter to return all Tags associated with the Sent mailing.
	EXCLUDE_ZERO_SENT	Optional parameter to exclude mailings with no contacts.
	MAILING_COUNT_ONLY	Optional parameter to return only the count of sent mailings for a specific date range.
	EXCLUDE_TEST_MAILINGS	Optional parameter requesting to exclude Test Mailings. If you do not provide this

		element, Engage will include all Test Mailings.		
Example	<Envelope> <Body> <GetSentMailingsForList> <DATE_START>07/25/2011 00:00:00</DATE_START> <DATE_END>09/30/201123:59:59</DATE_END> <LIST_ID>27546</ LIST_ID > <INCLUDE_CHILDREN/> </GetSentMailingsForList> </Body> </Envelope>			
Response	**<RESULT>**			
Elements	SUCCESS		**True** if successful.	
	Mailing		XML nodes defining the instances of the sent mailing.	
	Child Element	MailingId	Returns the Mailing ID.	
		ReportId	Returns the report ID for a particular instance of an automated mailing.	
		ScheduledTS	Returns the scheduled date and time in the following format: "mm/dd/yy hh:mm AMPM"	
		MailingName	Returns the mailing name.	
		ListName	Returns the database name.	
		ListId	Returns the database ID.	
		UserName	Returns the user name for the mailing owner.	
		SentTS	Returns the date/time when Engage sent the mailing.	
		NumSent	Returns the number of contacts to whom Engage sent the mailing.	
		SentMailingsCount	Returns the number of sent mailings.	
		Subject	Returns the mailing subject.	
		Visibility	Returns visibility information about the mailing. Values are **Private** or **Shared**.	
		QueryId	Returns the query ID if the mailing was sent to a query.	
		QueryName	Returns the query name if the mailing was sent to a query.	
		Tags	Returns any tags associated with the Sent mailing if you specified INCLUDE_TAGS in the request,.	
			Tag	XML nodes containing the Tags associated with the Sent mailing.

Example 1 with standard usage	<pre><Envelope>
	  <Body>
	    <RESULT>
	      <SUCCESS>TRUE</SUCCESS>
	      <Mailing>
	        <MailingId>5758</MailingId>
	        <ReportId>109118</ReportId>
	        <ScheduledTS>2011-06-14 10:54:06.0</ScheduledTS>
	        <MailingName><![CDATA[This is the mailing name]]></MailingName>
	        <ListId>27546</ListId>
	        <QueryId>4185</QueryId>
	        <ListName><![CDATA[This is the database name]]></ListName>
	        <QueryName><![CDATA[This is the query name]]></QueryName>
	        <UserName> John Doe </UserName>
	        <SentTS/>
	        <NumSent>0</NumSent>
	        <Subject><![CDATA[Summer Sale]]></Subject>
	        <Visibility>Private</Visibility>
	      </Mailing>
	      <Mailing>
	        <MailingId>5758</MailingId>
	        <ReportId>109118</ReportId>
	        <ScheduledTS>2011-06-14 10:54:06.0</ScheduledTS>
	        <MailingName><![CDATA[My Mailing 2]]></MailingName>
	        <ListName><![CDATA[This is the database name 2]]></ListName>
	        <ListId>4616</ListId>
	        <UserName>John Doe</UserName>
	        <SentTS/>
	        <NumSent>0</NumSent>
	        <Subject><![CDATA[Summer Sale]]></Subject>
	        <Visibility>Private</Visibility>
	        </Mailing>
	    </RESULT>
	  </Body>
	</Envelope></pre>

Example 2 when MAILING_COUNT_ONLY parameter is used	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <Mailing> <SentMailingsCount>5758</SentMailingsCount> </Mailing> </RESULT> </Body> </Envelope>

Get Aggregate Tracking Metrics for a Mailing

GetAggregateTrackingForMailing

This interface extracts metrics for a specified mailing.

Operation	**<GetAggregateTrackingForMailing>**		
Elements	MAILING_ID	Required parameter specifying the Mailing ID.	
	REPORT_ID	Required parameter specifying the Report ID.	
	TOP_DOMAIN	Optional parameter requesting top domain statistics for each mailing.	
	INBOX_MONITORING	Optional parameter requesting Inbox Monitoring statistics for each mailing.	
	PER_CLICK	Optional parameter requesting the total clicks for each link in the mailing by Body Type.	
Example	<Envelope> <Body> <GetAggregateTrackingForMailing> <MAILING_ID>123</MAILING_ID> <REPORT_ID>12332</REPORT_ID> </GetAggregateTrackingForMailing> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS	**True** if successful.	
	Mailing	XML nodes defining the instances of the sent mailing.	
	Child Element	MailingId	Returns the Mailing ID.
		ReportId	Returns the Report ID.
		MailingName	Returns the mailing name.
		SentDateTime	Returns the date and time Engage sent the mailing.
		NumSent	Returns the number of contacts to whom Engage sent the mailing.
		NumSeeds	Returns the number of contacts seeded into the mailing.
		NumSuppressed	Returns the number of contacts suppressed from the mailing.
		NumInboxMonitored	Returns the number of Inbox Monitoring addresses seeded into this mailing.
		NumBounceHard	Returns the number of hard bounces.
		NumBounceSoft	Returns the number of soft bounces.
		NumUniqueOpen	Returns the number of unique opens.
		NumGrossOpen	Returns the number of gross opens.
		NumUniqueClick	Returns the number of unique clicks.
		NumGrossClick	Returns the number of gross clicks.
		NumUniqueAttach	Returns the number of unique attachment downloads.
		NumGrossAttach	Returns the number of gross attachment downloads.
		NumUniqueClickstreams	Returns the number of unique clickstream clicks.
		NumGrossClickstreams	Returns the number of gross clickstream clicks.
		NumUniqueMedia	Returns the number of unique media plays.
		NumGrossMedia	Returns the number of gross media plays.
		NumGrossAbuse	Returns the number of abuse complaints.
		NumGrossChangeAddress	Returns the number of change of address replies.

		NumGrossMailBlock	Returns the number of mail block replies.
		NumGrossMailRestriction	Returns the number of mail restriction replies.
		NumGrossOther	Returns the number of other replies.
		NumConversions	Returns the number of conversions.
		NumConversionAmount	Returns the total conversion amount.
		NumBounceHardFwd	Returns the number of hard bounces that occurred for forwarded contacts.
		NumBounceSoftFwd	Returns the number of soft bounces that occurred for forwarded contacts.
		NumConversionAmountFwd	Returns the total conversion amount that occurred for forwarded contacts.
		NumClickFwd	Returns the number of clicks for forwarded contacts.
		NumUniqueForwardFwd	Returns the number of unique forwards by forwarded contacts.
		NumGrossForwardFwd	Returns the number of gross forwards by forwarded contacts.
		NumUniqueConversionsFwd	Returns the number of unique conversions by forwarded contacts.
		NumGrossConversionsFwd	Returns the number of gross conversions by forwarded contacts.
		NumUniqueClickstreamFwd	Returns the number of unique clickstream clicks by forwarded contacts.
		NumGrossClickstreamFwd	Returns the number of gross clickstream clicks by forwarded contacts.
		NumUniqueClickFwd	Returns the number of unique clicks by forwarded contacts.
		NumGrossClickFwd	Returns the number of gross clicks by forwarded contacts.
		NumUniqueAttachOpenFwd	Returns the number of unique attachment downloads by forwarded contacts.
		NumGrossAttachOpenFwd	Returns the number of gross attachment downloads by forwarded contacts.
		NumUniqueMediaFwd	Returns the number of unique media plays by forwarded contacts.
		NumGrossMediaFwd	Returns the number of gross media plays by forwarded contacts.
		NumUniqueOpenFwd	Returns the number of unique opens by forwarded contacts.
		NumGrossOpenFwd	Returns the number of gross opens by forwarded contacts.
		NumAbuseFwd	Returns the number of abuse complaints by forwarded contacts.
		NumChangeAddressFwd	Returns the number of change address replies by forwarded contacts.
		NumMailRestrictionFwd	Returns the number of mail restriction replies by forwarded contacts.
		NumMailBlockFwd	Returns the number of mail block replies by forwarded contacts.
		NumOtherFwd	Returns the number of other replies by forwarded contacts.
		NumSuppressedFwd	Returns the number of suppressed forwards.
		NumUnsubscribes	Returns the number of Opt Outs.
	TopDomain	XML nodes defining top domain reporting for the sent mailings. If you do not include TopDomain in the request, Engage will not populate this node.	

	Child Element	MailingId	Returns the Mailing ID.
		ReportId	Returns the Report ID.
		Domain	Returns the domain name.
		Sent	Returns the number of contacts sent for the domain.
		Bounce	Returns the number of bounces for the domain.
		Open	Returns the number of opens for contacts of the domain.
		Click	Returns the number of clicks for contacts of the domain.
		Unsubscribe	Returns the number of unsubscribe requests for contacts on that domain.

		Conversion	Returns the number of conversions for contacts of the domain.
		Conversion_Amount	Returns the total conversion amount for contacts of the domain.
		Reply_abuse	Returns the number of abuse complaint replies for contacts on that domain.
		Reply_mail_block	Returns the number of mail block replies for contacts of the domain.
		Reply_mail_restriction	Returns the number of mail restriction replies for contacts of the domain.
	InboxMonitored	XML nodes defining inbox monitoring reporting for the sent mailings. Child nodes will be encapsulated in a "InboxMonitoring" Node. This node will not be populated if the option is not included in the request.	
	Child Element	MailingId	Returns the Mailing ID.
		ReportId	Returns the Report ID.
		Domain	Returns the domain name.
		Sent	Returns the number of inbox monitoring contacts sent for the domain.
		Inbox	Returns the number of inbox monitoring contacts emails delivered to the inbox.
		Bulk	Returns the number of inbox monitoring contacts emails delivered to the bulk folder.
		NotReceived	Returns the number of inbox monitoring contacts emails not delivered.
	Clicks	XML nodes listing tracking information for tracked hyperlinks. Child nodes encapsulated in a "Click" node. Not populated if the option is not included in the request.	
	Child Element	MailingId	Returns the Mailing ID.
		ReportId	Returns the Report ID.
		LinkName	Returns the link name.
		LinkURL	Returns the link URL.
		TotalHTML	Returns the number of clicks on the link in the HTML body.
		TotalAOL	Returns the number of clicks on the link in the AOL body.
		TotalWEB	Returns the number of clicks on the link in the WEB body.
		TotalTEXT	Returns the number of clicks on the link in the TEXT body.
Example	See the example results for GetAggregateTrackingForUser.		

Get Aggregate Tracking Metrics for an Org

GetAggregateTrackingForOrg

This interface extracts a listing of mailings sent for an organization for a specified date range and provides metrics for those mailings.

Operation	**<GetAggregateTrackingForOrg>**	
Elements	DATE_START	Required Starting Date in the format "mm/dd/yyyy hh:mm:ss"
	DATE_END	Required Ending Date in the format "mm/dd/yyyy hh:mm:ss"
	PRIVATE	Optional parameter to retrieve private mailings. If neither private nor shared is passed to the API, Engage will return both private and shared mailings.
	SHARED	Optional parameter to retrieve shared mailings.
Elements	SCHEDULED	Optional Mailing Type parameter to retrieve scheduled mailings. If no mailing type is passed to the API, Engage will return mailings of all types.
	SENT	Optional Mailing Type parameter to retrieve sent mailings.
	SENDING	Optional Mailing Type parameter to retrieve mailings in the process of sending.
	OPTIN_CONFIRMATION	Optional Mailing Type parameter to retrieve Opt-in Autoresponder mailings.
	PROFILE_CONFIRMATION	Optional Mailing Type parameter to retrieve Edit Profile Autoresponder mailings.
	AUTOMATED	Optional Mailing Type parameter to retrieve Custom Autoresponder mailings.
	CAMPAIGN_ACTIVE	Optional Mailing Type parameter to retrieve active Groups of Automated Messages
	CAMPAIGN_COMPLETED	Optional Mailing Type parameter to retrieve completed Groups of Automated Messages.
	CAMPAIGN_CANCELLED	Optional Mailing Type parameter to retrieve canceled Groups of Automated Messages.
	TOP_DOMAIN	Optional parameter requesting top domain statistics for each mailing.
	INBOX_MONITORING	Optional parameter requesting Inbox Monitoring statistics for each mailing.
	PER_CLICK	Optional parameter requesting the total clicks for each link in the mailing by Body Type.
	EXCLUDE_TEST_MAILINGS	Optional parameter requesting to exclude Test Mailings. If you do not provide this element, Engage will include all Test Mailings.
Example	<Envelope> <Body> <GetAggregateTrackingForOrg> <PRIVATE/> <SENT/> <DATE_START>07/25/2011 00:00:00</DATE_START> <DATE_END>09/30/2011 23:59:59</DATE_END> </GetAggregateTrackingForOrg> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**True** if successful.
	Mailing	XML nodes defining the instances of the sent mailing.

	Child Element	MailingId	Returns the Mailing ID.
		ReportId	Returns the Report ID.
		MailingName	Returns the mailing name.
		SentDateTime	Returns the date and time Engage sent the mailing.
		NumSent	Returns the number of contacts to whom Engage sent the mailing.
		NumSeeds	Returns the number of contacts seeded into the mailing.
		NumSuppressed	Returns the number of contacts suppressed from the mailing.
		NumInboxMonitored	Returns the number of Inbox Monitoring addresses seeded into this mailing.
		NumBounceHard	Returns the number of hard bounces.
		NumBounceSoft	Returns the number of soft bounces.
		NumUniqueOpen	Returns the number of unique opens.
		NumGrossOpen	Returns the number of gross opens.
		NumUniqueClick	Returns the number of unique clicks.
		NumGrossClick	Returns the number of gross clicks.
		NumUniqueAttach	Returns the number of unique attachment downloads.
		NumGrossAttach	Returns the number of gross attachment downloads.
		NumUniqueClickstreams	Returns the number of unique clickstream clicks.
		NumGrossClickstreams	Returns the number of gross clickstream clicks.
		NumUniqueMedia	Returns the number of unique media plays.
		NumGrossMedia	Returns the number of gross media plays.
		NumGrossAbuse	Returns the number of abuse complaints.
		NumGrossChangeAddress	Returns the number of change of address replies.
		NumGrossMailBlock	Returns the number of mail block replies.
		NumGrossMailRestriction	Returns the number of mail restriction replies.
		NumGrossOther	Returns the number of other replies.
		NumConversions	Returns the number of conversions.
		NumConversionAmount	Returns the total conversion amount.
		NumBounceHardFwd	Returns the number of hard bounces that occurred for forwarded contacts.
		NumBounceSoftFwd	Returns the number of soft bounces that occurred for forwarded contacts.
		NumConversionAmountFwd	Returns the total conversion amount that occurred for forwarded contacts.
		NumClickFwd	Returns the number of clicks for forwarded contacts.
		NumUniqueForwardFwd	Returns the number of unique forwards by forwarded contacts.
		NumGrossForwardFwd	Returns the number of gross forwards by forwarded contacts.
		NumUniqueConversionsFwd	Returns the number of unique conversions by forwarded contacts.
		NumGrossConversionsFwd	Returns the number of gross conversions by forwarded contacts.
		NumUniqueClickstreamFwd	Returns the number of unique clickstream clicks by forwarded contacts.
		NumGrossClickstreamFwd	Returns the number of gross clickstream clicks by forwarded contacts.
		NumUniqueClickFwd	Returns the number of unique clicks by forwarded contacts.
		NumGrossClickFwd	Returns the number of gross clicks by forwarded contacts.
		NumUniqueAttachOpenFwd	Returns the number of unique attachment downloads by forwarded contacts.
		NumGrossAttachOpenFwd	Returns the number of gross attachment downloads by

			forwarded contacts.
		NumUniqueMediaFwd	Returns the number of unique media plays by forwarded contacts.
		NumGrossMediaFwd	Returns the number of gross media plays by forwarded contacts.
		NumUniqueOpenFwd	Returns the number of unique opens by forwarded contacts.
		NumGrossOpenFwd	Returns the number of gross opens by forwarded contacts.
		NumAbuseFwd	Returns the number of abuse complaints by forwarded contacts.
		NumChangeAddressFwd	Returns the number of change address replies by forwarded contacts.
		NumMailRestrictionFwd	Returns the number of mail restriction replies by forwarded contacts.
		NumMailBlockFwd	Returns the number of mail block replies by forwarded contacts.
		NumOtherFwd	Returns the number of other replies by forwarded contacts.
		NumSuppressedFwd	Returns the number of suppressed forwards.
		NumUnsubscribes	Returns the number of Opt Outs.
	TopDomain		XML nodes defining top domain reporting for the sent mailings. Node not populated if the option is not included in the request.
	Child Element	MailingId	Returns the Mailing ID.
		ReportId	Returns the Report ID.
		Domain	Returns the domain name.
		Sent	Returns the number of contacts sent for the domain.
		Bounce	Returns the number of bounces for the domain.
		Open	Returns the number of opens for contacts.of the domain.
		Click	Returns the number of clicks for contacts.of the domain.
		Unsubscribe	Returns the number of unsubscribe requests for contacts.of the domain.
		Conversion	Returns the number of conversions for contacts.of the domain.
		Conversion_Amount	Returns the total conversion amount for contacts.of the domain.
		Reply_abuse	Returns the number of abuse complain replies for contacts.of the domain.
		Reply_mail_block	Returns the number of mail block replies for contacts.of the domain.
		Reply_mail_restriction	Returns the number of mail restriction replies for contacts.of the domain.
	InboxMonitored		XML nodes defining inbox monitoring reporting for the sent mailings. Child nodes will be encapsulated in a "InboxMonitoring" Node. This node will not be populated if the option is not included in the request.

	Child Element	MailingId	Returns the Mailing ID.
		ReportId	Returns the Report ID
		Domain	Returns the domain name.
		Sent	Returns the number of inbox monitoring contacts.sent for the domain.
		Inbox	Returns the number of inbox monitoring contact.emails delivered to the inbox.
		Bulk	Returns the number of inbox monitoring contact emails delivered to the bulk folder.
		NotReceived	Returns the number of inbox monitoring contact emails not delivered.
	Clicks		XML nodes listing tracking information for tracked hyperlinks. Child nodes encapsulated in a "Click" Node. Not populated if the option is not included in the request.
	Child Element	MailingId	Returns the Mailing ID.
		ReportId	Returns the Report ID.
		LinkName	Returns the link Name.
		LinkURL	Returns the link URL.
		TotalHTML	Returns the number of clicks on the link in the HTML body.
		TotalAOL	Returns the number of clicks on the link in the AOL body.
		TotalWEB	Returns the number of clicks on the link in the WEB body.
		TotalTEXT	Returns the number of clicks on the link in the TEXT body.
Example	See the example results for GetAggregateTrackingForUser.		

Get Aggregate Tracking Metrics for a User

GetAggregateTrackingForUser

This interface extracts a listing of mailings sent by a user for a specified date range and provides metrics for those mailings.

Operation	<GetAggregateTrackingForUser>	
Elements	DATE_START	Required Starting Date in the format "mm/dd/yyyy hh:mm:ss"
	DATE_END	Required Ending Date in the format "mm/dd/yyyy hh:mm:ss"
	OPTIONALUSER	Optional User's Email Address parameter to retrieve mailings for a user other than the current user who created the session.
	PRIVATE	Optional parameter to retrieve private mailings. If neither private nor shared is passed to the API, Engage will return both private and shared mailings.
	SHARED	Optional parameter to retrieve shared mailings.
	SCHEDULED	Optional mailing type parameter to retrieve scheduled mailings. If no mailing type is passed to the API, Engage will return mailings of all types.
	SENT	Optional Mailing Type parameter to retrieve sent mailings.
	SENDING	Optional Mailing Type parameter to retrieve mailings in the process of sending.
	OPTIN_CONFIRMATION	Optional Mailing Type parameter to retrieve Optin Autoresponder mailings.
	PROFILE_CONFIRMATION	Optional Mailing Type parameter to retrieve Edit Profile Autoresponder mailings.
	AUTOMATED	Optional Mailing Type parameter to retrieve Custom Autoresponder mailings.
	CAMPAIGN_ACTIVE	Optional Mailing Type parameter to retrieve active Groups of Automated Messages.
	CAMPAIGN_COMPLETED	Optional Mailing Type parameter to retrieve completed Groups of Automated Messages.
	CAMPAIGN_CANCELLED	Optional Mailing Type parameter to retrieve canceled Groups of Automated Messages.
	TOP_DOMAIN	Optional parameter requesting top domain statistics for each mailing.
	INBOX_MONITORING	Optional parameter requesting Inbox Monitoring statistics for each mailing.
	PER_CLICK	Optional parameter requesting the total clicks for each link in the mailing by Body Type.
	EXCLUDE_TEST_MAILINGS	Optional parameter requesting to exclude Test Mailings. If you do not provide this element, Engage will include all Test Mailings.
Example	<Envelope> <Body> <GetAggregateTrackingForUser> <DATE_START>07/25/2011 12:12:11</DATE_START>	

	<DATE_END>09/30/2011 14:14:11</DATE_END> <PRIVATE/> <SENT/> <OPTIONALUSER>username@domain.com</OPTIONALUSER> </GetAggregateTrackingForUser> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS		**True** if successful.
	Mailing		XML nodes defining the instances of the sent mailing.
	Child Element	MailingId	Returns the Mailing ID.
		ReportId	Returns the Report ID.
		NumSent	Returns the number of contacts to whom Engage sent the mailing.
		MailingName	Returns the mailing name.
		SentDateTime	Returns the date and time the mailing was sent.
		NumSeeds	Returns the number of contacts seeded into the mailing.
		NumSuppressed	Returns the number of contacts suppressed from the mailing.
		NumInboxMonitored	Returns the number of Inbox Monitoring addresses seeded into this mailing.
		NumBounceHard	Returns the number of hard bounces.
		NumBounceSoft	Returns the number of soft bounces.
		NumUniqueOpen	Returns the number of unique opens.
		NumGrossOpen	Returns the number of gross opens.
		NumUniqueClick	Returns the number of unique clicks.
		NumGrossClick	Returns the number of gross clicks.
		NumUniqueAttach	Returns the number of unique attachment downloads.
		NumGrossAttach	Returns the number of gross attachment downloads.
		NumUniqueClickstreams	Returns the number of unique clickstream clicks.
		NumGrossClickstreams	Returns the number of gross clickstream clicks.
		NumUniqueMedia	Returns the number of unique media plays.
		NumGrossMedia	Returns the number of gross media plays.
		NumGrossAbuse	Returns the number of abuse complaints.
		NumGrossChangeAddress	Returns the number of change of address replies.

		NumGrossMailBlock	Returns the number of mail block replies.
		NumGrossMailRestriction	Returns the number of mail restriction replies.
		NumGrossOther	Returns the number of other replies.
		NumConversions	Returns the number of conversions.
		NumConversionAmount	Returns the total conversion amount.
		NumBounceHardFwd	Returns the number of hard bounces that occurred for forwarded contacts.
		NumBounceSoftFwd	Returns the number of soft bounces that occurred for forwarded contacts.
		NumConversionAmountFwd	Returns the total conversion amount that occurred for forwarded contacts.
		NumClickFwd	Returns the number of clicks for forwarded contacts.
		NumUniqueForwardFwd	Returns the number of unique forwards by forwarded contacts.
		NumGrossForwardFwd	Returns the number of gross forwards by forwarded contacts.
		NumUniqueConversionsFwd	Returns the number of unique conversions by forwarded contacts.
		NumGrossConversionsFwd	Returns the number of gross conversions by forwarded contacts.
		NumUniqueClickstreamFwd	Returns the number of unique clickstream clicks by forwarded contacts.
		NumGrossClickstreamFwd	Returns the number of gross clickstream clicks by forwarded contacts.
		NumUniqueClickFwd	Returns the number of unique clicks by forwarded contacts.
		NumGrossClickFwd	Returns the number of gross clicks by forwarded contacts.
		NumUniqueAttachOpenFwd	Returns the number of unique attachment downloads by forwarded contacts.
		NumGrossAttachOpenFwd	Returns the number of gross attachment downloads by forwarded contacts.

		NumUniqueMediaFwd	Returns the number of unique media plays by forwarded contacts.
		NumGrossMediaFwd	Returns the number of gross media plays by forwarded contacts.
		NumUniqueOpenFwd	Returns the number of unique opens by forwarded contacts.
		NumGrossOpenFwd	Returns the number of gross opens by forwarded contacts.
		NumAbuseFwd	Returns the number of abuse complaints by forwarded contacts.
		NumChangeAddressFwd	Returns the number of change address replies by forwarded contacts.
		NumMailRestrictionFwd	Returns the number of mail restriction replies by forwarded contacts.
		NumMailBlockFwd	Returns the number of mail block replies by forwarded contacts.
		NumOtherFwd	Returns the number of other replies by forwarded contacts.
		NumSuppressedFwd	Returns the number of suppressed forwards.
		NumUnsubscribes	Returns the number of Opt Outs.
	TopDomain	XML nodes defining top domain reporting for the sent mailings. Not populated if the option is not included in the request.	
	Child Element	MailingId	Returns the Mailing ID.
		ReportId	Returns the Report ID.
		Domain	Returns the domain name.
		Sent	Returns the number of contacts sent for the domain.
		Bounce	Returns the number of bounces for the domain.
		Open	Returns the number of opens for contacts of the domain.
		Click	Returns the number of clicks for contacts of the domain.
		Unsubscribe	Returns the number of unsubscribe requests for contacts of the domain.
		Conversion	Returns the number of conversions for contacts of the domain.
		Conversion_Amount	Returns the total conversion amount for contacts of the domain.
		Reply_abuse	Returns the number of abuse complain replies for contacts of the domain.
		Reply_mail_block	Returns the number of mail block replies for contacts of the domain.
		Reply_mail_restriction	Returns the number of mail restriction replies for contacts of the domain.
	InboxMonitored	XML nodes defining inbox monitoring reporting for the sent mailings. Child nodes are encapsulated in a "InboxMonitoring" node. Not populated if you do not include this option in the request.	
	Child Element	MailingId	Returns the Mailing ID.

		ReportId	Returns the Report ID.
		Domain	Returns the domain name.
		Sent	Returns the number of contacts sent for the domain.
		Inbox	Returns the number of Inbox Monitoring contact emails delivered to the inbox.
		Bulk	Returns the number of Inbox Monitoring contact emails delivered to the bulk folder.
		NotReceived	Returns the number of Inbox Monitoring contact emails not delivered.
	Clicks	XML nodes listing tracking information for tracked hyperlinks. Child nodes are encapsulated in a "Click" node. Not populated if you do not include this option in the request.	
	Child Element	MailingId	Returns the Mailing ID.
		ReportId	Returns the Report ID.
		LinkName	Returns the link name.
		LinkURL	Returns the link URL.
		TotalHTML	Returns the number of clicks on the link in the HTML body.
		TotalAOL	Returns the number of clicks on the link in the AOL body.
		TotalWEB	Returns the number of clicks on the link in the WEB body.
		TotalTEXT	Returns the number of clicks on the link in the TEXT body.

Example

```
<Envelope>
	<Body>
	<RESULT>
		<SUCCESS>TRUE</SUCCESS>
		<Mailing>
			<MailingId>9919</MailingId>
			<ReportId>108480</ReportId>
			<MailingName><![CDATA[API Demo Mailing)]]></MailingName>
			<SentDateTime>2011-09-21 08:49:55.0</SentDateTime>
			<NumSent>55520</NumSent>
			<NumSeeds>0</NumSeeds>
			<NumSuppressed>0</NumSuppressed>
			<NumInboxMonitored>190</NumInboxMonitored>
			<NumBounceHard>3902</NumBounceHard>
			<NumBounceSoft>5016</NumBounceSoft>
			<NumUniqueOpen>2815</NumUniqueOpen>
			<NumGrossOpen>0</NumGrossOpen>
			<NumUniqueClick>2815</NumUniqueClick>
			<NumGrossClick>0</NumGrossClick>
			<NumUniqueAttach>0</NumUniqueAttach>
			<NumGrossAttach>0</NumGrossAttach>
			<NumUniqueClickstreams>0</NumUniqueClickstreams>
			<NumGrossClickstreams>0</NumGrossClickstreams>
			<NumUniqueMedia>0</NumUniqueMedia>
			<NumGrossMedia>0</NumGrossMedia>
			<NumGrossAbuse>0</NumGrossAbuse>
			<NumGrossChangeAddress>1113</NumGrossChangeAddress>
			<NumGrossMailBlock>1115</NumGrossMailBlock>
			<NumGrossMailRestriction>1115</NumGrossMailRestriction>
			<NumGrossOther>3903</NumGrossOther>
			<NumConversions>0</NumConversions>
			<NumConversionAmount>0</NumConversionAmount>
			<NumBounceHardFwd>0</NumBounceHardFwd>
			<NumBounceSoftFwd>0</NumBounceSoftFwd>
			<NumConversionAmountFwd>0</NumConversionAmountFwd>
			<NumAttachOpenFwd>0</NumAttachOpenFwd>
			<NumClickFwd>0</NumClickFwd>
			<NumUniqueForwardFwd>0</NumUniqueForwardFwd>
			<NumGrossForwardFwd>0</NumGrossForwardFwd>
			<NumUniqueConversionsFwd>0</NumUniqueConversionsFwd>
			<NumGrossConversionsFwd>0</NumGrossConversionsFwd>
			<NumUniqueClickstreamFwd>0</NumUniqueClickstreamFwd>
			<NumGrossClickstreamFwd>0</NumGrossClickstreamFwd>
			<NumUniqueClickFwd>0</NumUniqueClickFwd>
			<NumGrossClickFwd>0</NumGrossClickFwd>
			<NumUniqueAttachOpenFwd>0</NumUniqueAttachOpenFwd>
			<NumGrossAttachOpenFwd>0</NumGrossAttachOpenFwd>
			<NumUniqueMediaFwd>0</NumUniqueMediaFwd>
	<NumGrossMediaFwd>0</NumGrossMediaFwd>
			<NumUniqueOpenFwd>0</NumUniqueOpenFwd>
			<NumGrossOpenFwd>0</NumGrossOpenFwd>
```

	<NumAbuseFwd>0</NumAbuseFwd> <NumChangeAddressFwd>0</NumChangeAddressFwd> <NumMailRestrictionFwd>0</NumMailRestrictionFwd> <NumMailBlockFwd>0</NumMailBlockFwd> <NumOtherFwd>0</NumOtherFwd> <NumSuppressedFwd>0</NumSuppressedFwd> <NumUnsubscribes>5</NumUnsubscribes> </Mailing> <TopDomains> <TopDomain> <MailingId>9919</MailingId> <ReportId>108480</ReportId> <Domain>cs.com</Domain> <Sent>540</Sent> <Bounce>83</Bounce> <Open>122</Open> <Click>122</Click> <Unsubscribe>0</Unsubscribe> <Conversion>0</Conversion> <Conversion_amount>0</Conversion_amount> <Reply_abuse>0</Reply_abuse> <Reply_mail_block>14</Reply_mail_block> <Reply_mail_restriction>12</Reply_mail_restriction> </TopDomain> <TopDomain> <MailingId>9919</MailingId> <ReportId>108480</ReportId> <Domain>aol.com</Domain> <Sent>12866</Sent> <Bounce>2060</Bounce> <Open>1278</Open> <Click>1289</Click> <Unsubscribe>0</Unsubscribe> <Conversion>0</Conversion> <Conversion_amount>0</Conversion_amount> <Reply_abuse>0</Reply_abuse> <Reply_mail_block>259</Reply_mail_block> <Reply_mail_restriction>260</Reply_mail_restriction> </TopDomain> </TopDomains> <InboxMonitored> <InboxMonitoring> <MailingId>9919</MailingId> <ReportId>0</ReportId> <Domain>aol.com</Domain> <Sent>10</Sent> <Inbox>0</Inbox> <Bulk>0</Bulk> <NotReceived>10</NotReceived> </InboxMonitoring> <InboxMonitoring>

	<MailingId>9919</MailingId> <ReportId>0</ReportId> <Domain>bellsouth.net</Domain> <Sent>10</Sent> <Inbox>0</Inbox> <Bulk>0</Bulk> <NotReceived>10</NotReceived> </InboxMonitoring> </InboxMonitored> <Clicks/> </RESULT> </Body> </Envelope>

Get Status of a Data Job

GetJobStatus

After initiating a data job, you can monitor the status of the job using this operation. This step is optional in the job process.

If a data job completes with errors, you can view detailed results in the Results file (.res) and Error file (.err) which you can find in the **Download** folder of the FTP account.

Operation	**<GetJobStatus>**	
Elements	JOB_ID	Identifies the Engage Background Job created and scheduled as a result of another API call.
Example	<Envelope> <Body> <GetJobStatus> <JOB_ID>40865</JOB_ID> </GetJobStatus> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**TRUE** if successful.
	JOB_ID	Identifies the Engage Background Job created and scheduled as a result of the API call. Will match the JOB_ID passed in the request
	JOB_STATUS	Identifies the current job status at the time of the API call. Valid values are: • WAITING • RUNNING • CANCELED • ERROR • COMPLETE
	JOB_DESCRIPTION	Description of the job being processed.
	PARAMETERS	Parameter list that contains <Parameter> elements with name/value pairs. (See each response example in the tables below for more detail.)
Example	<Envelope> <Body>	

```xml
<RESULT>
    <SUCCESS>TRUE</SUCCESS>
    <JOB_ID>789052</JOB_ID>
    <JOB_STATUS>COMPLETE</JOB_STATUS>
    <JOB_DESCRIPTION> Creating new contact source, Master Database</JOB_DESCRIPTION>
    <PARAMETERS>
        <PARAMETER>
            <NAME>NOT_ALLOWED</NAME>
            <VALUE>0</VALUE>
        </PARAMETER>
        <PARAMETER>
            <NAME>LIST_ID</NAME>
            <VALUE>116347</VALUE>
        </PARAMETER>
        <PARAMETER>
            <NAME>RESULTS_FILE_NAME</NAME>
            <VALUE>1241474.res</VALUE>
        </PARAMETER>
        <PARAMETER>
            <NAME>SQL_ADDED</NAME>
            <VALUE>65535</VALUE>
        </PARAMETER>
        <PARAMETER>
            <NAME>DUPLICATES</NAME>
            <VALUE>0</VALUE>
        </PARAMETER>
        <PARAMETER>
            <NAME>TOTAL_ROWS</NAME>
            <VALUE>65535</VALUE>
        </PARAMETER>
        <PARAMETER>
            <NAME>ERROR_FILE_NAME</NAME>
            <VALUE>1241474.err</VALUE>
        </PARAMETER>
        <PARAMETER>
            <NAME>LIST_NAME</NAME>
            <VALUE>Big List</VALUE>
        </PARAMETER>
        <PARAMETER>
            <NAME>BAD_ADDRESSES</NAME>
            <VALUE>0</VALUE>
        </PARAMETER>
        <PARAMETER>
            <NAME>SQL_UPDATED</NAME>
            <VALUE>0</VALUE>
        </PARAMETER>
```

```
                <PARAMETER>
                    <NAME>BAD_RECORDS</NAME>
                    <VALUE>0</VALUE>
                </PARAMETER>
                <PARAMETER>
                    <NAME>TOTAL_VALID</NAME>
                    <VALUE>65535</VALUE>
                </PARAMETER>
            </PARAMETERS>
        </RESULT>
    </Body>
</Envelope>
```

There are various job types that you can execute in Engage and monitor using the GetJobStatus operation. The response data will differ between types of jobs and the current status. For example, if the current status of a Database Import job is WAITING, the response element will contain different elements and values from the same call with a status of ERROR.

The following tables describe the different Parameters returned from **GetJobStatus** depending on the current state of the job and the API called to create the job.

Parameters when a Database Import Job is Waiting/Running

PARAMETERS	List of parameters beginning with <PARAMETER> element and containing name/value pairs with details on the status of the export database job.	
Name/Value Pairs	NOT_ALLOWED	Number of rows not allowed.
	LIST_ID	ID of database. Blank until Engage creates the database. Once created, this will have a value throughout processing and when the job is complete.
	SQL_ADDED	Number of rows added to the database. During processing, this number will increment in steps of 10,000 until the job is complete
	DUPLICATES	Number of duplicate rows detected.
	TOTAL_ROWS	Number of rows in database. During processing, this value will change as the initial source file parses and it reads all rows in the file.
	LIST_NAME	Name of database.
	BAD_ADDRESSES	Number of email addresses in incorrect format.
	SQL_UPDATED	Number of rows updated.
	TOTAL_VALID	Number of rows that were valid in the database. Increments as the source file processes.
	BAD_RECORDS	Number of rows that were not valid.

Parameters when Database Import Job is Complete

PARAMETERS	List of parameters beginning with <PARAMETER> element and containing name/value pairs with details on the status of the job.	
Name/Value Pairs	NOT_ALLOWED	Number of rows not allowed.
	LIST_ID	ID of database. This will be blank until Engage creates the database. Once the database is created, this will have a value throughout processing and when the job is complete.
	SQL_ADDED	Number of rows added to the database. During processing, this number will increment in steps of 10,000 until the job is complete.
	DUPLICATES	Number of duplicate rows detected.
	TOTAL_ROWS	Number of rows in database. During processing, this value will change as the initial source file is parsed until it reads all rows in the file.
	LIST_NAME	Name of database.
	BAD_ADDRESSES	Number of email addresses in incorrect format.
	SQL_UPDATED	Number of rows updated.
	BAD_RECORDS	Number of rows that were valid in the source file. This will increment as the source file is processed.
	TOTAL_VALID	Number of rows that were not valid.

Parameters when Database Export Job is Waiting/Running

PARAMETERS	List of parameters beginning with <PARAMETER> element and containing name/value pairs with details on the status of the job.	
Name/Value Pairs	NUM_PROCESSED	Number of rows processed. This will increment during processing of the job.
	LIST_ID	ID for exported database or query.
	LIST_NAME	Name of exported database or query.

Parameters when Database Export Job is Complete

PARAMETERS	List of parameters beginning with <PARAMETER> element and containing name/value pairs with details on the status of the job.	
Name/Value Pairs	NUM_PROCESSED	Number of rows processed. Value is the number of rows.
	LIST_ID	ID for the database processed.
	LIST_NAME	The name of the database being processed.

Delete a Data Job

DeleteJob

After verifying the successful completion of a data job, the user can remove the job from Engage by using the DeleteJobStatus operation.

Operation	**<DeleteJob>**	
Elements	JOB_ID	Identifies the Background Job to remove from Engage.
Example	<Envelope> <Body> <DeleteJob> <JOB_ID>1427</JOB_ID> </DeleteJob> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**True** if successful.
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> </RESULT> </Body> </Envelope>	

Get Folder Path for a Folder or Object

GetFolderPathThis interface returns the Folder Path for a specified Folder Id or a specified Object Id (e.g. Database, Query, Contact List, or Mailing).

Operation	**<GetFolderPath>**		
Elements	OBJECT_TYPE	The type of objects contained in the folder. Valid values are: • Data • Mailing	
	FOLDER_ID	Optional	The Id of folder whose folder path is being requested. This is a Parent Folder Id returned by another API call. **Note**: If TYPE is "Data", the FOLDER_ID will be an integer. If TYPE is "Mailing", the FOLDER_ID will be a GUID.
	OBJECT_ID	Optional	The Id of Mailing or Data object whose folder path is being requested. This is

			a List Id or Mailing Id returned by another API call.
Example 1 Folder	<Envelope> <Body> <GetFolderPath> <OBJECT_TYPE>Mailing</OBJECT_TYPE> <FOLDER_ID>12c734c-108b610e402-f528764d624db129b32c21fbcaocb8d6</FOLDER_ID> </GetFolderPath> </Body> </Envelope>		
Example 1 Object	<Envelope> <Body> <GetFolderPath> <OBJECT_TYPE>Data</OBJECT_TYPE> <OBJECT_ID>87456</OBJECT_ID> </GetFolderPath> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS	**TRUE** if successful.	
	FOLDER_PATH	The folder path of the specified Folder or Object.	
	OBJECT_SUB_TYPE	Optional	The sub-type of object matching the provided Object Id. This element is only populated when OBJECT_ID is passed in the request. If the OBJECT_TYPE is "Data", valid values are: • Database • Contact List • Query • Test List • Seed List • Suppression List • Relational Table If the OBJECT_TYPE is "Mailing", valid values are: • Template • Sent • Autoresponder • Inactive Autoresponder • Deleted
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <FOLDER_PATH>Shared/Promotions/January</FOLDER_PATH> <OBJECT_SUB_TYPE>Template</OBJECT_SUB_TYPE> </RESULT> </Body> </Envelope>		

Mailing Management Interfaces – User

Create or Update a Mailing Template

SaveMailing

This interface supports programmatically saving a new or updating an existing mailing template that may be used against a Database, Contact List or Query. It is critical to understand that this API will replace the existing template if it already exists (the lookup is based on MailingName element). Also, advanced features such as Dynamic Content, Relational Table, and Stored File assets are not supported currently in this version of the API. You can however, pass full HTML, personalization tags, and any references to externally hosted image URLs within that HTML body giving you a full range of desired look and feel.

Note: All elements are case sensitive to the spec.

Operation	**<SaveMailing>**			
Elements	**Header**	Defines all Mailing Template settings		
	Child Element	MailingName	The name of the new mailing template.	
		MailingID	Optional	Specify the Template Id if you want to update an existing Template instead of creating a new one.
		Subject	Subject of the Mailing Template	
		ListID	The ID of the database, contact list, or query being used for the mailing template.	
		FromName	Specify Mailing "From Name" value	
		FromAddress	Specify Mailing "From Address" value	
		ReplyTo	Specify Mailing "Reply To Address" value	
		Visibility	Defining the location of the saved mailing. Valid values are: 0 – Private 1 – Shared	
		FolderPath	Optional	Specify the path of the folder to save the Mailing Template. Example: • FolderName/SubfolderName **Note**: Folder names **are case sensitive**. If this element is not specified, Mailing Template will be saved in top-level directory.

		Encoding	This is the message encoding. Valid values are: 0 = Western European (iso-8859-1) 1 = Chinese (Simplified) 2 = Chinese (Traditional) 3 = Japanese (EUC) 4 = Japanese (Shift-JIS) 5 = Korean 6 = Unicode (utf8) 7 = Japanese (JIS)	
		TrackingLevel	The tracking level for the messages. Possible values are: 1 = None 2 = Opens Only 3 = Aggregate 4 = Unique	
		ClickHereMessage	Optional	Specify this parameter to add the 'click to view in browser' link at the top of the email message.
		IsCrmTemplate	Optional	Specify if this template is being shared with the CRM integration Valid values are • true • false
		HasSpCrmBlock	Optional	Specify if CRM Block is being used Valid values are • true • false
		PersonalFromName	Optional	Specify "From Name" Personalization
		PersonalFromAddress	Optional	Specify "From Address" Personalization
		PersonalReplyTo	Optional	Specify "Reply To Address" Personalization
	MessageBodies	Defines Mailing Body		
	Child Element	HTMLBody	Fully encapsulated HTML content used in the mailing body wrapped in a CDATA tag. Complete HTML tags (html, head, body) must be used in the content for HTMLBody. **NOTE**: Currently HTML Body supports Tracked Hyperlink, Non-Tracked Hyperlink, One Click Opt Out, Custom Opt Out and Click Stream Links. In order to define these values in this section,	

			following attribute needs to be added in the anchor tag. Non-Tracked Hyper Link : **xt="SPNOTRACK"** Tracked Hyper Link : **xt="SPCLICK"** Click Stream : **xt=" SPCLICKSTREAM"** One Click Opt Out : **xt=" SPONECLICKOPTOUT"** Custom Opt Out: **xt=" SPCUSTOMOPTOUT"** *Example of Tracked Hyper link used:* Silverpop	
		AOLBody	Optional	Fully encapsulated AOL content used in the mailing body wrapped in a CDATA tag.
		TextBody	Optional	Fully encapsulated Text content used in the mailing body wrapped in a CDATA tag.
	ClickThroughs	All Click Throughs are enclosed within this element. This Element is required if HTML Body consist of any Hyperlinks.		
	Child Element	ClickThrough	All click through elements are enclosed within this element	
		Child Element	ClickThroughName	The name of the url specified in the HTML content
			ClickThroughURL	The target url specified in the link within the HTML body **NOTE**: Specify following for one click opt out and custom opt out • **#SPONECLICKOPTOUT** • **#SPCUSTOMOPTOUT**
			ClickThroughType	Click through types. Supported values include: 1 = Not Tracked 2 = Regular Click Thru 5 = Clickstream Link 14 = One-Click Opt Out 19 = Custom Opt Out
	ForwardToFriend	Forward To Friend Link is currently **not supported** via API however, the Element is element is required in order to call the API		
	Child Element	ForwardType	Valid value: 0 = Not Embedded	

Use Case #1	This example **creates** an email template that contains an HTML body, personalization, and an opt out link:

Example

```xml
<Envelope>
        <Body>
        <SaveMailing>
                <Header>
                        <MailingName><![CDATA[Dummy Mailing template]]></MailingName>
                        <FromName><![CDATA[Silverpop]]></FromName>
                        <FromAddress><![CDATA[info@silverpop.com]]></FromAddress>
                        <ReplyTo><![CDATA[info@silverpop.com]]></ReplyTo>
                        <Visibility>0</Visibility>
                        <FolderPath>Shared/SPOP/MarketingTemplate</FolderPath>
                        <Subject>Subject of a Mailing</Subject>
                        <TrackingLevel>4</TrackingLevel>
                        <Encoding>6</Encoding>
                        <ListID>1924132</ListID>
                </Header>
                <MessageBodies>
                        <HTMLBody><![CDATA[<html>  <head>
    <title>Mailing</title>
  </head>
  <body>
    <p>This is an Example of a Mailing body. Visit <a name="Link1" href="http://www.silverpop.com">Silverpop</a> for
more information</p>
    <p>If you would like to no longer recieve any emails from this campaign, click <a href="#SPONECLICKOPTOUT"
xt="SPONECLICKOPTOUT" name="OOLink">here</a>t</p>
  </body>
</html>]]></HTMLBody>
       <AOLBody><![CDATA[<html>  <head>
    <title>Mailing</title>
  </head>
  <body>
    <p>This is an Example of a Mailing body. Visit <a name="Link1" href="http://www.silverpop.com">Silverpop</a> for
more information</p>
    <p>If you would like to no longer recieve any emails from this campaign, click <a href="#SPONECLICKOPTOUT"
xt="SPONECLICKOPTOUT" name="OOLink">here</a>t</p>
  </body>
</html>]]></AOLBody>
                </MessageBodies>
                <ClickThroughs>
                        <ClickThrough>
                        <ClickThroughName><![CDATA[Link1]]></ClickThroughName>
                        <ClickThroughType>2</ClickThroughType>
                        <ClickThroughURL>http://www.silverpop.com</ClickThroughURL>
                        </ClickThrough>
                        <ClickThrough>
```

	<ClickThroughName><![CDATA[OOLink]]></ClickThroughName> <ClickThroughURL><![CDATA[#SPONECLICKOPTOUT]]></ClickThroughURL> <ClickThroughType>14</ClickThroughType> </ClickThrough> </ClickThroughs> <ForwardToFriend> <ForwardType>0</ForwardType> </ForwardToFriend> </SaveMailing> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**TRUE** if successful
	MailingID	The ID of the template that was created or updated.
Example	<ENVELOPE> <BODY> <RESULT> <SUCCESS>TRUE</SUCCESS> <MailingID>81284534</MailingID> </RESULT> </BODY> </ENVELOPE>	

Schedule a Mailing

ScheduleMailing

Sends a template-based mailing to a specific database or query.

Operation	**<ScheduleMailing>**		
Elements	TEMPLATE_ID	ID of template upon which to base the mailing.	
	LIST_ID	ID of database, query, or contact list to send the template-based mailing.	
	MAILING_NAME	Name to assign to the generated mailing.	
	SEND_HTML	Optional	If this element is present, value is **true**. Leave blank if you do not want to include the HTML body. ***Note:*** at least one body type element is required.
	SEND_AOL	Optional	If this element is present, value is **true**. Leave blank if you do not want to include the AOL body.
	SEND_TEXT	Optional	If this element is present, value is **true**. Leave blank if you do not want to include the text body.
	SUBJECT	Optional	If you do not include this element, Engage uses the subject saved with the template.
	FROM_NAME	Optional	If you do not include this element, Engage uses the From Name saved with the template.
	FROM_ADDRESS	Optional	If you do not include this element, Engage uses the From Address saved with the template.
	REPLY_TO	Optional	If you do not include this element, Engage uses the Reply-To Address saved with the template.
	VISIBILITY	Value indicating where to save the sent mailing. Values are: 0 – Private Folder 1 – Shared Folder	
	SCHEDULED	Optional	Value specifying the date and time when the mailing will be scheduled to send. The format must be: MM/DD/YYYY HH:MM:SS AMPM If this element is not included, the mailing will be sent immediately.
	INBOX_MONITOR	Optional	If **True,** Engage activates Inbox Monitoring for the mailing.

	SEND_TIME_OPTIMIZATION	Optional	If specified, it is possible to send the mailing over a day or week. Valid values are: • NONE – will not use Send Time Optimization • SEND_24HRS – will send the mailing over a 24 hour period based on each recipient's optimal send time. • SEND_WEEK– will send the mailing over a 7 day period based on each recipient's optimal send time.
	WA_MAILINGLEVEL_CODE	Optional	This element is used in conjunction with Site Analytics Append String. If you have an append string that includes %%WA_MAILINGLEVEL_CODE%%, use the element to specify a mailing specific parameter to be passed via your links to your Site Analytics package. ***Note:*** This element is equivalent to the Mailing Specific Parameter box on the Send Mailing screen in Engage when using Site Analytics.
	SUPPRESSION_LISTS	Optional	XML node containing one or more SUPPRESSION_LIST_ID child elements.
	PARENT_FOLDER_PATH	Optional	Folder where you want to store your sent mailings.
	CREATE_PARENT_FOLDER	Optional	Create the folder if path does not already exist.
	CUSTOM_OPT_OUT	Optional	If provided, the element should contain a Custom Opt Out name that has been previously added in the Org Admin Opt Out settings. If the element is not provided, the standard opt out should be applied. If the element is provided but a custom opt out link is not included in template, the element will be ignored.
Optional Child Elements	SUPPRESSION_LIST_ID	If any suppression lists are specified, they will be utilized as the suppression lists when the mailing is sent.	
	SUBSTITUTIONS	Optional	XML node containing child element SUBSTITUTION
Optional Child Elements	SUBSTITUTION	XML nodes defining the substitution name and value. This feature allows template-level substitutions separate from the normal contact-level substitutions performed by personalization. They are not contact specific. They are also not related to HTML_GENERAL_DISCLAIMER or the TEXT_GENERAL_DISCLAIMER. **Note:** If used, each node must have a NAME tag and a VALUE tag.	
	Optional Child Element	NAME	The substitution tag used in the mailing template. This substitution name is case-sensitive. For example, if you include %%Sub_Value_1%% in the mailing template, the value for this child element is Sub_Value_1.
		VALUE	The value you would like to substitute into the mailing for the tag with the name given above.

Example

```
<Envelope>
    <Body>
        <ScheduleMailing>
            <TEMPLATE_ID>1000</TEMPLATE_ID>
            <LIST_ID>100</LIST_ID>
            <MAILING_NAME>New Mailing Name</MAILING_NAME>
            <SEND_HTML/>
            <SEND_AOL/>
            <SEND_TEXT/>
            <SUBJECT>This is the new subject</SUBJECT>
            <FROM_NAME>Senders Name</FROM_NAME>
            <FROM_ADDRESS>sender@domain.com</FROM_ADDRESS>
            <REPLY_TO>sender@domain.com</REPLY_TO>
            <VISIBILITY>0</VISIBILITY>
         <PARENT_FOLDER_PATH>Sent Folder Name</PARENT_FOLDER_PATH>
         <CREATE_PARENT_FOLDER/>
            <SCHEDULED>10/13/2011 12:00:00 AM</SCHEDULED>
            <SUPPRESSION_LISTS>
                <SUPPRESSION_LIST_ID>37782</SUPPRESSION_LIST_ID>
                <SUPPRESSION_LIST_ID>37744</SUPPRESSION_LIST_ID>
            </SUPPRESSION_LISTS>
            <SUBSTITUTIONS>
                <SUBSTITUTION>
                    <NAME>Sub_Value_1</NAME>
                 <VALUE>The value I would like to put in my mailing.</VALUE>
                </SUBSTITUTION>
                <SUBSTITUTION>
                    <NAME>Sub_Value_2</NAME>
                 <VALUE>Another value I would like to put in my mailing.</VALUE>
                </SUBSTITUTION>
            </SUBSTITUTIONS>
        </ScheduleMailing>
    </Body>
</Envelope>
```

Response	**<RESULT>**	
Elements	SUCCESS	**True** if successful.
	MAILING_ID	If the request was successful, this element contains the ID for the sent mailing.
Example Success	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <MAILING_ID>9700</MAILING_ID> </RESULT> </Body> </Envelope>	

Preview a Mailing Before Sending

PreviewMailing

This interface returns a preview of a mailing template. If a contact email address is provided in the request, the preview will include personalization for the specified contact.

Operation	**<PreviewMailing>**		
Elements	MailingId	Numeric identifier for the mailing template to be exported.	
	RecipientEmail	Optional	Email address of the contact that will be used to display personalization
Example	<Envelope> <Body> <PreviewMailing> <MailingId>59294</MailingId> </PreviewMailing> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS	TRUE if successful	
	HTMLBody	Returns the contents of the HTML body with personalization substituted.	
	AOLBody	Returns the contents of the AOL body with personalization substituted.	
	TextBody	Returns the contents of the Text body with personalization substituted.	
	SPAMScore	Returns the SPAM score based on the contents of the mailing template	
Example	<Envelope> <Body> <RESULT>		

	<SUCCESS>TRUE</SUCCESS> <HTMLBody><![CDATA[<html><HEAD><TITLE></TITLE></HEAD> <BODY> <P>Hello World,</P> <P>I am sending this test message with personalization:</P> %%FIRST_NAME%% </BODY></html>]]></HTMLBody> <AOLBody/> <TextBody/> <SpamScore>Score is: 2.5/5.0 ---- --------------------- -- 0.0 MIME_HTML_MOSTLY BODY: Multipart message mostly text/html MIME 0.0 HTML_MESSAGE BODY: HTML included in message 1.1 MPART_ALT_DIFF BODY: HTML and text parts are different 0.1 RDNS_NONE Delivered to trusted network by a host with no rDNS 1.2 AWL AWL: From: address is in the auto white-list</SpamScore> </RESULT> </Body> </Envelope>

Get Automated Message Group Details

GetMessageGroupDetails

This interface returns the description of an Automated Message Group.

Operation	**<GetMessageGroupDetails>**	
Elements	MESSAGE_GROUP_ID	Numeric identifier for the Automated Message Group. ***Note***: Due to legacy naming conventions, this Id is referred to as CAMPAIGN_ID in older API operations. The Campaign Id returned in the Raw Recipient Data Export file can be used as the MESSAGE_GROUP_ID.
Example	<Envelope> <Body> <GetMessageGroupDetails> <MESSAGE_GROUP_ID>59294</MESSAGE_GROUP_ID> </GetMessageGroupDetails> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	**TRUE** if successful.
	USER_ID	The user that created the Message group.
	NAME	The name of the Automated Message Group.
	STATUS	Indicates the status of the Group. Possible values are: • Draft • Active • Cancelled • Completed
	NOTES	Optional notes describing the Automated Message Group.
	LIST_ID	The Id of the database, contact list, or query associated with the Message Group.
	EVENT_TRIGGER	Indicates the event trigger for the automated messages. Possible values are: • OptInDate • CustomEventDate • CalendarDate • Recurring • Transact • None
	TRACKING_LEVEL	The tracking level for the automated messages. Possible values are: • Unique • Aggregate • Open Only • None
	CUSTOM_EVENT_DATE_COLUMN	When the EVENT_TRIGGER is "CustomEventDate", this element indicates the name of the database column containing the date. The date will determine when the automated mailing is sent.
	CALENDAR_DATE	When the EVENT_TRIGGER is "CalendarDate", this element indicates the date the automated mailing is sent.

	SEND_PREVIOUS	When the EVENT_TRIGGER is date based, this element indicates whether new contacts will be sent all previously delivered mailings. Valid values are: • true • false
	ACTIVATION_DATE	The date that the Automated Message Group was activated. This element will not contain a value when STATUS is "Draft".
	COMPLETION_DATE	The date that automated mailings will no longer be triggered.
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <USER_ID>12c734c-108b610e402-f528764d624db129b32c21fbcaocb8d6</USER_ID> <NAME>Welcome Campaign</NAME> <STATUS>Active</STATUS> <NOTES>Mailings will be sent when subscription begins.</NOTES> <LIST_ID>56432</LIST_ID> <EVENT_TRIGGER>CustomEventDate</EVENT_TRIGGER> <TRACKING_LEVEL>Unique</TRACKING_LEVEL> < CUSTOM_EVENT_DATE_COLUMN>Magazine Subscription Date</CUSTOM_EVENT_DATE_COLUMN> <ACTIVATION_DATE>01/29/2011</ACTIVATION_DATE> <COMPLETION_DATE>12/31/2011</COMPLETION_DATE> </RESULT> </Body> </Envelope>	

Add a Dynamic Content Ruleset

8.7 Note: Current Limitations on Dynamic Content vs. "Classic Dynamic Content:

In an effort to provide significant functional improvements to our customers, we have been adding new capabilities, as well as a fresh UI to the Dynamic Content in Engage. While this new functionality has been successfully delivered to the Engage UI, we have not yet updated our APIs to be compatible with these "new" Dynamic Content Rulesets. As a result, any API reference to a "new" Dynamic Content block or Ruleset should be avoided until upcoming product releases where we will begin to support all API methods for both "new" Dynamic Content and Classic Dynamic Content. All Classic Dynamic Content and Rulesets still work as expected with our APIs.

AddDCRuleset

This interface creates a Ruleset including its content areas, rules, and content. If user does not specify a MAILING_ID, Engage will create a new Ruleset in the user's Private folder of the Asset Library. **Note:** A ruleset may contain no more than 1,000 rules.

Operation	**<AddDCRuleset>**	
Elements	LIST_ID	The ID of the associated database.

	RULESET_NAME		The name of the DC Ruleset.			
	MAILING_ID		Optional	Associates a Ruleset directly to a mailing. This will NOT place a copy of the Ruleset in the asset library.		
Child Element	CONTENT_AREAS		The content area defines default content used when a user falls outside of all existing rule criteria as well as additional content areas in the mailing body.			
	Child Element	CONTENT_AREA	Optional	Included if content being defined. It is possible to create a Ruleset and associated rules in the asset library without related content areas.		
		Attribute	name	The Name of the Content Area being defined.		
			Type	The type of Content Area. The following values are valid: 'Body-HTML' 'Body-Text' 'Body-AOL' 'Subject' 'From' 'From-Name'		
		Child Element	DEFAULT_CONTENT	This is the content that will show if there is no matching rule. For HTML content, this must be CDATA.		
			Attribute	name	The name of the default Content Block	
Child Element	RULES					
	Child Element	RULE				
		Child Element	RULE_NAME	The Name of the Dynamic Content Rule.		
			PRIORITY	The priority of this rule vs. other rules. Number must be unique. Priority is determined in ascending order with '1' being the highest.		
			CRITERIA			
			Child Element	EXPRESSION		
				Child Element	AND_OR	Used to denote 'AND' or 'OR' when defining relationships between expressions.
					LEFT_PARENS	Used to denote a left parenthesis character when defining relationships between expressions.

					TYPE	The type of expression. Valid values are: ‘NE’ – Numeric ‘DE’ – Date/Time ‘TE’ – Text
					COLUMN_NAME	The name of the database field being evaluated.
					OPERATORS	The operator used for comparison. Valid values are: ‘!=’ ‘<’ ‘>’ ‘=’ ‘IS NOT null’ ‘IS null’ ‘LIKE’ ‘NOT like’ ‘IN’ ‘NOT in’ ‘BETWEEN’
					VALUES	The value being compared to the Engage database field. Could be text, numeric, or date value OR a list of values or the name of a Column. If specifying a Column, surround with square brackets. For example, [My Text Column]. If specifying a list of values, use | (pipe character) to separate values in the list.
					RIGHT_PARENS	Used to denote a right parenthesis character when defining relationships between expressions.
			CONTENTS			
			Child Element	CONTENT	Optional	This is the content that will show for contacts matching the criteria of this rule. For HTML content, this must be CDATA.

				Attribute	content_area	The previously defined Content Area where the content will be placed in the mailing body.
					Name	The name of the Content Block being defined.

Example

```
<Envelope>
    <Body>
        <AddDCRuleset>
            <RULESET_NAME>Promo By Gender</RULESET_NAME>
            <LIST_ID>22205</LIST_ID>
            <MAILING_ID>41781</MAILING_ID>
            <CONTENT_AREAS>
                <CONTENT_AREA name="dc_mailing_subject" type="Subject">
                    <DEFAULT_CONTENT name="Default.dc_mailing_subject"><![CDATA[All
clothing on sale this week.]]></DEFAULT_CONTENT>
                </CONTENT_AREA>
                <CONTENT_AREA name="dc_promo" type="Body-HTML">
                    <DEFAULT_CONTENT name="Default.dc_promo"><![CDATA[10% Off
Everything]]></DEFAULT_CONTENT>
                </CONTENT_AREA>
            </CONTENT_AREAS>
            <RULES>
                <RULE>
                    <RULE_NAME>MalePromotion</RULE_NAME>
                    <PRIORITY>1</PRIORITY>
                    <CRITERIA>
                        <EXPRESSION>
                            <AND_OR/>
                            <LEFT_PARENS>(</LEFT_PARENS>
                            <TYPE>TE</TYPE>
                            <COLUMN_NAME>Gender</COLUMN_NAME>
                            <OPERATORS><![CDATA[=]]></OPERATORS>
                            <VALUES><![CDATA[M]]></VALUES>
                            <RIGHT_PARENS>)</RIGHT_PARENS>
                        </EXPRESSION>
                        <EXPRESSION>
                            <AND_OR>AND</AND_OR>
                            <LEFT_PARENS>(</LEFT_PARENS>
                            <TYPE>NE</TYPE>
                            <COLUMN_NAME>Salary</COLUMN_NAME>
                            <OPERATORS><![CDATA[>]]></OPERATORS>
                            <VALUES><![CDATA[90000]]></VALUES>
                            <RIGHT_PARENS>)</RIGHT_PARENS>
                        </EXPRESSION>
                    </CRITERIA>
                    <CONTENTS>
                        <CONTENT name="male_subject"
content_area="dc_mailing_subject"><![CDATA[Men's clothing on sale this week.]]></CONTENT>
                        <CONTENT name="male_promo"
content_area="dc_promo"><![CDATA[10% Off All Men's Clothing]]></CONTENT>
                    </CONTENTS>
                </RULE>
```

```
                    <RULE>
                        <RULE_NAME>Female Promotion</RULE_NAME>
                        <PRIORITY>2</PRIORITY>
                        <CRITERIA>
                            <EXPRESSION>
                                <AND_OR/>
                                <LEFT_PARENS>(</LEFT_PARENS>
                                <TYPE>TE</TYPE>
                                <COLUMN_NAME>Gender</COLUMN_NAME>
                                <OPERATORS><![CDATA[=]]></OPERATORS>
                                <VALUES><![CDATA[F]]></VALUES>
                                <RIGHT_PARENS>)</RIGHT_PARENS>
                            </EXPRESSION>
                            <EXPRESSION>
                                <AND_OR>AND</AND_OR>
                                <LEFT_PARENS>(</LEFT_PARENS>
                                <TYPE>NE</TYPE>
                                <COLUMN_NAME>Salary</COLUMN_NAME>
                                <OPERATORS><![CDATA[>]]></OPERATORS>
                                <VALUES><![CDATA[90000]]></VALUES>
                                <RIGHT_PARENS>)</RIGHT_PARENS>
                            </EXPRESSION>
                        </CRITERIA>
                        <CONTENTS>

    <CONTENT name="female_subject" content_area="dc_mailing_subject"><![CDATA[Women's clothing on sale
this week.]]></CONTENT>
                            <CONTENT name="female_promo"
content_area="dc_promo"><![CDATA[10% Off All Women's Clothing]]></CONTENT>
                        </CONTENTS>
                    </RULE>
                </RULES>
            </AddDCRuleset>
    </Body>
</Envelope>
```

Response	**<RESULT>**	
Elements	SUCCESS	"True" indicates the rule creation was successful. "False" indicates an error occurred in which the error code displays with a message reporting the issue.
	RULESET_ID	Contains the new Ruleset ID.

Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <RULESET_ID>28217</RULESET_ID> </RESULT> </Body> </Envelope>

Import a Dynamic Content Ruleset

ImportDCRuleset

This interface imports a .zip containing the XML file defining a **New or Classic Dynamic** Content Ruleset and any associated HTML or images. Imported files are loaded from the user's FTP upload directory.

If the user does not specify a MAILING_ID or RULESET_ID, Engage creates a new Ruleset in the user's Private folder of the Asset Library. If the user does specify a RULESET_ID, Engage replaces the Ruleset.

Note: A ruleset may contain no more than 1,000 rules.

Operation	**<ImportDCRuleset>**		
Elements	FILE_NAME	The name of the .zip file located in the user's FTP upload directory.	
	LIST_ID	The ID of the associated database.	
	MAILING_ID	Optional	Associates a Ruleset directly to a mailing. This will NOT place a copy of the Ruleset in the Asset Library.
	RULESET_ID	Optional	Specifies an existing Ruleset to use in place of imported content.
Example	<Envelope> <Body> <ImportDCRuleset> <FILE_NAME>promo.zip</FILE_NAME> <MAILING_ID>41781</MAILING_ID> <LIST_ID>22205</LIST_ID> </ImportDCRuleset> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS	"True" indicates successful Data Job submission. "False" indicates an error occurred in which the error code displayed with a message reporting the issue.	
	JOB_ID	Contains the Data Job ID for the import.	
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <JOB_ID>28216</JOB_ID> </RESULT> </Body> </Envelope>		

The following table describes the .zip file format for **Classic Dynamic Content Ruleset**

RULESET						
Child Element	RULESET_NAME		Optional	The name of the DC Ruleset if creating a new Ruleset using Import.		
	CONTENT_AREAS		The content area defines default content used when a user falls outside of all existing rule criteria.			
	Child Element	CONTENT_AREA	Optional	Included if content being defined. It is possible to create a Ruleset and associated rules in the Asset Library without related content areas.		
		Attribute	name	The name of the Content Area being defined		
			type	The type of Content Area. The following values are valid: 'Body-HTML' 'Body-Text' 'Body-AOL' 'Subject' 'From' 'From-Name'		
		Child Element	DEFAULT_CONTENT	Content that displays if there is no matching rule. Must be set to CDATA for HTML content.		
			Attribute	name	The name of the Content Block being defined.	
Child Element	RULES					
	Child Element	RULE				
		Child Element	RULE_NAME	The name of the Dynamic Content Rule.		
			PRIORITY	The priority of the rule vs. other rules. Number must be unique. Priority is determined in ascending order with '1' being the highest.		
			CRITERIA			
			Child Element	EXPRESSION		
				Child Element	AND_OR	Used to denote 'AND' or 'OR' when defining relationships between expressions.
					LEFT_PARENS	Used to denote a left parenthesis character when defining relationships between expressions.

					TYPE	The type of expression. Valid values are: 'NE' – Numeric 'DE' – Date/Time 'TE' – Text
					COLUMN_NAME	The name of the database field being evaluated.
					OPERATORS	The operator used for comparison. Valid values are: '!=' '<' '>' '=' 'IS NOT null' 'IS null' 'LIKE' 'NOT like' 'IN' 'NOT in' 'BETWEEN'
					VALUES	The value being compared to the Engage database field. Can be text, numeric, or date value OR a list of values or the name of a Column. If specifying a Column, surround with square brackets, for example, [My Text Column]. If specifying a list of values, use | (pipe character) to separate values in the list.
					RIGHT_PARENS	Used to denote a right parenthesis character when defining relationships between expressions.
			CONTENTS			

			Child Element	CONTENT	Optional	This is the content that will show for contacts matching the criteria of this rule. For HTML content, this must be CDATA.
				Attribute	content_area	The previously defined Content Area where the content will be placed in the mailing body.
					Name	The name of the Content Block being defined.

Example

```
<RULESET>
	<RULESET_NAME>Promo By Gender</RULESET_NAME>
	<CONTENT_AREAS>
		<CONTENT_AREA name=”dc_mailing_subject” type=”Subject”>
			<DEFAULT_CONTENT name=”Default.dc_mailing_subject”><![CDATA[All clothing on sale this week.]]></DEFAULT_CONTENT>
		</CONTENT_AREA>
		<CONTENT_AREA name=”dc_promo” type=”Body-HTML”>
			<DEFAULT_CONTENT name=”Default.dc_promo”><![CDATA[10% Off Everything]]></DEFAULT_CONTENT>
		</CONTENT_AREA>
	</CONTENT_AREAS>
	<RULES>
		<RULE>
			<RULE_NAME>MalePromotion</RULE_NAME>
			<PRIORITY>1</PRIORITY>
			<CRITERIA>
				<EXPRESSION>
					<AND_OR/>
					<LEFT_PARENS>(</LEFT_PARENS>
					<TYPE>TE</TYPE>
					<COLUMN_NAME>Gender</COLUMN_NAME>
					<OPERATORS><![CDATA[=]]></OPERATORS>
					<VALUES><![CDATA[M]]></VALUES>
					<RIGHT_PARENS>)</RIGHT_PARENS>
				</EXPRESSION>
				<EXPRESSION>
					<AND_OR>AND</AND_OR>
					<LEFT_PARENS>(</LEFT_PARENS>
					<TYPE>NE</TYPE>
					<COLUMN_NAME>Salary</COLUMN_NAME>
					<OPERATORS><![CDATA[>]]></OPERATORS>
					<VALUES><![CDATA[90000]]></VALUES>
					<RIGHT_PARENS>)</RIGHT_PARENS>
				</EXPRESSION>
			</CRITERIA>
```

```
                <CONTENTS>
                        <CONTENT name="male_subject"
content_area="dc_mailing_subject"><![CDATA[Men's clothing on sale this week.]]></CONTENT>
                        <CONTENT name="male_promo" content_area="dc_promo"><![CDATA[20%
Off All Men's Clothing]]></CONTENT>
                </CONTENTS>
        </RULE>
        <RULE>
                <RULE_NAME>Female Promotion</RULE_NAME>
                <PRIORITY>2</PRIORITY>
                <CRITERIA>
                        <EXPRESSION>
                                <AND_OR/>
                                <LEFT_PARENS>(</LEFT_PARENS>
                                <TYPE>TE</TYPE>
                                <COLUMN_NAME>Gender</COLUMN_NAME>
                                <OPERATORS><![CDATA[=]]></OPERATORS>
                                <VALUES><![CDATA[F]]></VALUES>
                                <RIGHT_PARENS>)</RIGHT_PARENS>
                        </EXPRESSION>
                </CRITERIA>
                <CONTENTS>
                        <CONTENT name="female_promo" content_area="dc_promo"><![CDATA[20%
Off All Women's Clothing]]></CONTENT>
                        <CONTENT name="female_subject"
content_area="dc_mailing_subject"><![CDATA[Women's clothing on sale this week.]]></CONTENT>
                </CONTENTS>
        </RULE>
    </RULES>
</RULESET>
```

The following table describes the .zip file format for **New Dynamic Content Ruleset**.

RULESET					
Child Element	RULEBUILDER_TYPE		Optional	XML node to distinguish between Classic and New DC Ruleset. Valid value for new Dynamic Content Rule set: • Rulebuilder • Classic Rulebuilder NOTE: if you don't specify this element, it will try to create a Classic Dynamic Content Ruleset	
	RULESET_NAME		Optional	The name of the DC Ruleset if creating a new Ruleset using Import.	
	CONTENT_AREAS		The content area defines default content used when a user falls outside of all existing rule criteria.		
	Child Element	CONTENT_AREA	Optional	Included if content being defined. It is possible to create a Ruleset and associated rules in the Asset Library without related content areas.	
		Attribute	name	The name of the Content Area being defined	
			type	The type of Content Area. The following values are valid: 'Body-HTML' 'Body-Text' 'Body-AOL' 'Subject' 'From' 'From-Name'	
		Child Element	DEFAULT_CONTENT	Content that displays if there is no matching rule. Must be set to CDATA for HTML content.	
			Attribute	name	The name of the Content Block being defined.
	RULES				
	Child Element	RULE			
		Child Element	RULE_NAME	The name of the Dynamic Content Rule.	
			PRIORITY	The priority of the rule vs. other rules. Number must be unique. Priority is determined in ascending order with '1' being the highest.	
	CRITERIA				
	Child Element	RT_EXPRESSION		Describes the expressions to perform on one or more columns in the relational table.	

	Child Element	EXPRESSION		Describes the expressions to perform on one or more columns in the database. It is used with combination of three attributes depending upon the Query that is being created.	
		Attributes	criteria_type	Different criteria type used for Queries. Valid Values are: • profile • behavior • crm • relational_table • rt_profile	
		Child Element	COLUMN	Optional	The name of the database field to query against.
			OPERATOR	Optional	Used for value comparison for different criteria types. Valid values are : • is null • is not null • contains • does not contain • starts with • does not start with • ends with • does not end with • is equal to • is not equal to • contains one of the following • does not contain one of the following • starts with one of the following • does not start with one of the following • is equal to one of the following • is not equal to one of the following • is blank • is not blank • anniversary is today • is today • is after today • is not after today • is before today • is not before today • is after • is before • year is equal to • month is equal to • day is equal to

					• is within the last • is not within the last • is within the next • is not the within the next • is exactly • is not exactly • anniversary is • is between • is within the last range of • is less than • is less than or equal to • is greater than • is great than or equal to • is between • is in contact list • is not in contact list • is in query • is not in query • has been sent • has clicked link in email • has opened • has not been sent • has not clicked link in email • has not opened • has been in program • has not been in program • has submitted a web form • has not submitted a web form • has visited webpage • has visited web site • has downloaded file • has not vied webpage • has visited web site • has downloaded file • has triggered custom tracking event Values specific to relational table criteria • match • do not match Values specific to CRM criteria • stages • not in stages
			TIMEFRAME	Optional	Used for value comparison when criteria type is

					'behavior'. Valid values are: • at any time • on today • before today • on • before • after • between • within the last • exactly • more than • less than
			VALUE	Optional	The value being compared to the Engage database field. This can include text or numeric or date value. This element is ONLY used when specifying multiple values in an expression. It should be always wrapped around with <VALUES> element.
			START_VALUE	Optional	Used for start value.
			END_VALUE	Optional	Used for end value.
			TIME_UNIT	Optional	Valid values are : • weeks • days • months
			UNIT	Optional	Valid values are : • ago • away
			EMAIL_LINK	Optional	Used to specify the link name in the email.
			PARENS	Optional	Used to specify one or more left or right parentheses when relating more than one expression. For example, left parentheses would be indicated as **(** and right would be indicated as **)**. You can specify more than one right or left parentheses as: **))** or **((** .
			CONJUNCTION	Optional	Used to specify **AND** or **OR** when relating multiple expressions.
			ID	Optional	This element may contain • The ID of the mailing on which to search the contact activity

					• The ID of the program on which to search the contact activity • Relational Table ID when criteria type is 'relational_table' • Contact List ID • Query ID	
			WEB_FORM_NAME	Optional	The name of the web form name on which to search the contact activity.	
			WEB_PAGE	Optional	The name of the web page on which to search the contact activity. Must include 'http(s)://'.	
			WEB_SITE	Optional	The website link on which to search the contact activity.	
			EVENT_NAME	Optional	Custom Web-tracking Event	
			FRIENDLY_NAME	Optional	Friendly name for Custom Web-tracking Event	
				Child Element	ANY	Optional element to select *Any* event for Friendly Name
			FILE	Optional	Downloaded File Name f or 'has downloaded file' event	

Below example shows all functional operators and elements that can be used inside a Rule

```
<RULESET>
    <RULEBUILDER_TYPE>Rulebuilder</RULEBUILDER_TYPE>
    <RULESET_NAME>Promo By Gender</RULESET_NAME>
    <CONTENT_AREAS>
        <CONTENT_AREA name="dc_mailing_subject" type="Subject">
            <DEFAULT_CONTENT name="Default.dc_mailing_subject"><![CDATA[All clothing on sale this week.]]></DEFAULT_CONTENT>
        </CONTENT_AREA>
        <CONTENT_AREA name="dc_promo" type="Body-HTML">
            <DEFAULT_CONTENT name="Default.dc_promo"><![CDATA[10% Off Everything]]></DEFAULT_CONTENT>
        </CONTENT_AREA>
    </CONTENT_AREAS>
    <RULES>
        <RULE>
            <RULE_NAME>MalePromotion</RULE_NAME>
            <PRIORITY>1</PRIORITY>
```

```
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>is after</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUE><![CDATA[11/27/2012]]></VALUE>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <CONJUNCTION>AND</CONJUNCTION>
        <PARENS>(</PARENS>
        <OPERATOR>is blank</OPERATOR>
        <COLUMN>foo</COLUMN>
    </EXPRESSION>
    <EXPRESSION criteria_type="profile">
        <CONJUNCTION>AND</CONJUNCTION>
        <OPERATOR>is not blank</OPERATOR>
        <COLUMN>foo</COLUMN>
        <PARENS>)</PARENS>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>anniversary is</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUE><![CDATA[2]]></VALUE>
        <TIME_UNIT>weeks</TIME_UNIT>
        <UNIT>ago</UNIT>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>anniversary is</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUE><![CDATA[2]]></VALUE>
        <TIME_UNIT>weeks</TIME_UNIT>
        <UNIT>away</UNIT>
```

```
        </EXPRESSION>
    </CRITERIA>
    <CRITERIA>
        <EXPRESSION criteria_type="profile">
            <OPERATOR>is before</OPERATOR>
            <COLUMN>foo</COLUMN>
            <VALUE><![CDATA[11/27/2012]]></VALUE>
        </EXPRESSION>
    </CRITERIA>
    <CRITERIA>
        <EXPRESSION criteria_type="profile">
            <OPERATOR>is before today</OPERATOR>
            <COLUMN>foo</COLUMN>
        </EXPRESSION>
    </CRITERIA>
    <CRITERIA>
        <EXPRESSION criteria_type="profile">
            <OPERATOR>is between</OPERATOR>
            <COLUMN>foo</COLUMN>
            <START_VALUE><![CDATA[10]]></START_VALUE>
            <END_VALUE><![CDATA[20]]></END_VALUE>
        </EXPRESSION>
    </CRITERIA>
    <CRITERIA>
        <EXPRESSION criteria_type="profile">
            <OPERATOR>is blank</OPERATOR>
            <COLUMN>foo</COLUMN>
        </EXPRESSION>
    </CRITERIA>
    <CRITERIA>
        <EXPRESSION criteria_type="behavior">
            <OPERATOR>has clicked link in email</OPERATOR>
            <ID>2</ID>
            <EMAIL_LINK><![CDATA[abc]]></EMAIL_LINK>
        </EXPRESSION>
    </CRITERIA>
    <CRITERIA>
```

```
<EXPRESSION criteria_type="behavior">
    <OPERATOR>has clicked link in email</OPERATOR>
    <ID>2</ID>
    <EMAIL_LINK><![CDATA[abc]]></EMAIL_LINK>
</EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>is in contact list</OPERATOR>
        <ID>1</ID>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>is in contact list</OPERATOR>
        <ID>1234</ID>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>contains one of the following</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUES>
            <VALUE><![CDATA[[columnA]]]></VALUE>
            <VALUE><![CDATA[[193olumn]]]></VALUE>
            <VALUE><![CDATA[[193olumn]]]></VALUE>
        </VALUES>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>contains one of the following</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUES>
            <VALUE><![CDATA[a]]></VALUE>
            <VALUE><![CDATA[b]]></VALUE>
            <VALUE><![CDATA[c]]></VALUE>
```

	<VALUE><![CDATA[abc,efg]]></VALUE> </VALUES> </EXPRESSION> </CRITERIA> <CRITERIA> <EXPRESSION criteria_type="profile"> <OPERATOR>contains</OPERATOR> <COLUMN>foo</COLUMN> <VALUE><![CDATA[abc]]></VALUE> </EXPRESSION> </CRITERIA> <CRITERIA> <EXPRESSION criteria_type="behavior"> <OPERATOR>has triggered custom tracking event</OPERATOR> <EVENT_NAME><![CDATA[testEvent]]></EVENT_NAME> <FRIENDLY_NAME><![CDATA[testFriendlyName]]></FRIENDLY_NAME> </EXPRESSION> </CRITERIA> <CRITERIA> <EXPRESSION criteria_type="behavior"> <OPERATOR>has triggered custom tracking event</OPERATOR> <EVENT_NAME><![CDATA[testEvent]]></EVENT_NAME> <FRIENDLY_NAME> <ANY>true</ANY> </FRIENDLY_NAME> </EXPRESSION> </CRITERIA> <CRITERIA> <EXPRESSION criteria_type="profile"> <OPERATOR>day is equal to</OPERATOR> <COLUMN>foo</COLUMN> <VALUE><![CDATA[1]]></VALUE> </EXPRESSION> </CRITERIA> <CRITERIA> <EXPRESSION criteria_type="behavior">

```
                    <OPERATOR>has downloaded file</OPERATOR>
                    <FILE><![CDATA[test.url1]]></FILE>
                    <TIMEFRAME>on today</TIMEFRAME>
               </EXPRESSION>
          </CRITERIA>
          <CRITERIA>
               <EXPRESSION criteria_type="profile">
                    <OPERATOR>is equal to</OPERATOR>
                    <COLUMN>EMAIL_TYPE</COLUMN>
                    <VALUE><![CDATA[AOL]]></VALUE>
               </EXPRESSION>
          </CRITERIA>
          <CRITERIA>
               <EXPRESSION criteria_type="profile">
                    <OPERATOR>is equal to</OPERATOR>
                    <COLUMN>EMAIL_TYPE</COLUMN>
                    <VALUE><![CDATA[HTML]]></VALUE>
               </EXPRESSION>
          </CRITERIA>
          <CRITERIA>
               <EXPRESSION criteria_type="profile">
                    <OPERATOR>is equal to</OPERATOR>
                    <COLUMN>EMAIL_TYPE</COLUMN>
                    <VALUE><![CDATA[TEXT]]></VALUE>
               </EXPRESSION>
          </CRITERIA>
          <CRITERIA>
               <EXPRESSION criteria_type="profile">
                    <OPERATOR>is equal to one of the following</OPERATOR>
                    <COLUMN>EMAIL_TYPE</COLUMN>
                    <VALUES>
                         <VALUE><![CDATA[0]]></VALUE>
                         <VALUE><![CDATA[1]]></VALUE>
                         <VALUE><![CDATA[2]]></VALUE>
                    </VALUES>
               </EXPRESSION>
          </CRITERIA>
```

```
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>ends with</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUE><![CDATA[abc]]></VALUE>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>is equal to</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUE><![CDATA[[abc]]]></VALUE>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>is equal to one of the following</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUES>
            <VALUE><![CDATA[[columnA]]]></VALUE>
            <VALUE><![CDATA[[196olumn]]]></VALUE>
            <VALUE><![CDATA[[196olumn]]]></VALUE>
        </VALUES>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>is equal to one of the following</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUES>
            <VALUE><![CDATA[a]]></VALUE>
            <VALUE><![CDATA[b]]></VALUE>
            <VALUE><![CDATA[c]]></VALUE>
            <VALUE><![CDATA[abc,efg]]></VALUE>
        </VALUES>
    </EXPRESSION>
</CRITERIA>
```

```
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>is equal to</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUE><![CDATA[abc]]></VALUE>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>is exactly</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUE><![CDATA[2]]></VALUE>
        <TIME_UNIT>weeks</TIME_UNIT>
        <UNIT>ago</UNIT>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>is exactly</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUE><![CDATA[2]]></VALUE>
        <TIME_UNIT>weeks</TIME_UNIT>
        <UNIT>away</UNIT>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <CONJUNCTION>AND</CONJUNCTION>
        <OPERATOR>is blank</OPERATOR>
        <COLUMN>foo</COLUMN>
    </EXPRESSION>
    <EXPRESSION criteria_type="profile">
        <CONJUNCTION>AND</CONJUNCTION>
        <OPERATOR>is not blank</OPERATOR>
        <COLUMN>foo</COLUMN>
    </EXPRESSION>
</CRITERIA>
```

```xml
<CRITERIA>
    <EXPRESSION criteria_type="behavior">
        <OPERATOR>has been in program</OPERATOR>
        <ID>1234</ID>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>is greater than</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUE><![CDATA[10]]></VALUE>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>is greater than or equal to</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUE><![CDATA[10]]></VALUE>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>is within the last range of</OPERATOR>
        <COLUMN>foo</COLUMN>
        <START_VALUE><![CDATA[10]]></START_VALUE>
        <END_VALUE><![CDATA[20]]></END_VALUE>
        <TIME_UNIT>weeks</TIME_UNIT>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>is less than</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUE><![CDATA[10]]></VALUE>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
```

```
                <EXPRESSION criteria_type="profile">
                        <OPERATOR>is less than or equal to</OPERATOR>
                        <COLUMN>foo</COLUMN>
                        <VALUE><![CDATA[10]]></VALUE>
                </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
                <EXPRESSION criteria_type="behavior">
                        <OPERATOR>has been in program</OPERATOR>
                        <ID>1</ID>
                </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
                <EXPRESSION criteria_type="profile">
                        <OPERATOR>month is equal to</OPERATOR>
                        <COLUMN>foo</COLUMN>
                        <VALUE><![CDATA[8]]></VALUE>
                </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
                <EXPRESSION criteria_type="profile">
                        <OPERATOR>is not before today</OPERATOR>
                        <COLUMN>foo</COLUMN>
                </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
                <EXPRESSION criteria_type="profile">
                        <OPERATOR>is not blank</OPERATOR>
                        <COLUMN>foo</COLUMN>
                </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
                <EXPRESSION criteria_type="behavior">
                        <OPERATOR>has not clicked link in email</OPERATOR>
                        <ID>2</ID>
                        <EMAIL_LINK><![CDATA[abc]]></EMAIL_LINK>
                </EXPRESSION>
```

```
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="behavior">
        <OPERATOR>has not clicked link in email</OPERATOR>
        <ID>2</ID>
        <EMAIL_LINK><![CDATA[abc]]></EMAIL_LINK>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>is not in contact list</OPERATOR>
        <ID>1</ID>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>does not contain one of the following</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUES>
            <VALUE><![CDATA[[columnA]]]></VALUE>
            <VALUE><![CDATA[[2000lumn]]]></VALUE>
            <VALUE><![CDATA[[2000lumn]]]></VALUE>
        </VALUES>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <OPERATOR>does not contain one of the following</OPERATOR>
        <COLUMN>foo</COLUMN>
        <VALUES>
            <VALUE><![CDATA[a]]></VALUE>
            <VALUE><![CDATA[b]]></VALUE>
            <VALUE><![CDATA[c]]></VALUE>
            <VALUE><![CDATA[abc,efg]]></VALUE>
        </VALUES>
    </EXPRESSION>
</CRITERIA>
```

```
            <CRITERIA>
                <EXPRESSION criteria_type="profile">
                    <OPERATOR>does not contain</OPERATOR>
                    <COLUMN>foo</COLUMN>
                    <VALUE><![CDATA[abc]]></VALUE>
                </EXPRESSION>
            </CRITERIA>
            <CRITERIA>
                <EXPRESSION criteria_type="behavior">
                    <OPERATOR>has not triggered custom tracking event</OPERATOR>
                    <EVENT_NAME><![CDATA[testEvent]]></EVENT_NAME>

    <FRIENDLY_NAME><![CDATA[testFriendlyName]]></FRIENDLY_NAME>
                </EXPRESSION>
            </CRITERIA>
            <CRITERIA>
                <EXPRESSION criteria_type="behavior">
                    <OPERATOR>has not downloaded file</OPERATOR>
                    <FILE><![CDATA[test.url1]]></FILE>
                    <TIMEFRAME>on today</TIMEFRAME>
                </EXPRESSION>
            </CRITERIA>
            <CRITERIA>
                <EXPRESSION criteria_type="profile">
                    <OPERATOR>does not end with</OPERATOR>
                    <COLUMN>foo</COLUMN>
                    <VALUE><![CDATA[abc]]></VALUE>
                </EXPRESSION>
            </CRITERIA>
            <CRITERIA>
                <EXPRESSION criteria_type="profile">
                    <OPERATOR>is not equal to one of the following</OPERATOR>
                    <COLUMN>foo</COLUMN>
                    <VALUES>
                        <VALUE><![CDATA[[columnA]]]></VALUE>
                        <VALUE><![CDATA[[2010lumn]]]></VALUE>
                        <VALUE><![CDATA[[2010lumn]]]></VALUE>
```

	``` </VALUES> </EXPRESSION> </CRITERIA> <CRITERIA> <EXPRESSION criteria_type="profile"> <OPERATOR>is not equal to one of the following</OPERATOR> <COLUMN>foo</COLUMN> <VALUES> <VALUE><![CDATA[a]]></VALUE> <VALUE><![CDATA[b]]></VALUE> <VALUE><![CDATA[c]]></VALUE> <VALUE><![CDATA[abc,efg]]></VALUE> </VALUES> </EXPRESSION> </CRITERIA> <CRITERIA> <EXPRESSION criteria_type="profile"> <OPERATOR>is not equal to</OPERATOR> <COLUMN>foo</COLUMN> <VALUE><![CDATA[abc]]></VALUE> </EXPRESSION> </CRITERIA> <CRITERIA> <EXPRESSION criteria_type="profile"> <OPERATOR>is not exactly</OPERATOR> <COLUMN>foo</COLUMN> <VALUE><![CDATA[2]]></VALUE> <TIME_UNIT>weeks</TIME_UNIT> <UNIT>ago</UNIT> </EXPRESSION> </CRITERIA> <CRITERIA> <EXPRESSION criteria_type="profile"> <OPERATOR>is not exactly</OPERATOR> <COLUMN>foo</COLUMN> <VALUE><![CDATA[2]]></VALUE> <TIME_UNIT>weeks</TIME_UNIT> ```

```
                <UNIT>away</UNIT>
            </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
            <EXPRESSION criteria_type="behavior">
                <OPERATOR>has not been in program</OPERATOR>
                <ID>1</ID>
            </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
            <EXPRESSION criteria_type="behavior">
                <OPERATOR>has not opened email</OPERATOR>
                <ID>2</ID>
            </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
            <EXPRESSION criteria_type="crm">
                <OPERATOR>is not associated with any opportunity</OPERATOR>
            </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
            <EXPRESSION criteria_type="crm">
                <OPERATOR>is not in stages</OPERATOR>
                <VALUES>
                    <VALUE><![CDATA[abc]]></VALUE>
                    <VALUE><![CDATA[xyz]]></VALUE>
                </VALUES>
            </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
            <EXPRESSION criteria_type="behavior">
                <OPERATOR>is not in query</OPERATOR>
                <ID>1</ID>
            </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
            <EXPRESSION criteria_type="behavior">
```

```
                        <OPERATOR>has not been sent email</OPERATOR>
                        <ID>2</ID>
                </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
                <EXPRESSION criteria_type="profile">
                        <OPERATOR>does not start with one of the following</OPERATOR>
                        <COLUMN>foo</COLUMN>
                        <VALUES>
                                <VALUE><![CDATA[a]]></VALUE>
                                <VALUE><![CDATA[b]]></VALUE>
                                <VALUE><![CDATA[c]]></VALUE>
                                <VALUE><![CDATA[abc,efg]]></VALUE>
                        </VALUES>
                </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
                <EXPRESSION criteria_type="profile">
                        <OPERATOR>does not start with</OPERATOR>
                        <COLUMN>foo</COLUMN>
                        <VALUE><![CDATA[abc]]></VALUE>
                </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
                <EXPRESSION criteria_type="behavior">
                        <OPERATOR>has not submitted web form</OPERATOR>
                        <WEB_FORM_NAME><![CDATA[my web form
name]]></WEB_FORM_NAME>
                </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
                <EXPRESSION criteria_type="behavior">
                        <OPERATOR>has not visited web page</OPERATOR>
                        <WEB_PAGE><![CDATA[http://www.xyz.com]]></WEB_PAGE>
                </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
```

```
            <EXPRESSION criteria_type="behavior">
                <OPERATOR>has not visited web site</OPERATOR>
                <WEB_SITE><![CDATA[my web site name]]></WEB_SITE>
            </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
            <EXPRESSION criteria_type="profile">
                <OPERATOR>is not within the last</OPERATOR>
                <COLUMN>foo</COLUMN>
                <VALUE><![CDATA[5]]></VALUE>
                <TIME_UNIT>months</TIME_UNIT>
            </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
            <EXPRESSION criteria_type="profile">
                <OPERATOR>is not within the next</OPERATOR>
                <COLUMN>foo</COLUMN>
                <VALUE><![CDATA[5]]></VALUE>
                <TIME_UNIT>months</TIME_UNIT>
            </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
            <EXPRESSION criteria_type="behavior">
                <OPERATOR>has opened email</OPERATOR>
                <ID>2</ID>
            </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
            <EXPRESSION criteria_type="crm">
                <OPERATOR>is associated with any opportunity</OPERATOR>
            </EXPRESSION>
        </CRITERIA>
        <CRITERIA>
            <EXPRESSION criteria_type="crm">
                <OPERATOR>is in stages</OPERATOR>
                <VALUES>
                    <VALUE><![CDATA[abc]]></VALUE>
```

	``` <VALUE><![CDATA[xyz]]></VALUE> </VALUES> </EXPRESSION> </CRITERIA> <CRITERIA> <EXPRESSION criteria_type="profile"> <CONJUNCTION>AND</CONJUNCTION> <PARENS>(</PARENS> <OPERATOR>is blank</OPERATOR> <COLUMN>foo</COLUMN> </EXPRESSION> <EXPRESSION criteria_type="profile"> <CONJUNCTION>OR</CONJUNCTION> <OPERATOR>is not blank</OPERATOR> <COLUMN>foo</COLUMN> <PARENS>)</PARENS> </EXPRESSION> </CRITERIA> <CRITERIA> <EXPRESSION criteria_type="profile"> <CONJUNCTION>AND</CONJUNCTION> <PARENS>((</PARENS> <OPERATOR>is not blank</OPERATOR> <COLUMN>foo</COLUMN> </EXPRESSION> <EXPRESSION criteria_type="profile"> <CONJUNCTION>AND</CONJUNCTION> <OPERATOR>is today</OPERATOR> <COLUMN>foo</COLUMN> <PARENS>)</PARENS> </EXPRESSION> <EXPRESSION criteria_type="profile"> <CONJUNCTION>OR</CONJUNCTION> <OPERATOR>is blank</OPERATOR> <COLUMN>foo</COLUMN> <PARENS>)</PARENS> </EXPRESSION> ```

```
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="profile">
        <CONJUNCTION>AND</CONJUNCTION>
        <PARENS>(</PARENS>
        <OPERATOR>is blank</OPERATOR>
        <COLUMN>foo</COLUMN>
    </EXPRESSION>
    <EXPRESSION criteria_type="profile">
        <CONJUNCTION>OR</CONJUNCTION>
        <PARENS>(</PARENS>
        <OPERATOR>is not blank</OPERATOR>
        <COLUMN>foo</COLUMN>
    </EXPRESSION>
    <EXPRESSION criteria_type="profile">
        <CONJUNCTION>AND</CONJUNCTION>
        <OPERATOR>is today</OPERATOR>
        <COLUMN>foo</COLUMN>
        <PARENS>))</PARENS>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="behavior">
        <OPERATOR>has been in program</OPERATOR>
        <ID>1</ID>
        <TIMEFRAME>at any time</TIMEFRAME>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="behavior">
        <OPERATOR>has been in program</OPERATOR>
        <ID>1</ID>
        <TIMEFRAME>between</TIMEFRAME>
        <START_VALUE><![CDATA[01/01/1970]]></START_VALUE>
        <END_VALUE><![CDATA[01/02/1970]]></END_VALUE>
    </EXPRESSION>
</CRITERIA>
```

```
<CRITERIA>
    <EXPRESSION criteria_type="behavior">
        <OPERATOR>has been in program</OPERATOR>
        <ID>1</ID>
        <TIMEFRAME>exactly</TIMEFRAME>
        <VALUE><![CDATA[450]]></VALUE>
        <TIME_UNIT>days</TIME_UNIT>
        <UNIT>ago</UNIT>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="behavior">
        <OPERATOR>has been in program</OPERATOR>
        <ID>1</ID>
        <TIMEFRAME>before</TIMEFRAME>
        <VALUE><![CDATA[01/01/1970]]></VALUE>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="behavior">
        <OPERATOR>has been in program</OPERATOR>
        <ID>1</ID>
        <TIMEFRAME>within the last</TIMEFRAME>
        <VALUE><![CDATA[450]]></VALUE>
        <TIME_UNIT>days</TIME_UNIT>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="behavior">
        <OPERATOR>is in query</OPERATOR>
        <ID>1</ID>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type="relational_table">
        <CONJUNCTION>AND</CONJUNCTION>
        <OPERATOR>not match</OPERATOR>
```

	``` <ID>123</ID> <RT_EXPRESSIONS> <EXPRESSION criteria_type="rt_profile"> <CONJUNCTION>AND</CONJUNCTION> <OPERATOR>starts with</OPERATOR> <COLUMN>foo</COLUMN> <VALUE><![CDATA[COLUMN1]]></VALUE> </EXPRESSION> <EXPRESSION criteria_type="rt_profile"> <CONJUNCTION>AND</CONJUNCTION> <OPERATOR>is blank</OPERATOR> <COLUMN>foo</COLUMN> </EXPRESSION> </RT_EXPRESSIONS> </EXPRESSION> <EXPRESSION criteria_type="profile"> <CONJUNCTION>AND</CONJUNCTION> <OPERATOR>is blank</OPERATOR> <COLUMN>foo</COLUMN> </EXPRESSION> </CRITERIA> <CRITERIA> <EXPRESSION criteria_type="profile"> <CONJUNCTION>AND</CONJUNCTION> <OPERATOR>is today</OPERATOR> <COLUMN>foo</COLUMN> </EXPRESSION> <EXPRESSION criteria_type="relational_table"> <CONJUNCTION>OR</CONJUNCTION> <OPERATOR>match</OPERATOR> <ID>123</ID> <RT_EXPRESSIONS> <EXPRESSION criteria_type="rt_profile"> <CONJUNCTION>AND</CONJUNCTION> <OPERATOR>starts with</OPERATOR> <COLUMN>foo</COLUMN> <VALUE><![CDATA[COLUMN1]]></VALUE> ```

```
            </EXPRESSION>
            <EXPRESSION criteria_type=”rt_profile”>
                <CONJUNCTION>AND</CONJUNCTION>
                <OPERATOR>is blank</OPERATOR>
                <COLUMN>foo</COLUMN>
            </EXPRESSION>
        </RT_EXPRESSIONS>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type=”relational_table”>
        <OPERATOR>match</OPERATOR>
        <ID>123</ID>
        <RT_EXPRESSIONS>
            <EXPRESSION criteria_type=”rt_profile”>
                <CONJUNCTION>AND</CONJUNCTION>
                <OPERATOR>starts with</OPERATOR>
                <COLUMN>foo</COLUMN>
                <VALUE><![CDATA[COLUMN1]]></VALUE>
            </EXPRESSION>
            <EXPRESSION criteria_type=”rt_profile”>
                <CONJUNCTION>AND</CONJUNCTION>
                <OPERATOR>is blank</OPERATOR>
                <COLUMN>foo</COLUMN>
            </EXPRESSION>
        </RT_EXPRESSIONS>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type=”behavior”>
        <OPERATOR>has been sent email</OPERATOR>
        <ID>2</ID>
    </EXPRESSION>
</CRITERIA>
<CRITERIA>
    <EXPRESSION criteria_type=”profile”>
        <OPERATOR>starts with one of the following</OPERATOR>
```

```
							<COLUMN>foo</COLUMN>
							<VALUES>
								<VALUE><![CDATA[[columnA]]]></VALUE>
								<VALUE><![CDATA[[211olumn]]]></VALUE>
								<VALUE><![CDATA[[211olumn]]]></VALUE>
							</VALUES>
						</EXPRESSION>
					</CRITERIA>
					<CRITERIA>
						<EXPRESSION criteria_type="profile">
							<OPERATOR>starts with one of the following</OPERATOR>
							<COLUMN>foo</COLUMN>
							<VALUES>
								<VALUE><![CDATA[a]]></VALUE>
								<VALUE><![CDATA[b]]></VALUE>
								<VALUE><![CDATA[c]]></VALUE>
								<VALUE><![CDATA[abc,efg]]></VALUE>
							</VALUES>
						</EXPRESSION>
					</CRITERIA>
					<CRITERIA>
						<EXPRESSION criteria_type="profile">
							<OPERATOR>starts with</OPERATOR>
							<COLUMN>foo</COLUMN>
							<VALUE><![CDATA[abc]]></VALUE>
						</EXPRESSION>
					</CRITERIA>
					<CRITERIA>
						<EXPRESSION criteria_type="behavior">
							<OPERATOR>has submitted web form</OPERATOR>
							<WEB_FORM_NAME><![CDATA[my web form
name]]></WEB_FORM_NAME>
						</EXPRESSION>
					</CRITERIA>
					<CRITERIA>
						<EXPRESSION criteria_type="behavior">
							<OPERATOR>has visited web page</OPERATOR>
```

	``` <WEB_PAGE><![CDATA[http://www.xyz.com]]></WEB_PAGE> </EXPRESSION> </CRITERIA> <CRITERIA> <EXPRESSION criteria_type="behavior"> <OPERATOR>has visited web site</OPERATOR> <WEB_SITE><![CDATA[my web site name]]></WEB_SITE> </EXPRESSION> </CRITERIA> <CRITERIA> <EXPRESSION criteria_type="profile"> <OPERATOR>is within the last</OPERATOR> <COLUMN>foo</COLUMN> <VALUE><![CDATA[10]]></VALUE> <TIME_UNIT>days</TIME_UNIT> </EXPRESSION> </CRITERIA> <CRITERIA> <EXPRESSION criteria_type="profile"> <OPERATOR>is within the next</OPERATOR> <COLUMN>foo</COLUMN> <VALUE><![CDATA[10]]></VALUE> <TIME_UNIT>days</TIME_UNIT> </EXPRESSION> </CRITERIA> <CRITERIA> <EXPRESSION criteria_type="profile"> <OPERATOR>year is equal to</OPERATOR> <COLUMN>foo</COLUMN> <VALUE><![CDATA[2000]]></VALUE> </EXPRESSION> </CRITERIA> </RULE> </RULES> </RULESET> ```

Export a Dynamic Content Ruleset

ExportDCRuleset

This interface exports the specified Ruleset and all associated content (including images), from the Asset Library as a .zip file. Engage places the file in the FTP or Stored Files directory based on the user’s choice (Stored Files is the default directory if user does not select a specific location).

This interface supports both Classic and New Dynamic Content Rulesets. The exported file will be appropriately formatted (Classic or New Dynamic Content Ruleset) in accordance with RULESET_ID that was passed.

Operation	**<ExportDCRuleset>**		
Elements	RULESET_ID	Specifies the Ruleset ID to export. Supports both Classic and New DC Ruleset	
	MOVE_TO_FTP	Optional	Use the MOVE_TO_FTP parameter to retrieve the output file programmatically. If specified, Engage will move the files to the download directory of the user's FTP space. Omitting the MOVE_TO_FTP parameter will move exported files to the user's Export Files directory in Engage.
Example	<Envelope> <Body> <ExportDCRuleset> <RULESET_ID>12604</RULESET_ID> <MOVE_TO_FTP>TRUE</MOVE_TO_FTP> </ExportDCRuleset> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS	"True" indicates successful Data Job submission. "False" indicates an error occurred in which the error code displayed with a message reporting the issue.	
	Job ID	Contains the Data Job ID for the export.	
	File name	Name of the resulting file containing the exported Ruleset.	

Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <JOB_ID>28217</JOB_ID> <FILE_NAME>download/RS1 - Feb 23 2011 11-17-09 AM.XML</FILE_NAME> </RESULT> </Body> </Envelope>

List Dynamic Content Rulesets for a Mailing

ListDCRulesetsForMailing

This interface returns a list of available DC Rulesets for the specified mailing.

Operation	**<ListDCRulesetsForMailing>**		
Elements	MAILING_ID		The ID of the mailing whose Rulesets are being retrieved. This may be the ID of a mailing template or a sent mailing, in addition to an Autoresponder or Automated Message.
Example	<Envelope> <Body> <ListDCRulesetsForMailing> <MAILING_ID>31986</MAILING_ID> </ListDCRulesetsForMailing> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS		"True" indicates successful listing call. "False" indicates an error occurred in which the error code displays with a message reporting the issue.
Child Element	RULESET		
	Child Element	RULESET_ID	The ID of the associated Ruleset.
		LIST_ID	The ID of the associated Database
		RULESET_NAME	The name of the associated Ruleset.

Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <RULESET> <RULESET_ID>3762</RULESET_ID> <LIST_ID>11015</LIST_ID> <RULESET_NAME>Save-A-Pile Ruleset</RULESET_NAME> </RULESET> <RULESET> <RULESET_ID>3922</RULESET_ID> <LIST_ID>11015</LIST_ID> <RULESET_NAME>Ruleset for DC</RULESET_NAME> </RULESET> </RESULT> </Body> </Envelope>

Get Dynamic Content Ruleset Details

GetDCRuleset

8.7 Note: Current Limitations on Dynamic Content vs. "Classic Dynamic Content:

In an effort to provide significant functional improvements to our customers, we have been adding new capabilities as well as a fresh UI to the Dynamic Content in Engage. While this new functionality has been successfully delivered to the Engage UI, we have not yet updated our APIs to be compatible with these "new" Dynamic Content Rulesets. As a result, any API reference to a "new" Dynamic Content block or Ruleset should be avoided until upcoming product releases where we will begin to support all API methods for both "new" Dynamic Content and Classic Dynamic Content. All Classic Dynamic Content and Rulesets still work as expected with our APIs.

This interface returns the specified Ruleset in XML format. It does not support returning content images. To retrieve a complete Ruleset and its associated content, use the ExportDCRuleset operation.

Operation	**<GetDCRuleset>**		
Elements	RULESET_ID		The ID of the Ruleset to retrieve.
Example	<Envelope> <Body> <GetDCRuleset> <RULESET_ID>123456789</RULESET_ID> </GetDCRuleset> </Body> </Envelope>		
Response	**<RESULT>**		
Elements	SUCCESS		"True" indicates a successful "Get Ruleset" call. "False" indicates an error occurred in which the error code displays with a message reporting the issue.
	RULESET		
	Child Elements	RULESET_ID	The ID of the requested Ruleset.
		RULESET_NAME	The name of the Ruleset.
		CONTENT_AREAS	See AddDCRuleset for details
		RULES	See AddDCRuleset for details

Example	``` <Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <RULESET> <RULESET_ID>42579</RULESET_ID> <RULESET_NAME>Promo By Gender</RULESET_NAME> <CONTENT_AREAS> <CONTENT_AREA name="dc_mailing_subject" type="Subject"> <DEFAULT_CONTENT name="Default.dc_mailing_subject"><![CDATA[All clothing on sale this week.]]></DEFAULT_CONTENT> </CONTENT_AREA> <CONTENT_AREA name="dc_promo" type="Body-HTML"> <DEFAULT_CONTENT name="Default.dc_promo"><![CDATA[10% Off Everything]]></DEFAULT_CONTENT> </CONTENT_AREA> </CONTENT_AREAS> <RULES> <RULE> <RULE_NAME>MalePromotion</RULE_NAME> <PRIORITY>1</PRIORITY> <CRITERIA> <EXPRESSION> <AND_OR/> <LEFT_PARENS>(</LEFT_PARENS> <TYPE>TE</TYPE> <COLUMN_NAME>Gender</COLUMN_NAME> <OPERATORS><![CDATA[=]]></OPERATORS> <VALUES><![CDATA[M]]></VALUES> <RIGHT_PARENS>)</RIGHT_PARENS> </EXPRESSION> <EXPRESSION> <AND_OR>AND</AND_OR> <LEFT_PARENS>(</LEFT_PARENS> <TYPE>NE</TYPE> <COLUMN_NAME>Salary</COLUMN_NAME> <OPERATORS><![CDATA[>]]></OPERATORS> <VALUES><![CDATA[90000]]></VALUES> <RIGHT_PARENS>)</RIGHT_PARENS> </EXPRESSION> </CRITERIA> <CONTENTS> <CONTENT name="male_promo" content_area="dc_promo"><![CDATA[10% Off All Men's Clothing]]></CONTENT> <CONTENT name="male_subject" content_area="dc_mailing_subject"><![CDATA[Men's clothing on sale this week.]]></CONTENT> </CONTENTS> </RULE> <RULE> <RULE_NAME>Female Promotion</RULE_NAME> <PRIORITY>2</PRIORITY> <CRITERIA> ```

```xml
                                                <EXPRESSION>
                                                        <AND_OR/>
                                                        <LEFT_PARENS>(</LEFT_PARENS>
                                                        <TYPE>TE</TYPE>
        <COLUMN_NAME>Gender</COLUMN_NAME>
        <OPERATORS><![CDATA[=]]></OPERATORS>
                                <VALUES><![CDATA[F]]></VALUES>
                                        <RIGHT_PARENS>)</RIGHT_PARENS>
                                                </EXPRESSION>
                                                <EXPRESSION>
                                                        <AND_OR>AND</AND_OR>
                                                        <LEFT_PARENS>(</LEFT_PARENS>
                                                        <TYPE>NE</TYPE>
        <COLUMN_NAME>Salary</COLUMN_NAME>
        <OPERATORS><![CDATA[>]]></OPERATORS>
        <VALUES><![CDATA[90000]]></VALUES>
                                                <RIGHT_PARENS>)</RIGHT_PARENS>
                                                </EXPRESSION>
                                        </CRITERIA>
                                        <CONTENTS>
                                                <CONTENT name="female_promo"
content_area="dc_promo"><![CDATA[10% Off All Women's Clothing]]></CONTENT>
                                                <CONTENT name="female_subject"
content_area="dc_mailing_subject"><![CDATA[Women's clothing on sale this week.]]></CONTENT>
                                        </CONTENTS>
                                </RULE>
                        </RULES>
                </RULESET>
            </RESULT>
        </Body>
</Envelope>
```

Replace a Dynamic Content Ruleset

ReplaceDCRuleset

8.7 Note: Current Limitations on Dynamic Content vs. "Classic Dynamic Content:

In an effort to provide significant functional improvements to our customers, we have been adding new capabilities as well as a fresh UI to the Dynamic Content in Engage. While this new functionality has been successfully delivered to the Engage UI, we have not yet updated our APIs to be compatible with these "new" Dynamic Content Rulesets. As a result, any API reference to a "new" Dynamic Content block or Ruleset should be avoided until upcoming product releases where we will begin to support all API methods for both "new" Dynamic Content and Classic Dynamic Content. All Classic Dynamic Content and Rulesets still work as expected with our APIs.

This interface replaces an existing Ruleset with newly specified parameters. **Note:** A ruleset cannot contain more than 1,000 rules.

Operation	**<ReplaceDCRuleset>**	
Elements	RULESET_ID	The ID of the Ruleset to replace.
	CONTENT_AREAS	See AddDCRuleset for details
	RULES	See AddDCRuleset for details
Example	<Envelope> <Body> <ReplaceDCRuleset> <RULESET_ID>8309</RULESET_ID> <CONTENT_AREAS> <CONTENT_AREA name="dc_mailing_subject" type="Subject"> <DEFAULT_CONTENT name="Default.dc_mailing_subject"><![CDATA[All clothing on sale this week.]]></DEFAULT_CONTENT> </CONTENT_AREA> <CONTENT_AREA name="dc_promo" type="Body-HTML"> <DEFAULT_CONTENT name="Default.dc_promo"><![CDATA[10% Off Everything]]></DEFAULT_CONTENT> </CONTENT_AREA> </CONTENT_AREAS> <RULES> <RULE> <RULE_NAME>MalePromotion</RULE_NAME> <PRIORITY>1</PRIORITY> <CRITERIA> <EXPRESSION> <AND_OR/> <LEFT_PARENS>(</LEFT_PARENS> <TYPE>TE</TYPE> <COLUMN_NAME>Gender</COLUMN_NAME> <OPERATORS><![CDATA[=]]></OPERATORS>	

```
                    <VALUES><![CDATA[M]]></VALUES>
                    <RIGHT_PARENS>)</RIGHT_PARENS>
                </EXPRESSION>
                <EXPRESSION>
                    <AND_OR>AND</AND_OR>
                    <LEFT_PARENS>(</LEFT_PARENS>
                    <TYPE>NE</TYPE>
                    <COLUMN_NAME>Salary</COLUMN_NAME>
                    <OPERATORS><![CDATA[>]]></OPERATORS>
                    <VALUES><![CDATA[90000]]></VALUES>
                    <RIGHT_PARENS>)</RIGHT_PARENS>
                </EXPRESSION>
            </CRITERIA>
            <CONTENTS>
                <CONTENT name="male_subject"
content_area="dc_mailing_subject"><![CDATA[Men's clothing on sale this week.]]></CONTENT>
                <CONTENT name="male_promo"
content_area="dc_promo"><![CDATA[10% Off All Men's Clothing]]></CONTENT>
            </CONTENTS>
        </RULE>
        <RULE>
            <RULE_NAME>Female Promotion</RULE_NAME>
            <PRIORITY>2</PRIORITY>
            <CRITERIA>
                <EXPRESSION>
                    <AND_OR/>
                    <LEFT_PARENS>(</LEFT_PARENS>
                    <TYPE>TE</TYPE>
                    <COLUMN_NAME>Gender</COLUMN_NAME>
                    <OPERATORS><![CDATA[=]]></OPERATORS>
                    <VALUES><![CDATA[F]]></VALUES>
                    <RIGHT_PARENS>)</RIGHT_PARENS>
                </EXPRESSION>
                <EXPRESSION>
                    <AND_OR>AND</AND_OR>
                    <LEFT_PARENS>(</LEFT_PARENS>
                    <TYPE>NE</TYPE>
                    <COLUMN_NAME>Salary</COLUMN_NAME>
                    <OPERATORS><![CDATA[>]]></OPERATORS>
                    <VALUES><![CDATA[90000]]></VALUES>
                    <RIGHT_PARENS>)</RIGHT_PARENS>
                </EXPRESSION>
            </CRITERIA>
            <CONTENTS>
                <CONTENT name="female_subject"
content_area="dc_mailing_subject"><![CDATA[Women's clothing on sale this week.]]></CONTENT>
                <CONTENT name="female_promo"
content_area="dc_promo"><![CDATA[10% Off All Women's Clothing]]></CONTENT>
```

	</CONTENTS> </RULE> </RULES> </ReplaceDCRuleset> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	"True" indicates successful rule creation. "False" indicates an error occurred and the error code displays with a message reporting the issue.
	RULESET_ID	Contains the new Ruleset ID.
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <RULESET_ID>28217</RULESET_ID> </RESULT> </Body> </Envelope>	

Validate a Dynamic Content Ruleset

ValidateDCRuleset

8.7 Note: Current Limitations on Dynamic Content vs. "Classic Dynamic Content:

In an effort to provide significant functional improvements to our customers, we have been adding new capabilities as well as a fresh UI to the Dynamic Content in Engage. While this new functionality has been successfully delivered to the Engage UI, we have not yet updated our APIs to be compatible with these "new" Dynamic Content Rulesets. As a result, any API reference to a "new" Dynamic Content block or Ruleset should be avoided until upcoming product releases where we will begin to support all API methods for both "new" Dynamic Content and Classic Dynamic Content. All Classic Dynamic Content and Rulesets still work as expected with our APIs.

This interface performs a validation check against all Dynamic Content Rulesets (including nested Rulesets) associated with the specified mailing. The validation check ensures that all placeholders and criteria match the specified mailing and its Rulesets.

Operation	**<ValidateDCRuleset>**	
Elements	MAILING_ID	Mailing ID for a mailing with one or more associated Rulesets.
Example	<Envelope> <Body> <ValidateDCMailingRuleset> <MAILING_ID>31986</MAILING_ID> </ValidateDCMailingRuleset> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	"True" indicates the mailing and all related Dynamic Content is valid. "False" indicates that Engage will display a collection of errors.
	MAILING_ID	Represents the mailing against which Engage performed a validation.
	ERRORS	Displayed when the SUCCESS parameter is set to "False." Represents the collection of errors resulting from the validation.

Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <MAILING_ID>8309</MAILING_ID> </RESULT> </Body> </Envelope>

Delete a Dynamic Content Ruleset

DeleteDCRuleset

This interface deletes the specified Ruleset from the Asset Library. You can only delete Rulesets that are not associated with a mailing. If you attempt to delete a mailing's Ruleset, you will receive an error indicating the reason for failure (the Ruleset is currently associated with a mailing). If the desired result is to delete the Ruleset in order to update its content, utilize the ReplaceDCRuleset operation.

Operation	**<DeleteDCRuleset>**	
Elements	RULESET_ID	The ID of the Ruleset to delete.
Example	<Envelope> <Body> <DeleteDCRuleset> <RULESET_ID>123</RULESET_ID> </DeleteDCRuleset> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	"True" indicates successful Ruleset deletion. "False" indicates an error occurred and the error code displays with a message reporting the issue.
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> </RESULT> </Body> </Envelope>	

Get a List of Mailing Templates

GetMailingTemplates

This interface extracts a list of Shared or Private mailing templates for your Organization. The templates returned may be limited by the date they were last modified.

Operation	**<GetMailingTemplates>**		
Elements	VISIBILITY	Defines the visibility of the mailing templates to return. 0 – Private 1 – Shared	
	IS_CRM_ENABLED	Optional	When set to "true", the API only provides a list of mailing templates that are enabled for use within a CRM package. This allows users to create mailing templates that are built specifically for sending email from within a CRM product, like Salesforce.com or Microsoft Dynamics.
	LAST_MODIFIED_START_DATE	Optional	Start Date used to filter mailing templates by the date and time the mailing template was last modified.
	LAST_MODIFIED_END_DATE	Optional	End Date used to filter mailing templates by the date and time the mailing template was last modified.

Example

```
<Envelope>
    <Body>
        <GetMailingTemplates>
            <VISIBILITY>0</VISIBILITY>
            <IS_CRM_ENABLED>true</IS_CRM_ENABLED>
            <LAST_MODIFIED_START_DATE>01/01/2011 13:00:00</LAST_MODIFIED_START_DATE>
            <LAST_MODIFIED_END_DATE>01/31/2011 12:59:59</LAST_MODIFIED_END_DATE>
        </GetMailingTemplates>
    </Body>
</Envelope>
```

Response	**<RESULT>**		
Elements	SUCCESS	**True** if successful.	
	MAILING_TEMPLATE	XML nodes defining the instances of each mailing template.	
	Child Element	MAILING_ID	Returns the ID of the mailing template.
		MAILING_NAME	Returns the name of the mailing template.
		SUBJECT	The subject line of the mailing template.
		LAST_MODIFIED	Returns the date and time when the mailing template was last modified.
		VISIBILITY	Returns the visibility of the mailing template (Shared or Private).
		USER_ID	Returns the User ID for the mailing template owner.
		ALLOW_CRM_BLOCK	Identifies that a CRM Block has been inserted into the body of the template and is available for use. This element should only be included in the response for a CRM List.

Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <MAILING_TEMPLATE> <MAILING_ID>365333</ MAILING_ID> <MAILING_NAME>Mailing One</ MAILING_NAME> <SUBJECT>Mailing One</ SUBJECT> <LAST_MODIFIED>08/23/07 04:18 PM</LAST_MODIFIED> <VISIBILITY>0</VISIBILITY> <USER_ID>8c3747-111fae2b32c21fbca0cb8d6</USER_ID> <ALLOW_CRM_BLOCK>true</ALLOW_CRM_BLOCK> </MAILING_TEMPLATE > <MAILING_TEMPLATE > <MAILING_ID>323543</ MAILING_ID> <MAILING_NAME>Mailing Two</ MAILING_NAME> <SUBJECT>Mailing Two</ SUBJECT> <LAST_MODIFIED>09/26/07 10:31 AM</LAST_MODIFIED> <VISIBILITY>0</VISIBILITY> <USER_ID>8c3747-111fae23972-f520cb8d6</USER_ID> <ALLOW_CRM_BLOCK>false</ALLOW_CRM_BLOCK> </MAILING_TEMPLATE > </RESULT> </Body> </Envelope>

Note: When a mailing template is marked as a CRM template, the creator can enable a CRM Block that can be inserted into the body of the template (%%SP_CRM_BLOCK%%) When one of the CRMSendMailing APIs is called, the content for the %%SP_CRM_BLOCK%% is specified in the XML envelope and is substituted before the mailing(s) are sent.

If CRM_BLOCK is set to false then this option was not enabled in the template and the CRM integration should not allow the user to enter notes into the mailing.

Export a Mailing Template

ExportMailingTemplate

This interface exports a mailing template. The results are an .stl file (a zip file containing XML with a Silverpop Template extension) which will be written to the FTP account associated with the current session. You can (optionally) copy the resulting file to Stored Files.

Operation	**<ExportMailingTemplate>**		
Elements	TEMPLATE_ID	Numeric identifier for the mailing template to be exported.	
	ADD_TO_STORED_FILES	Optional	Use the ADD_TO_STORED_FILES parameter to write the output to the stored files folder within Engage. Omitting the ADD_TO_STORED_FILES parameter will cause export files to be moved to the user's FTP space, under the download directory.

Example	<Envelope> <Body> <ExportMailingTemplate> <TEMPLATE_ID>59294</TEMPLATE_ID> <ADD_TO_STORED_FILES/> </ExportMailingTemplate> </Body> </Envelope>	
Response	**<RESULT>**	
Elements	SUCCESS	TRUE if successful
	FILE_PATH	Returns the file path and name of the export file
Example	<Envelope> <Body> <RESULT> <SUCCESS>TRUE</SUCCESS> <FILE_PATH>/download/my_mailing_template.stl</FILE_PATH> </RESULT> </Body> </Envelope>	

Appendices

Frequently Asked Questions: Non Email Key (NEK) Databases

Q: Does OptOutRecipient opt out all contacts with the email address or just the contact who matches the unique key?

A: You can use the OptOutRecipient API on an NEK database by including all unique key columns with their corresponding name/value pairs or by specifying the email address. If you do specify a unique key, Engage only opts out the contact who matches the unique key, not other contacts with the same email address. If you only specify the email address, the system opts out all contacts with the specified email address.

Q: Can you change the unique key in an existing database?

A: No, once you define the database, you can only add unique keys.

Q: Do you have to use the email address along with the unique key when calling SelectRecipientData?

A: No, you can leave out the email address and just provide the unique key.

Q: If you have multiple unique keys and call AddRecipient, do you have to provide the email address and all unique keys?

A: Yes. You must provide all unique keys **and** the email address when calling the AddRecipient operation.

Q: Should my master Suppression list have the same key as my databases?

A: No, your master Suppression list should have Email as its key. Add to Master Suppression List uses email address for opting out (without a unique key, other than the email specified in the master Suppression list.)

Q: Does the RemoveRecipient operation require that you specify all unique keys **and** the email address to remove a contact?

A: Yes. If you have more than one instance of the same email address in the database, specifying all unique keys prevents removal of records with the same email address.

Q: When creating a database using ImportList, do you have to specify the unique keys in the parameters?

A: Yes. It won't automatically use the default unique key set for the Org.

Q: Are unique keys always required even if you set <IS_REQUIRED> to False?

A: Yes. For unique keys, Engage ignores a False value for the <IS_REQUIRED> element.

Q: When using the UpdateRecipient operation, can you update the email address for a contact when email is not the primary key?

A: Yes. If there are two contacts with the same email address with different unique keys, you can update the email address using the unique key and it will change the email address just for the contact who matches the unique key.

Code Samples

PHP

The following example uses a PHP file to log into Engage and fetch information about contact sources using the API. To use this code, you must host it on a Web server and execute it with a PHP engine running on that server.

```
<?php

require_once __DIR__ . '/lib/Engage.php';

/**********************************************************************/

// Configuration section - replace with your values before running

// replace with your Engage API endpoint hostname:

$apiHost = 'api1.silverpop.com';


// replace with your Engage API username:

$username = 'user@host.com';


// replace with your Engage API password:

$password = 'xxxyz123';

/**********************************************************************/

try {


    echo "Logging into Engage API on {$apiHost} as {$username}\n";


    $engage = new Engage($apiHost);

    $engage->login($username, $password);


    echo "Fetching list of shared databases, queries, and contact lists\n";


    $request = '<GetLists><VISIBILITY>1</VISIBILITY><LIST_TYPE>2</LIST_TYPE></GetLists>';
```

```
    $response = $engage->execute($request);


    foreach ($response->RESULT->LIST as $list) {


        if ($list->TYPE == 0) {
            $type = 'Database';
        } elseif ($list->TYPE == 1) {
            $type = 'Query';
        } elseif ($list->TYPE == 18) {
            $type = 'Contact list';
        } else {
            $type = 'Unknown list type';
        }


        $id = $list->ID;
        $name = $list->NAME;
        $size = $list->SIZE;


        echo "{$type} ID {$list->ID} '{$list->NAME}' has {$list->SIZE} recipients\n";
    }


} catch (Exception $e) {
    echo $e->getMessage() . "\n";
    print_r($engage->getLastRequest());
    print_r($engage->getLastResponse());
    print_r($engage->getLastFault());
}
```

```
//PHP REQUIRED CLASS Engage.php in subdirectory /lib

<?php



class Engage
{
    protected $apiHost = null;

    protected $username = null;
    protected $password = null;

    protected $sessionId = null;

    protected $lastRequest = null;
    protected $lastResponse = null;
    protected $lastFault = null;

    public function __construct($apiHost)
    {
        $this->apiHost = $apiHost;
    }

    public function execute($request)
    {
        if ($request instanceof SimpleXMLElement) {
            $requestXml = $request->asXML();
```

```
} else {
    $requestXml = "<?xml version=\"1.0\"?>\n<Envelope><Body>{$request}</Body></Envelope>";
}

// NOTE: Make sure that your request string uses UTF-8 encoding

$this->lastRequest = $requestXml;
$this->lastResponse = null;
$this->lastFault = null;

$curl = curl_init();

curl_setopt($curl, CURLOPT_URL, $this->getApiUrl());
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, $requestXml);
curl_setopt($curl, CURLINFO_HEADER_OUT, true);
curl_setopt($curl, CURLOPT_HTTPHEADER, array('Content-Type: text/xml; charset=UTF-8', 'Content-Length: ' . strlen($requestXml)));
curl_setopt($curl, CURLOPT_CONNECTTIMEOUT, 60);
curl_setopt($curl, CURLOPT_TIMEOUT, 180);

$responseXml = @curl_exec($curl);

if ($responseXml === false) {
    throw new Exception('CURL error: ' . curl_error($curl));
}

curl_close($curl);
```

```
        if ($responseXml === true || !trim($responseXml)) {
            throw new Exception('Empty response from Engage');
        }

        $this->lastResponse = $responseXml;

        // NOTE: You may want to check that the Engage response is in valid UTF-8 encoding before parsing the XML

        $response = @simplexml_load_string('<?xml version="1.0"?>' . $responseXml);

        if ($response === false) {
            throw new Exception('Invalid XML response from Engage');
        }

        if (!isset($response->Body)) {
            throw new Exception('Engage response contains no Body');
        }

        $response = $response->Body;

        $this->checkResult($response);

        return $response;
    }

    public function getApiUrl()
    {
```

```
        $url = "https://{$this->apiHost}/XMLAPI";

        if ($this->sessionId !== null) {
            $url .= ';jsessionid=' . urlencode($this->sessionId);
        }

        return $url;
    }

    public function checkResult($xml)
    {
        if (!isset($xml->RESULT)) {
            throw new Exception('Engage XML response body does not contain RESULT');
        }

        if (!isset($xml->RESULT->SUCCESS)) {
            throw new Exception('Engage XML response body does not contain RESULT/SUCCESS');
        }

        $success = strtoupper($xml->RESULT->SUCCESS);

        if (in_array($success, array('TRUE', 'SUCCESS'))) {
            return true;
        }

        if ($xml->Fault) {
            $this->lastFault = $xml->Fault;
            $code = (string)$xml->Fault->FaultCode;
```

```
        $error = (string)$xml->Fault->FaultString;
        throw new Exception("Engage fault '{$error}'" . ($code ? "(code: {$code})" : ''));
    }

    throw new Exception('Unrecognized Engage API response');
}

public function getLastRequest()
{
    return $this->lastRequest;
}

public function getLastResponse()
{
    return $this->lastResponse;
}

public function getLastFault()
{
    return $this->lastFault;
}

public function login($username, $password)
{
    $this->username = $username;
    $this->password = $password;
    $this->sessionId = null;
```

```
        $request =
"<Login><USERNAME><![CDATA[{$username}]]></USERNAME><PASSWORD><![CDATA[{$password}]]></PASSWORD></Login>";


        try {
            $response = $this->execute($request);
        } catch (Exception $e) {
            throw new Exception('Login failed: ' . $e->getMessage());
        }


        if (!isset($response->RESULT->SESSIONID)) {
            throw new Exception('Login response did not include SESSIONID');
        }


        $this->sessionId = $response->RESULT->SESSIONID;
    }
}
```

Java

The following example uses Java to call the AddRecipient XML API.

```
String request = "<Envelope><Body></Body></Envelope>"; // put the actual XML here.
  String sendEncoding = "utf-8";
  HttpURLConnection urlConn = null;
  OutputStream out = null;
  InputStream in = null;
  try {
   URL url = new URL(apiUrl);
   urlConn = (HttpURLConnection)url.openConnection();
   urlConn.setRequestMethod("POST");
   urlConn.setDoOutput(true);
   urlConn.setRequestProperty("Content-Type","text/xml;charset=" + sendEncoding);
   urlConn.connect();
   out = urlConn.getOutputStream();
   out.write(request.getBytes(sendEncoding));
   out.flush();
   in = urlConn.getInputStream();
   InputStreamReader inReader = new InputStreamReader(in, sendEncoding);
   StringBuffer responseBuffer = new StringBuffer();
   char[] buffer = new char[BUFFER_SIZE];
   int bytes;
   while ((bytes = inReader.read(buffer)) != -1) {
    responseBuffer.append(buffer, 0, bytes);
   }
   response = responseBuffer.toString();
  } finally {
   if (out != null) {
    try {out.close();} catch (Exception e) {}
   }
   if (in != null) {
    try {in.close();} catch (Exception e) {}
   }
   if (urlConn != null) {
    urlConn.disconnect();
   }
  }
```

API Error Codes

The following are possible error codes in the XML API. In most cases, the error description found in the Fault String element gives more detail on the piece of data causing the error for the particular API.

Error Number	Description
1	FTF_NON_NUMERIC_MAILING_KEY
2	FTF_NON_NUMERIC_SENDER_KEY
3	FTF_BAD_MAILING
4	FTF_INVALID_EMAIL_ADDRESS
5	FTF_INVALID_ENCRYPTED_SENDER_KEY
6	FTF_INVALID_COMMENT_SIZE
50	Server Error (typically returned if the API is invoked incorrectly such as no XML passed in the request)
51	Invalid XML Request
52	Missing XML parameter
100	Parameter "x" was not provided in API call
101	Name already in use. Engage cannot rename the template directory.
102	Directory already exists.
103	Parent directory does not exist.
104	Visibility is not valid.
105	List type is not valid.
106	List ID is not valid.
107	Mailing ID is not valid.
108	Tracking Level is not valid.
109	Error saving mailing to the database.
110	Retain flag is not valid.
111	Mailing Type is not valid.
112	Click Through Type is not valid.
113	TextSize is not an integer.
114	Parameter "x" was not provided in API call
115	Name already in use. Engage cannot rename template directory.
116	ERR_INVALID_CREATED_FROM
117	ERR_INVALID_ALLOW_HTML
118	ERR_INVALID_SEND_AUTOREPLY

Error Number	Description
119	ERR_INVALID_UPDATE_IF_FOUND
120	Error saving recipient to the database.
121	Unable to add recipient. No EMAIL provided.
122	Unable to add recipient. Recipient already exists.
123	Unable to update recipient / recipient does not exist.
124	Recipient ID is not valid.
125	No List ID or Mailing ID provided with the Recipient ID.
126	Mailing does not exist.
127	Mailing deleted.
128	Recipient is not a member of the list.
129	Recipient has opted out of the list.
130	Unable to send mailing; Internal error.
131	ERR_INVALID_IMPORT_TYPE
132	Unable to create import job.
133	File type is not valid.
134	File type is not valid.
135	Job ID is not valid.
136	Unable to create Delete job. Internal error.
137	Unable to destroy mailing. Internal error.
138	Unable to remove recipient from list. Internal error.
139	Unable to create DC ruleset export job.
140	Editor type is not valid.
141	Encoding is not valid.
143	List is a query, cannot delete list query recipients.
145	Session has expired or is invalid.
146	Invalid default value for List Column type
147	Include All Lists is not valid.
150	Organization permissions prohibit using this API.
151	ERR_LIST_META_DENIED
152	Unable to create set column values job. Internal error.
153	ERR_EXPORT_NOT_LIST_COLUMN
154	Action code is not valid.

Error Number	Description
155	Ruleset 'x' does not exist
156	Unable to create Export job. Internal error.
160	Can only send Custom Automated Mailings. Please provide the Mailing ID for a Custom Automated Mailing.
161	COLUMN_NAME is not valid for this list.
162	Mailing is not active.
170	SQLException deleting ruleset.
171	Error deleting rule.
172	Usage was not an integer.
173	SQLException listing Dynamic Content ruleset.
174	SQLException listing Dynamic Content rulesets for list.
180	Unable to check if user exists. Internal error.
181	You cannot schedule Multimatch Mailings through the API.
182	A Mailing with the provided name already exists.
183	Errors found validating mailing.
184	Numerous errors related to dates.
185	Report ID for Behavior is invalid.
186	ERR_INVALID_SENT_MAILING_TYPE
187	RECURSIVE flag is not valid.
188	Cannot use a System field name for a List column.
190	Unable to locate element in the definition. Unable to continue.
211	Unable to complete request. Required unique key column values were not provided
256	Unable to create query. New List name already exists.
300	A Ruleset with the provided name already exists.
301	A Ruleset with the provided name does not exist.
310	Invalid value for Element: LIST_ID. Not an integer. Value: 'x'
311	Unable to create Recipient Data Job. Internal error.
312	List is not the right type for this API.
313	Column is not the right type for this API.
314	Column 'x' not found in list.
315	Unable to opt out recipient from list. Internal error.
316	Invalid XML in request: COLUMN Element found without a NAME.
320	Both MAILING_ID and LIST_ID provided. Please pick only one.

Error Number	Description
321	Export Format is not valid.
322	Mailing content archived.
323	Specified folder ID must be a number.
324	Visibility of the list and parent folder must match.
325	Specified folder ID does not exist.
326	Unable to update recipient's EMAIL. EMAIL is part of Unique Identifier
329	SYNC_FIELD Element found without a NAME.
500	Detailed report data for this mailing is not available at this time. Please try again later.
600	Error saving query to the database.
806	Autoresponder is not active
1001	Recipient Id Not Found in List
1002	Sync Id Not Found in List
1003	Error Saving Recipient
1004	Error Retrieving Recipient
1005	Error Adding Recipient
1006	Missing Recipient Key Info
1007	Column Does Not Exist in List
1008	Recipient Not Found in List
1009	Recipient Already Deleted in Engage
1011	Cannot Update System Column
1015	Error Merging Recipient
1016	Missing Required Column Value
1017	Email Address is Invalid
1018	Error Adding Contact; Contact Already Exists in Contact List

Sample Error Response


```
<Envelope>
	<Body>
		<RESULT>
			<SUCCESS>false</SUCCESS>
		</RESULT>
		<Fault>
			<Request/>
			<FaultCode/>
			<FaultString><![CDATA[Unable to remove the recipient. The list is private and you are not the owner.]]></FaultString>
			<detail>
				<error>
					<errorid>140</errorid>
					<module/>
					<class>SP.Recipients</class>
					<method/>
				</error>
			</detail>
		</Fault>
	</Body>
</Envelope>
```