

Last modified February 3, 2011 | <http://www.epsilon.com>

DREAMmail Programming Guide

Version 4.9

Copyrights

This documentation and related technology are governed by a user agreement and shall remain the sole and exclusive property of Epsilon. No part of this documentation or related technology may be used, reproduced, translated, displayed, distributed, disclosed, stored in a retrieval system or transmitted in any form or by any means without the written permission of Epsilon, unless otherwise stated in the user agreement. The information contained in this documentation is confidential and proprietary to Epsilon.

Disclaimer

Epsilon does not warrant, guarantee or make any representations or otherwise concerning the contents of this documentation or the applicability thereof. Epsilon reserves the right to change the contents of this document at any time without prior notification of such updates.

Trademarks

DREAMmail is a registered service mark of Epsilon.

TABLE OF CONTENTS

Update History.....	vi
Preface to the <i>DREAMmail</i> Programming Guide	vii
Introduction to the Programming Guide	1
Writing <i>DMConnect</i> Requests.....	2
Review of XML Document Structure	3
Special Characters and Non-English Text.....	4
Language Support and Character Sets	7
Recommendations for Writing Requests	11
Implementing <i>DMConnect</i>	13
Overview of the Implementing <i>DMConnect</i>	14
Using <i>DMConnect</i> With Other Products	15
Syntax of <i>DMConnect</i> Requests and Responses	16
<i>DMConnect</i> Requests, Organized by Type.....	19
Responses Returned by <i>DMConnect</i> Requests.....	22
Summary of <i>DMConnect</i> Requests and Responses	23
Parameters Accepted in <i>DMConnect</i> Report Requests	26
FromDate and EndDate Parameters.....	27
AddProfileFields Request	28
ChangeRTMMailingStatus.....	31
ClearList Request	32
CreateCampaign Request	34
CreateConditionalContent Request	37
CreateContent Request	43
CreateDuplicateMailing Request	46
CreateList Request	48
CreateMailing Request	50
CreateMailingFromTemplate Request	61

CreateRTMMailing.....	67
ExportReport Request.....	68
GetAllExportNames Request.....	70
GetExportedReport Request.....	72
GetExportReportStatus Request.....	74
GetFtplImportStatus Request.....	76
GetFtplImportStatusRs Response.....	78
GetHygieneRuleNames Request.....	80
GetHygieneRuleNamesRs Response.....	81
GetJobStatus Request.....	82
GetListNames Request.....	84
GetListNamesRs Response.....	86
GetMailingContent Request.....	89
GetMailingContentRs Response.....	90
GetMailingStatus Request.....	92
GetMailingStatusRs Response.....	94
GetMessageSize Request.....	98
GetMessageSizeRs Response.....	100
GetObjectNames Request.....	102
GetObjectNamesRs Response.....	105
GetProfileFields Request.....	109
GetProfileFieldsRs Response.....	111
GetTemplateVariables Request.....	115
GetTemplateVariablesRs Response.....	117
GetTestMailingStatus Request.....	118
GetTestMailingStatusRs Response.....	120
GetTrackedURL Request.....	123
GetTrackedURLRs Response.....	125
ImportList Request.....	127
ManageLists Request.....	130
SendMailingForApproval Request.....	132
StartMailing Request.....	134
TestMailing Request.....	136
TriggerFtplImport Request.....	138
TriggerMailingToEmailAddresses Request.....	143
TriggerMailingToList Request.....	149
Unsubscribe Request.....	154
UpdateConditionalContent Request.....	157
UpdateContent Request.....	163
UpdateMailing Request.....	167
UpdateProfile Request.....	179
UpdateRTMMailing.....	182
UpdateTrackedURL Request.....	183

Generating Standard Reports With <i>DMConnect</i>	186
DeliveryResponse Report Request.....	187
Bouncebacks Report Request.....	190
Clickthroughs Report Request.....	193
ListActivity Report Request.....	196
EmailAddressChanges Report Request	200
ExportList Report Request	202
ProfileChanges Report Request.....	205
RTMDelivered Report Request	208
RTMNotDelivered Report Request	209
RTMClicks Report Request.....	211
RTMHTMLOpens Report Request	213
RTMSpotlight Report Request.....	214
Unsubscription Report Request.....	216
Posting <i>DMConnect</i> Requests	219
Request Form Parameters	220
Sending Requests through HTTP Posts.....	225
Responses to <i>DMConnect</i> Requests	236
Notification Response Message.....	237
<i>DMConnect</i> Messages and Response Codes	238
GetJobStatus and GetExportReportStatus Responses ..	257
Using <i>Spotlight</i>.....	259
About <i>Spotlight</i>	260
Description of <i>Spotlight</i> Tags	264
Creating <i>Spotlight</i> Tags	267
Generating Random Numbers in <i>Spotlight</i> Tags	277
Appendix - ASCII Escape Codes	279
Index.....	282

UPDATE HISTORY

TABLE: Revisions to DMConnect

Version	Revision	Date
4.9.3	Addition of Referral From Address to <i>TriggerMailingToEmailAddress</i> and <i>TriggerMailingToList</i> requests.	2/27/2010
4.9.6	Addition of the <i>GetTrackedURL</i> , <i>GetTrackedURLRs</i> , and <i>UpdateTrackedURL</i> requests.	9/2010
4.9.7	Updated description of <i>TriggerMailingToEmailAddresses</i> <i>EventVariable</i> element.	12/2010
4.9.7	Added nesting syntax note for event variable element.	02/2011

PREFACE TO THE *DREAMmail* PROGRAMMING GUIDE

About this guide

This guide is part of the set of documents that describe *DREAMmail*. This guide is intended for users, such as webmasters and application programmers, who use add-on programming features to interact with *DREAMmail*. These programming features include: *DMConnect* and *Spotlight*.

- *DMConnect* remotely submit requests to *DREAMmail*. *DMConnect* is an XML-based programming interface that lets you integrate your existing customer and content management systems, and reports with *DREAMmail*. You can automatically create campaigns, create and update messages and content, send various types of messages (including Real-Time Messaging), check the status and size of messages and test messages, manipulate profile fields, initiate single imports, and manage your mailing lists. You can also generate and retrieve reports.
- *Spotlight* is a built-in feature of *DREAMmail* that helps you analyze the return on investment (ROI) from an email campaign. You can use *Spotlight* to measure ecommerce activities including the amount of revenue that is generated by a website, or to track the number of subscribers who performed specific activities such as filling out a registration form or requesting a brochure.

This guide assumes that you have knowledge of the terminology and concepts of *DREAMmail* and that you have intermediate to advanced HTML programming skills, with experience creating forms and the ability to implement client-side scripting (for example, JavaScript).

How this guide is organized

This guide is divided into the following chapters:

- [Introduction to the Programming Guide](#) provides a brief overview of each of the features in *DREAMmail* that require some programming.
- [Writing DMConnect Requests](#) provides recommendations for writing requests in *DMConnect*.
- [Implementing DMConnect](#) describes *DMConnect* and discusses implementation issues.
- [Syntax of DMConnect Requests and Responses](#) provides an overview of the requests accepted by *DMConnect*, defines each element needed to create your requests, and lists the parameters that can be used in requests. The chapter can be used as a reference section as you write XML script to be submitted to *DMConnect*.

- [Generating Standard Reports With DMConnect](#) provides an overview of the reports that can be created and retrieved using *DMConnect*.
- [Posting DMConnect Requests](#) explains how you send requests to *DREAMmail*.
- [Responses to DMConnect Requests](#) describes how *DMConnect* receives and processes requests, and describes the response messages that are returned.
- [Using Spotlight](#) explains how you can use this feature in *DREAMmail* to analyze the return on investment (ROI) from email campaigns and track ecommerce or other subscriber-initiated activities that result from email campaigns.
- The [Appendix - ASCII Escape Codes](#) provides a reference table that you can use to escape non-English and special characters that you include in message text.

INTRODUCTION TO THE PROGRAMMING GUIDE

In addition to using *DREAMmail* to create all of the elements you need to produce and send messages, you can use XML to perform these same tasks using the *DMConnect* feature. There is also a programming feature, *Spotlight*, that allow you to collect information about your subscribers and their activities.

What is *DMConnect*?

DMConnect is an XML-based API (application programming interface) that lets you programmatically create campaigns, create and update messages, content, and mailing lists; send various types of messages; check the status and size of messages and test messages; manipulate profile fields; initiate single imports; and create, generate, and retrieve reports.

What is *Spotlight*?

Spotlight is a built-in feature of *DREAMmail* that helps you analyze the return on investment (ROI) from an email campaign. You can use *Spotlight* to measure ecommerce activities including the amount of revenue that is generated by a website, or to track the number of subscribers who performed specific activities such as filling out a registration form or requesting a brochure. For more information, see [Using *Spotlight*](#).

WRITING *DMCONNECT* REQUESTS

DMConnect is an XML-based API (application programming interface). This section provides a brief review of XML that you use to create requests in *DMConnect*.

This section discusses the following topics:

- [Review of XML Document Structure](#)
- [Special Characters and Non-English Text](#)
- [Language Support and Character Sets](#)
- [Recommendations for Writing Requests.](#)

Review of XML Document Structure

XML (Extensible Mark-up Language) is a specification developed by the W3C that is used to enable the definition, validation, and interpretation of data between applications. XML is text-based and is similar to HTML in many ways. Like an HTML document, an XML document holds text annotated by tags. However, unlike an HTML document, an XML document can have an unlimited number of tags, each indicating not how something looks, but what it means.

This section discusses:

- [XML elements](#)
- [Guidelines for creating an XML document.](#)

XML elements

An XML document consists of a series of XML elements. Each element contains a start tag `<URL>` and an end tag `</URL>`. The information between the two tags is the content. The tags represent a piece of data, and in *DMConnect* the data can correspond to a field in a database table. An XML document may contain data from more than one database table.

Note: XML tags are case-sensitive. Thus, the following three tags are different: `<URL>`, `<Url>`, and `<url>`

Mandatory elements: An XML document must contain all mandatory elements. If you do not include these elements, your XML script is invalid.

Empty XML elements: XML has a short cut for an empty element. A single tag ending with `/>` signals that the element has no contents. The following two examples are equivalent:

```
<ListName></ListName>
<ListName/>
```

Guidelines for creating an XML document

There are several guidelines that must be followed when writing an XML document:

- All elements must have a start and an end tag.
- All elements must be cleanly nested. Do not overlap elements.
- All attribute values must be enclosed in quotation marks.
- Each document must have a unique first element, the root element.

Special Characters and Non-English Text

Not all characters can be included in a message without first being modified to allow *DREAMmail* to correctly deliver the message. When you include certain characters — special characters, accented, non-English, double-byte, and multi-byte — in the content of a message, there are some modifications that you must make. You must either put text with these characters in a CDATA section or you must **escape** them. Escaping characters is the process of hiding certain characters from the XML parser by replacing them with an escape sequence. You must make sure that **both** the unsafe characters in the XML script **and** in your HTML script are shielded.

This section discusses:

- [Types of characters in message content](#)
- [How to include special characters in XML](#)
- [How to include special characters in HTML and AOL format messages.](#)

Types of characters in message content

Content can include characters in four categories:

- those in US-ASCII format (ASCII codes 0 to 127), such as the letters *A* to *Z*
- special characters, which are punctuation marks, symbols, and accented characters. There are **safe** special characters and **unsafe** special characters. Both safe and unsafe special characters are contained in the US-ASCII character set (ASCII codes 0 to 127). Safe special characters can be left unescaped. However, unsafe special characters **must** be coded in escaped ASCII format in HTML and XML tags (see [How to include special characters in HTML and AOL format messages](#)).
- extended ASCII characters (accented and non-English letters that have ASCII codes 128 to 255)
- double-byte or multi-byte characters (usually Asian-language characters).

Any character that does not have a graphical representation in the US-ASCII character set (ASCII codes 0 to 127) requires special handling in both HTML and XML. You must replace these characters with character combinations that use the escaped ASCII format so that they are not parsed as HTML or XML; see [How to include special characters in XML](#), [How to include special characters in HTML and AOL format messages](#), and [Appendix - ASCII Escape Codes](#).

How to include special characters in XML

In XML, sections of text that contain special and reserved characters must be shielded, or escaped, from the attention of the parser. This escaping is important if an XML document contains reserved characters that have special meanings in XML. It is

important to note that HTML tags that occur within the body of a message or a content segment in the *DMConnect* request must also be escaped.

There are two ways to escape reserved characters from the XML parser:

- use an escape sequence to replace the reserved characters
- include the reserved characters as text in a CDATA section.

Escape sequences for reserved characters in XML: The characters shown in the table below are part of the syntax of XML and cause problems if placed within an XML document. Substitute the escape sequence shown here for these characters.

TABLE: XML reserved characters

Character	Escape sequence
<	<
&	&
>	>
"	"
'	'

For other special characters, see the explanation in [How to include special characters in HTML and AOL format messages](#)

Including message content in CDATA section in XML requests: A CDATA section can be used to mark elements or reserved characters to prevent them from being literally interpreted by XML. The CDATA section is especially useful for escaping markup language and scripting in HTML and AOL messages.

The syntax for CDATA sections in XML is `<![CDATA[]]>`. An example of a CDATA section is

```
<EmailAddress><![CDATA[shane.o'neil@Epsilon.com]]></EmailAddress>
```

Important: Always place all text, AOL, and HTML format messages, content segments, and email addresses in CDATA sections in case reserved characters are used. For examples of message text and content segments in CDATA sections, see [CreateMailing Request](#), and [CreateContent Request](#).

Note: A CDATA section cannot contain another CDATA section. If a CDATA section contains the characters `<![CDATA[or]]>`, your request fails. Also make sure there are no spaces or line breaks inside the `]]>` string.

How to include special characters in HTML and AOL format messages

Special characters in HTML can be safe, unsafe, or reserved. Special characters are the HTML reserved characters used for their reserved purposes as well as the following characters: \$ - _ . + ! * ' () , ". Safe characters can be used unescaped in HTML and AOL format messages. All other special characters are considered *unsafe* in HTML. That is, they are reserved for special meanings, and **must** be represented using the escaped ASCII format in HTML.

Safe characters: *Safe* characters are the alphanumeric characters A-Z, a-z and 0-9. Safe characters can be used unescaped.

Reserved characters: Certain characters have reserved purposes in HTML:

- The characters < and > are used as the delimiters around HTML tags.
- A quotation mark (") is used to delimit URLs.
- The character # is used to delimit a URL from a fragment/anchor identifier that might follow it.
- The character % is used for encoding other characters.

Unsafe characters: Some characters such as { } | ^ ~ [] and \ are considered *unsafe* in HTML and XML because when they are sent in URLs via the Web, gateways, and other transport agents, these characters are sometimes modified. When you want to include unsafe characters in the content of a message, you must use escaped ASCII formatting.

Escaping characters in HTML: *Escaped ASCII formatting* is a method to represent unsafe, accented, non-English, double-byte, and multi-byte characters using a combination of the percent character (%) and a two-digit hexadecimal number. When the text is viewed in a browser, the escaped character sequence is replaced by the equivalent ASCII value. Characters that are not included within the US-ASCII coded character set must be included in *DREAMmail* HTML tags in escaped ASCII format.

For information about the escaped ASCII codes for accented and non-English characters, see [Appendix - ASCII Escape Codes](#).

Note: Because there are so many different ASCII codes used to represent the double-byte and multi-byte based languages, Epsilon does not provide a table of these conversion values. Consult one of the many online references to obtain this kind of information.

Language Support and Character Sets

DMConnect supports sending messages and content segments that include text in standard ASCII, extended ASCII, and multi-byte characters, but does not interpret the content. The content must be passed in an ISO-charset encoding. *DREAMmail* does **not** support Unicode content.

Character sets

DREAMmail uses character sets to support deployments in multiple languages. A character set is the specific set of characters that constitute a language. Because multiple languages can use one character set, *DREAMmail*'s 14 certified character sets allow messages to be sent in over 50 languages. The following table shows each *DREAMmail* certified character set, its encoding standard, and the languages that are based on the character set:

TABLE: Certified character sets supported in *DREAMmail*

Character Set Name	Encoding Standard	Languages Supported
English	US-ASCII	English
Western European	ISO-8859-1	Dutch, French, German, Italian, Portuguese, Spanish, Swedish, English, Afrikaans, Albanian, Basque, Catalan, Danish, Faroese, Finnish, Icelandic, Irish, Norwegian, Rhaeto-Romanic, Scottish, Swahili
Eastern European	ISO-8859-2	Croatian, Czech, Hungarian, Polish, Romanian, Slovakian, Slovenian, Sorbian
Southern European	ISO-8859-3	Esperanto, Maltese
Northern European	ISO-8859-4	Estonian, Greenlandic, Lappish, Latvian, Lithuanian
Cyrillic	ISO-8859-5	Bulgarian, Byelorussian, Macedonian, Russian, Serbian, Ukrainian
Greek	ISO-8859-7	Greek
Turkish	ISO-8859-9	Turkish

TABLE: Certified character sets supported in *DREAMmail*

Character Set Name	Encoding Standard	Languages Supported
Japanese	<i>DREAMmail</i> supports EUC-JP encoding for data imports, content submission, and message composition. At the deployment stage, <i>DREAMmail</i> ensures that mailings are deployed in the ISO-2022-JP encoding, an encoding standard more widely accepted by mail servers.	Japanese
Traditional Chinese	Big5	Traditional Chinese
Simplified Chinese	GB2312	Simplified Chinese
Korean	EUC-KR	Korean
Arabic	Windows 1256 Note: The ISO encoding standard is not supported for Arabic.	Arabic
Hebrew	Windows 1255 Note: The ISO encoding standard is not supported for Hebrew and Yiddish.	Hebrew, Yiddish

Multiple character sets

In *DMConnect*, if your site supports multiple character sets, the `<Language>` tag must be used within the script to identify the character set. See [CreateMailing Request](#) and [UpdateMailing Request](#) for details on using the `<Language>` tag.

What does it mean for a language to be supported?

Language support only takes place at the deployment level, which means you can include content in any of the languages that require the certified character sets in *DREAMmail*. Language support does **not** include the localization or translation of the *DREAMmail* application, reports, email addresses, user time zones, or the names of items such as list name and message name.

Similarly, *DREAMmail* cannot interpret the text in inbound messages in any language except English. However, *DREAMmail* can forward all non-English inbound messages to an email address provided by the *DREAMmail* customer. (Examples of inbound messages are subscribe and unsubscribe requests, change of address requests, replies sent to the `From` address in a *DREAMmail* message, and so forth.)

Guidelines for using character sets

The character set that you use to deploy messages in *DREAMmail* must match the character set of your operating system and web browser. For example, when creating a message in Russian, you should use the Cyrillic (ISO-8859-5) versions of Windows 2000 and Internet Explorer. (If you deploy a message in Russian, while using an English version of the operating system or web browser, the email message may display corrupted characters.)

Similarly, the character set of the recipient's Windows operating system and web browser must match those of the message creator.

Exceptions: The Southern European character set (Esperanto and Maltese) and the Northern European character set (Estonian, Greenlandic, Lappish, Latvian, and Lithuanian) do not have localized versions of the Windows operating system. If you deploy a message that is based on these character sets, use the English versions of Windows 2000 and Internet Explorer.

Any source file that you import into *DREAMmail* must be generated using the same character encoding that *DREAMmail* supports. For example, Hebrew has both a Windows encoding and an ISO encoding standard. However, *DREAMmail* only supports the Windows encoding for Hebrew. Therefore, any Hebrew source file that you generate and import into *DREAMmail* by using the ISO encoding results in corrupted data. In this case, use the Windows encoding to maintain data integrity.

Note: When you specify a language other than English for a message, Epsilon recommends that you select the language's character set instead of the name of the language. For example, if you are creating a message in German, select *Western European (ISO-8859-1)* from the *Language* drop-down list as opposed to *German*.

Submitting multi-language content

To use languages other than English, your site must support multiple character sets, otherwise *DMConnect* returns an error. If you specify a character set that is not supported, *DMConnect* returns an error.

If the specified site does not support multiple character sets, *DMConnect* returns an error, even if the language specified is English. Do not use the `<Language>` tag if the site is not enabled for multi-language support.

Note: All language content other than English must either be escaped or placed in CDATA sections. For more information, see [Special Characters and Non-English Text](#).

Recommendations for Writing Requests

Creating successful *DMConnect* requests involves some organization and planning. Use the tips in this section to help make the process easier.

- [Maximum file size](#)
- [Tips for XML development](#)
- [Placement of content](#)
- [Entering and tracking URLs](#)
- [Time between requests](#)
- [Optimize reports](#)
- [Using Reply To addresses.](#)

Maximum file size

For maximum performance, keep *DMConnect* documents to a maximum size of 2 MB. Break up larger files into a series of smaller, interrelated files.

Important: Use either static or script content segment requests to reduce the size of your messages. For more information about content segments, see the *DREAMmail* help system and the *Advanced Messaging Guide*.

Tips for XML development

When writing XML scripts, keep the following guidelines in mind:

- Put a carriage return after every end tag. This keeps the lines shorter, making it easier to troubleshoot problems in the XML script. The XML parser also handles shorter strings better than long strings.
- Use <C> and </C> instead of <Column> and </Column>, and <R> and </R> instead of <Row> and </Row> to cut down the size of the XML document.
- Make sure that your XML scripts react appropriately to the error response messages sent by *DMConnect*.
- Before submitting a request, validate it. You can validate requests at the following website:

<http://www.stg.brown.edu/service/xmlvalid/>

Also, Microsoft has a downloadable XML validation tool. To download a copy go to:

<http://msdn.microsoft.com/downloads/default.asp?URL=/code/sample.asp?url=msdn-files/027/001/573/msdncompositedoc.xml>

Placement of content

To prevent content or special characters from being misinterpreted:

- Place all text in CDATA sections in case any special or reserved characters are present.
- All content in languages other than English must be placed within CDATA sections.

For more information, see [Special Characters and Non-English Text](#).

Entering and tracking URLs

When placing the URLs in a message, the URLs in the message body must match the URLs in the <URLName> tags. If you place a punctuation mark directly at the end of a URL in the body of the message, *DMConnect* returns an error. Therefore, you must either not use a punctuation mark at the end of the URL, or else leave a space between the punctuation mark and the URL. Also, if you include a request to track a URL that does not exist in the message body or content segment, *DMConnect* returns an error.

Time between requests

For optimal results, when you send requests, wait until you receive a confirmation message before you send the next request. Although it is possible to issue a request without receiving the confirmation for a previously issued request, an asynchronous request is not considered complete until you receive a confirmation message.

Optimize reports

To reduce workload and file size of reports:

- Do not run the *GetExportReportStatus* check more than once per minute.
- Run reports daily for the previous day to keep the file size manageable. In order to run a report for a single day, you must use the same date for both *FromDate* and *EndDate*.

Using *Reply To* addresses

You cannot specify a reply-to address for messages created in *DMConnect*. Messages created in *DMConnect* use the default reply-to address that was set up for the campaign that the message belongs to. If you do not want to use the default reply-to address for messages sent through *DMConnect*, you must edit the campaign and remove the address.

IMPLEMENTING *DMCONNECT*

The time frame for implementing *DMConnect* varies based on the expertise of the people writing the XML requests, as well as the complexity of the XML document that is being posted. The duration of this period can range from a few weeks to a few months for coding and performing a comprehensive testing phase.

This section discusses the following topics:

- [Overview of the Implementing DMConnect](#)
- [Using DMConnect With Other Products.](#)

Overview of the Implementing *DMConnect*

The process of implementing *DMConnect* is comprised of the following steps:

Stage	Explanation
1	<p>You create the request as an XML document.</p> <p>Before submitting the request, ensure that you have included all mandatory elements and escaped all reserved characters.</p> <p>Note: All XML documents require a root element. The root element within the DMConnect XML document is <DMConnect>.</p>
2	<p>You submit the request.</p> <p>The request is submitted to <i>DREAMmail</i> using an HTTP or HTTPS post. For more information, see Posting DMConnect Requests.</p>
3	<p><i>DREAMmail</i> receives the file and passes the file to an XML file management utility which then saves the file.</p>
4	<p><i>DMConnect</i> parses the XML request and validates the file.</p> <ul style="list-style-type: none"> If the XML structure is valid, <i>DMConnect</i> produces an XML DOM tree for the information in the XML file. If the structure is invalid, <i>DMConnect</i> returns an error response message. <p>For more information, see DMConnect Messages and Response Codes.</p>
5	<p><i>DMConnect</i> authenticates the user and the server to which the request is posted.</p> <p>If the operation is unsuccessful, <i>DMConnect</i> returns an error response message. For more information, see DMConnect Messages and Response Codes.</p>
6	<p><i>DMConnect</i> interprets the request.</p> <p>The XML DOM tree is parsed and <i>DMConnect</i> determines which request has been made. It then performs the requested action in <i>DREAMmail</i>.</p> <p>If the operation is:</p> <ul style="list-style-type: none"> unsuccessful, <i>DMConnect</i> returns an error response message. successful, <i>DMConnect</i> returns a success message. <p>For more information, see DMConnect Messages and Response Codes.</p>
7	<p><i>DMConnect</i> sends additional notifications (optional).</p> <p><i>DREAMmail</i> can notify specified parties other than the submitter when a request has been successfully processed. This is not the response code message that is automatically sent out by <i>DMConnect</i>. To receive a notification email, the appropriate email addresses must be listed in the <AcknowledgementsTo> element of the XML document.</p>

Using *DMConnect* With Other Products

Epsilon uses the XML requests and responses in *DMConnect* to integrate *DREAMmail* and other Epsilon products such as *ChannelView* and *Ensemble*. *DREAMmail* can also integrate *DREAMmail* with *SiteCatalyst*, a web analytics product offered by Omniture, Inc.

To integrate *DREAMmail* and *ChannelView* or *DREAMmail* and *SiteCatalyst*, you must first use the *Site Setup* component in *DREAMmail* to enable integration with either application.

Non-Epsilon products

You can use *DMConnect* to integrate *DREAMmail* and other ecommerce systems such as CRM, CMS, and so on.

SYNTAX OF *DMCONNECT* REQUESTS AND RESPONSES

This chapter is an overview of the requests accepted by *DMConnect* and lists the parameters you can use in the requests. You can use this as reference when you write XML scripts to submit to *DMConnect*.

This chapter also includes an overview of the responses that *DMConnect* returns in response to specific *DMConnect* requests.

This guide assumes that you have a working knowledge of XML. For more information, see [Writing DMConnect Requests](#).

This section discusses the following topics:

- [DMConnect Requests, Organized by Type](#)
- [Responses Returned by DMConnect Requests](#)
- [Summary of DMConnect Requests and Responses](#)
- [Parameters Accepted in DMConnect Report Requests](#)
- [FromDate and EndDate Parameters](#)
- [AddProfileFields Request](#)
- [ChangeRTMMailingStatus](#)
- [ClearList Request](#)
- [CreateCampaign Request](#)
- [CreateConditionalContent Request](#)
- [CreateContent Request](#)
- [CreateDuplicateMailing Request](#)
- [CreateList Request](#)
- [CreateMailingFromTemplate Request](#)
- [CreateMailing Request](#)
- [CreateRTMMailing](#)
- [ExportReport Request](#)
- [GetAllExportNames Request](#)
- [GetExportedReport Request](#)
- [GetExportReportStatus Request](#)

- [GetFtpImportStatus Request](#)
- [GetFtpImportStatusRs Response](#)
- [GetHygieneRuleNames Request](#)
- [GetHygieneRuleNamesRs Response](#)
- [GetJobStatus Request](#)
- [GetListNames Request](#)
- [GetListNamesRs Response](#)
- [GetMailingContent Request](#)
- [GetMailingContentRs Response](#)
- [GetMailingStatus Request](#)
- [GetMailingStatusRs Response](#)
- [GetMessageSize Request](#)
- [GetMessageSizeRs Response](#)
- [GetObjectNames Request](#)
- [GetObjectNamesRs Response](#)
- [GetProfileFields Request](#)
- [GetProfileFieldsRs Response](#)
- [GetTemplateVariables Request](#)
- [GetTemplateVariablesRs Response](#)
- [GetTestMailingStatus Request](#)
- [GetTestMailingStatusRs Response](#)
- [GetTrackedURL Request](#)
- [GetTrackedURLRs Response](#)
- [ImportList Request](#)
- [ManageLists Request](#)
- [SendMailingForApproval Request](#)
- [StartMailing Request](#)
- [TestMailing Request](#)
- [TriggerFtpImport Request](#)
- [TriggerMailingToEmailAddresses Request](#)
- [TriggerMailingToList Request](#)
- [Unsubscribe Request](#)

- [UpdateConditionalContent Request](#)
- [UpdateContent Request](#)
- [UpdateMailing Request](#)
- [UpdateProfile Request](#)
- [UpdateRTMMailing](#)
- [UpdateTrackedURL Request](#)

DMConnect Requests, Organized by Type

DMConnect handles XML-based requests and currently supports content in over 50 languages. *DMConnect* requests can be group by type of request as shown in this section.

Creating or updating message requests

These requests are:

- [CreateMailing Request](#)
- [UpdateMailing Request](#)
- [CreateMailingFromTemplate Request](#)
- [CreateDuplicateMailing Request](#)

Creating or updating content requests

These requests are:

- [CreateContent Request](#)
- [UpdateContent Request](#)
- [CreateConditionalContent Request](#)
- [UpdateConditionalContent Request](#)

Sending or triggering mailing requests

These requests are:

- [SendMailingForApproval Request](#)
- [TestMailing Request](#)
- [StartMailing Request](#)
- [TriggerMailingToList Request](#)
- [TriggerMailingToEmailAddresses Request](#).

Creating or updating mailing list requests

These requests are:

- [CreateList Request](#)
- [TriggerFtpImport Request](#)
- [GetHygieneRuleNames Request](#)
- [ClearList Request](#)
- [GetListNames Request](#)
- [ImportList Request](#)

- [UpdateProfile Request](#)
- [Unsubscribe Request](#)
- [GetFtpImportStatus Request](#).

Running or exporting report and information requests

These requests are:

- [ExportReport Request](#)
- [GetExportedReport Request](#)
- [GetExportReportStatus Request](#)
- [GetAllExportNames Request](#)
- [GetJobStatus Request](#).

Creating campaigns

These request is:

- [CreateCampaign Request](#).

Manipulating profile fields

These requests are:

- [GetProfileFields Request](#)
- [AddProfileFields Request](#).

Querying DREAMmail objects

The requests are:

- [GetObjectNames Request](#).
- [GetMailingContent Request](#)
- [GetTemplateVariables Request](#)

Obtaining message status and information

These requests are:

- [GetMailingStatus Request](#)
- [GetTestMailingStatus Request](#)
- [GetMessageSize Request](#).

Creating and updating Real-Time Messaging (RTM) requests

These requests are:

- [CreateRTMMailing](#)
- [UpdateRTMMailing](#)
- [ChangeRTMMailingStatus](#).

Updating URLs

These requests are:

- [GetTrackedURL Request](#)
- [GetTrackedURLs Response](#)
- [UpdateTrackedURL Request](#)

Responses Returned by *DMConnect* Requests

When you issue a *DMConnect* request, *DMConnect* responds with a XML document, which includes a machine readable success or failure code, as well as a descriptive text status line. For more information, see [DMConnect Messages and Response Codes](#).

Some *DMConnect* requests have an associated response that *DMConnect* returns after you submit the request. *DMConnect* responses are XML documents that contain the data output of your request. For example, the result of the [GetProfileFields Request](#) is provided by the [GetProfileFieldsRs Response](#).

You can write a program to parse the XML output to determine the result of the request and to perform subsequent actions based on the results.

The following responses are used in *DMConnect*:

- [GetFtpImportStatusRs Response](#)
- [GetHygieneRuleNamesRs Response](#)
- [GetListNamesRs Response](#)
- [GetMailingContentRs Response](#)
- [GetMailingStatusRs Response](#)
- [GetMessageSizeRs Response](#)
- [GetObjectNamesRs Response](#)
- [GetProfileFieldsRs Response](#)
- [GetTemplateVariablesRs Response](#)
- [GetTestMailingStatusRs Response](#)
- [GetTrackedURL Request](#)
- [GetTrackedURLRs Response](#).

For more information, see [Summary of DMConnect Requests and Responses](#).

Summary of *DMConnect* Requests and Responses

This section describes the requests that you can submit to *DMConnect* and the responses that *DMConnect* returns.

In this table, *DMConnect* responses are indicated by shaded rows.

TABLE: *DMConnect* requests and responses

Request/Response Name	Action
AddProfileFields	Adds a profile field at the client or site level.
ChangeRTMMailingStatus	Changes the status of a <i>RTM</i> message.
ClearList	Erases all subscribers from a mailing list.
CreateCampaign	Creates a campaign in <i>DREAMmail</i> .
CreateContent	Creates a content segment and sets the content attributes to those specified with the request.
CreateDuplicateMailing	Creates a duplicate of a specific message. All attributes, except status, of the source message are copied to the duplicate message.
CreateList	Creates a mailing list.
CreateMailing	Creates a message and sets the message attributes to those specified with the request.
CreateMailingFromTemplate	Creates a message from an existing message template.
CreateRTMMailing	Creates a <i>RTM</i> message and sets the attributes to those specified with the request.
ExportReport	Exports and saves a report.
GetAllExportNames	Lists the reports that the user can access.
GetExportedReport	Returns the binary data that contains the exported report.
GetExportReportStatus	Returns information about whether the specified <i>ExportReport</i> request has completed generating and exporting the report.
GetFtpImportStatus	Returns the status of a single import initiated by the <i>TriggerFtpImport</i> request.
GetFtpImportStatusRs	Contains output from the <i>GetFtpImportStatus</i> request.

TABLE: *DMConnect* requests and responses

Request/Response Name	Action
GetHygieneRuleNames	Returns the names of the hygiene rules defined for a specific site.
GetHygieneRuleNamesRs	Contains output from the <i>GetHygieneRuleNames</i> request.
GetJobStatus	Returns the status of a requested job.
GetListNames	Returns names of the mailing lists for a specific site.
GetListNamesRs	Contains output from the <i>GetListNames</i> request.
GetMailingContent	Returns the content for a specific message.
GetMailingContentRs	Contains output from the <i>GetMailingContent</i> request.
GetMailingStatus	Returns the status of a message.
GetMailingStatusRs	Contains output from the <i>GetMailingStatus</i> request.
GetMessageSize	Returns the size of a specific message.
GetMessageSizeRs	Contains output from the <i>GetMessageSize</i> request.
GetObjectNames	Returns the main <i>DREAMmail</i> object names such as clients, sites, campaigns, messages, and mailing lists.
GetObjectNamesRs	Contains output from the <i>GetObjectNames</i> request.
GetProfileFields	Returns the system and user-customized profile fields at the client level, the site level, or both.
GetProfileFieldsRs	Contains output from the <i>GetProfileFields</i> request.
GetTemplateVariables	Returns the variables of a specific template.
GetTemplateVariablesRs	Contains output from the <i>GetTemplateVariables</i> request.
GetTestMailingStatus	Returns the status of a specific test message.
GetTestMailingStatusRs	Contains output from the <i>GetTestMailingStatus</i> request.
GetTrackedURL	Contains the information associated with the tracked URLs within a message.
GetTrackedURLRs	Returns the output from the GetTrackedURL request.
ImportList	Adds a set of email addresses to a list.
SendMailingForApproval	Sends a specified message for approval.
StartMailing	Sends a specified message.

TABLE: *DMConnect* requests and responses

Request/Response Name	Action
TestMailing	Sends a specified message for proof reading.
TriggerFtpImport	Initiates a single FTP import.
TriggerMailingToEmailAddresses	Triggers a message to a set of selected email addresses.
TriggerMailingToList	Triggers a message to a set of mailing lists.
Unsubscribe	Removes the specified email addresses from a client, sites, or lists.
UpdateContent	Modifies an existing content segment and sets the content attributes to those specified with the request.
UpdateMailing	Modifies an existing message and sets the message attributes to those specified with the request.
UpdateProfile	Updates the profile fields for the specified email addresses.
UpdateRTMMailing	Updates an existing <i>RTM</i> message and sets the message attributes to those specified with the request.
UpdateTrackedURL	Contains the information associated with a message URL to be updated.

Parameters Accepted in *DMConnect* Report Requests

You can only include certain parameters in a *DMConnect* request. The only parameter names accepted are:

- ClientName
- SiteName
- CampaignName
- MailingName
- ListName
- IncludeField
- ActivityType
- FromDate
- EndDate.

When using date-related formats there are some specific details that you must understand. For more information, see [FromDate and EndDate Parameters](#).

Note: When you include parameters in your request, remember that XML is case-sensitive.

FromDate and EndDate Parameters

The use of the *FromDate* and *EndDate* are subject to some restrictions. This section describes the format and constraints of these parameters.

Format You enter the *FromDate* and *EndDate* parameter using either the slash (/) or a hyphen (-) as a separator. You must specify the date as year, month, day. The two formats are:

- `yyyy/mm/dd` (for example, 2005/01/30)
- `yyyy-mm-dd` (for example, 2005-01-30).

In a XML document, you can enter the parameters as shown in the two examples below.

Example 1: This example uses a slash as a separator.

```
<parameter>
  <Name>FromDate</Name>
  <Value>2002/1/30</Value>
</parameter>
<parameter>
  <Name>EndDate</Name>
  <Value>2002/06/30</Value>
</parameter>
```

Example 2: This example uses a hyphen as a separator.

```
<parameter>
  <Name>FromDate</Name>
  <Value>2002-01-30</Value>
</parameter>
<parameter>
  <Name>EndDate</Name>
  <Value>2002-06-30</Value>
</parameter>
```

Interval and date range

The *FromDate* and *EndDate* are inclusive. This means that the date range you specify includes the data for both the starting and ending dates. Also, the interval between the *FromDate* and the *EndDate* cannot exceed six months.

AddProfileFields Request

An *AddProfileFields* request adds a profile field at the client or site level. A profile field at the client level cannot have the same name as a profile field at the site level.

When you issue an *AddProfileFields* request, *DMConnect* responds with a generic *DMResponse* element that indicates the request's success or failure. For *DMConnect*'s response codes, see [Types of DMConnect response messages](#).

Elements in the request

The *AddProfileFields* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in an *AddProfileFields* request

Element Name	Description	Nesting
<AddProfileFields>	Contains the information about an <i>AddProfileFields</i> request.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>GetProfileFields</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientFields>	A composite element that consists of the <i>ClientName</i> and <i>FieldDfn</i> elements.	Nest within a <i>AddProfileFields</i> element.

TABLE: Description of the elements in an *AddProfileFields* request

Element Name	Description	Nesting
<ClientName>	The client in which the profile file is to be added.	Nest within a <i>ClientFields</i> element. You can include this more than once.
<FieldDfn>	A composite element that contains the <i>Name</i> , <i>FieldType</i> , and <i>Length</i> elements. If not present, no new client-level profile fields are added.	Nest within a <i>ClientFields</i> element.
<Name>	Name of the profile field.	Nest within a <i>FieldDfn</i> element.
<FieldType>	The data type of the profile field: <ul style="list-style-type: none"> <i>DATE</i> — format: yyyy-mm-dd <i>INT</i> — integer <i>STRING</i> — a string of characters of variable length. 	Nest within a <i>FieldDfn</i> element.
<Length>	Length of the profile field. This is only used if the profile field type is <i>STRING</i> .	Nest within a <i>FieldDfn</i> element. This is mandatory when <i>FieldType</i> is <i>STRING</i> , otherwise do not include it.
<SiteFields>	A composite element that consists of the <i>SiteName</i> and <i>FieldDfn</i> elements.	Optional. Nest within a <i>AddProfileFields</i> element.
<SiteName>	The site in which the profile file is to be added.	Nest within a <i>SiteFields</i> element.
<FieldDfn>	A composite element that contains the <i>Name</i> , <i>FieldType</i> , and <i>Length</i> elements. See previous definition of this element in this table.	Nest within a <i>SiteFields</i> element.

Example of an *AddProfileFields* request:

```

<DMConnect>
  <AddProfileFields>
    <ClientFields>
      <ClientName>OnlineDeals</ClientName>
      <FieldDfn>
        <Name>Age</Name>
        <FieldType>int</FieldType>

```

```
        </FieldDfn>
        <FieldDfn>
            <Name>Sex</Name>
            <FieldType>string</FieldType>
            <Length>1</Length>
        </FieldDfn>
    </ClientFields>
    <SiteFields>
        <SiteName>CarsForLess</SiteName>
        <FieldDfn>
            <Name>CarType</Name>
            <FieldType>string</FieldType>
            <Length>20</Length>
        </FieldDfn>
    </SiteFields>
</AddProfileFields>
</DMConnect>
```

ChangeRTMMailingStatus

The *ChangeRTMMailingStatus* request updates the status of an existing *RTM* message. A *RTM* message can have a status of *Armed*, *Open*, or *Close*.

For more information about *RTM* and a description of related elements, see the *Real-Time Messaging Guide*.

ClearList Request

A *ClearList* request erases all subscribers from a mailing list.

Elements in the request

The *ClearList* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *ClearList* request

Element Name	Description	Nesting
<ClearList>	Contains information for erasing all names from a mailing list.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses to which the notification request is sent.	Optional. Nest within a <i>ClearList</i> element. You can include this more than once.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client.	Nest within a <i>ClearList</i> element.
<SiteName>	The site within the specified client.	Nest within a <i>ClearList</i> element.
<ListName>	The list within the specified client and site.	Nest within a <i>ClearList</i> element.

Example of a ClearList request:

```
<DMConnect>
  <ClearList>
    <ClientName>CompanyABC</ClientName>
    <SiteName>Region1</SiteName>
    <ListName>TestList</ListName>
  </ClearList>
</DMConnect>
```

CreateCampaign Request

A *CreateCampaign* request creates an email campaign. For more information about creating campaigns, see the *DREAMmail* help system.

Elements in the request

The *CreateCampaign* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *CreateCampaign* request

Element Name	Description	Nesting
<CreateCampaign>	Contains information about the report to be exported.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>CreateCampaign</i> element. You can include this more than once.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
CampaignType	<p>The campaign type to create:</p> <ul style="list-style-type: none">• <i>DirectMailCampaign</i>• <i>ApprovalCampaign</i>.	Nest within a <i>CreateCampaign</i> element.
ClientName	The client that contains the site where you want to create campaigns.	Nest within a <i>CreateCampaign</i> element.

TABLE: Description of the elements in a *CreateCampaign* request

Element Name	Description	Nesting
SiteName	The site where you want to create campaigns.	Nest within a <i>CreateCampaign</i> element.
CampaignName	The campaign that you want to create. This must be unique within a site.	Nest within a <i>CreateCampaign</i> element.
ExpiresDate	The last date that a message from this campaign can be sent.	Nest within a <i>CreateCampaign</i> element.
PublisherEmailAddress	Address of the person who can modify or send an approval message.	Optional. Nest within a <i>CreateCampaign</i> element.
FromAlias	<i>From</i> address for the campaign. <i>Note: This address must already exist in DREAMmail.</i>	Nest within a <i>CreateCampaign</i> element.
ReplyToAlias	<i>Reply To</i> address for the campaign. <i>Note: This address must already exist in DREAMmail.</i>	Optional. Nest within a <i>CreateCampaign</i> element.
SubscribeAlias	Equivalent to the <i>Display Name</i> field for the <i>Subscribe</i> address in the campaign. <i>Note: This address must already exist in DREAMmail.</i>	Optional. Nest within a <i>CreateCampaign</i> element.
UnsubscribeAlias	Equivalent to the <i>Display Name</i> field for the <i>Unsubscribe</i> address in the campaign. <i>Note: This address must already exist in DREAMmail.</i>	Optional. Nest within a <i>CreateCampaign</i> element.
ApprovalList	The mailing list to which approval messages are sent.	Optional. Nest within a <i>CreateCampaign</i> element.
TestList	The list to which the test messages are sent.	Optional. Nest within a <i>CreateCampaign</i> element.

Example of a CreateCampaign request:

```
<DMConnect>
  <CreateCampaign>
    <AcknowledgementsTo>
      <EmailAddress>Bob@Epsilon.com</EmailAddress>
      <EmailAddress>Dave@Epsilon.com</EmailAddress>
      <Option>9</Option>
    </AcknowledgementsTo>
    <CampaignType>ApprovalCampaign</CampaignType>
    <ClientName>DREAMmailTipster</ClientName>
    <SiteName>DREAMmailTips</SiteName>
    <CampaignName>Stock_Alert</CampaignName>
    <ExpiresDate>2002-12-31</ExpiresDate>
    <PublisherEmailAddress></PublisherEmailAddress>
    <FromAlias>DMT</FromAlias>
    <ReplyToAlias>reply@reply.com</ReplyToAlias>
    <SubscribeAlias>JoinUs</SubscribeAlias>
    <UnsubscribeAlias>LeaveUs</UnsubscribeAlias>
    <ApprovalList>prooflist_1</ApprovalList>
    <TestList>Tipstestlist_1</TestList>
  </CreateCampaign>
</DMConnect>
```

CreateConditionalContent Request

A *CreateConditionalContent* request enables sending of content that varies from subscriber to subscriber based on subscriber data.

Elements in the request

The *CreateConditionalContent* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *CreateConditionalContent* request

Element Name	Description	Nesting
<CreateConditionalContent>	Contains the information associated with a content segment.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>CreateConditionalContent</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client that contains the site that contains the content segments.	Nest within a <i>CreateConditionalContent</i> element.
<SiteName>	The site that contains the content segment.	Nest within a <i>CreateConditionalContent</i> element.

TABLE: Description of the elements in a *CreateConditionalContent* request

Element Name	Description	Nesting
<ContentName>	The content segment to create.	Nest within a <i>CreateConditionalContent</i> element.
<Language>	<p>Specifies the character set of the message's content. Only use if the site supports multiple character sets. Default is English (US-ASCII).</p> <p>Specify character sets, such as <i>Cyrillic (ISO-8859-5)</i>, instead of languages. Check the <i>DREAMmail</i> application for the exact spelling. For more information, see Language Support and Character Sets and Submitting multi-language content.</p>	Optional. Nest within a <i>CreateConditionalContent</i> element.
<ConditionalContent>	Includes information pertaining to specific condition which include the condition itself, the profile field to which condition will be applied, contents to be included etc.	This element must be nested within a <i>CreateConditionalContent</i> element.
< IF>	It includes profile field to which condition will be applied.	This element must be nested within a <i>ConditionalContent</i> element.

TABLE: Description of the elements in a *CreateConditionalContent* request

Element Name	Description	Nesting
< Condition >	It includes the condition operator. Text operators: contains, equals, does not equal, comes before, comes before or equals, comes after, comes after or equals. Date/Numeric Operators in CreateConditionalContent are as follows: <ul style="list-style-type: none">• equals• notequals• lessthan• lessthanorequalto• greaterthan• greaterthanorequalto	This element must be nested within a <i>ConditionalContent</i> element.
<ConditionValue>	It include condition data for the conditional profile field.	This element must be nested within a <i>ConditionalContent</i> element.
< ContentBody>	It include body to be included for a successful specific condition.	This element must be nested within a <i>ConditionalContent</i> element.
<TextBody>	Content for text-only email applications.	Optional. Nest within a <i>ContentBody</i> element.
<HtmlBody>	Content for HTML-capable email applications.	Optional. Nest within a <i>ContentBody</i> element.
<AolBody>	Content for AOL-capable email applications.	Optional. Nest within a <i>ContentBody</i> element.

TABLE: Description of the elements in a *CreateConditionalContent* request

Element Name	Description	Nesting
<BodyText>	<p>The message content text.</p> <p>Important: You should always use escaped ASCII formatting for the content in the event that special or reserved characters are used. For more information, see Special Characters and Non-English Text.</p> <p>Important: When you include URLs in this element, the URLs must exactly match the way the URLs are specified in the <URLName> element. Do not place punctuation at the end of the URL in this element. Punctuation placed at the end of a URL causes the request to fail.</p>	Nest within a <i>TextBody</i> , <i>AolBody</i> , or <i>HtmlBody</i> element.
<DefaultContent>	Includes information pertaining to the default condition which includes the condition itself, profile field to which condition will be applied. This element is exactly the same as the ConditionalContent element in structure.	This element must be nested within a <i>CreateConditionalContent</i> element.
<TrackedUrls>	The URLs to track. Each URL is identified with its own <URL></URL> element.	Optional. This element must be nested within a <i>CreateConditionalContent</i> element.
<URL>	The URL to track.	Nest within a <i>TrackedUrls</i> element. You can include this more than once.
<URLName>	The actual URL to track; for example <code>http://www.somedomain.com</code>	Nest within a <i>URL</i> element.
<URLLabel>	Assigns a label to the URL to track.	Optional. Nest within a <i>URL</i> element.

Example of a CreateConditionalContent request:

```
<?xml version="1.0" encoding="UTF-8"?>
  <DMConnect>
    <CreateConditionalContent>
```



```

    <AcknowledgementsTo>
      <EmailAddress>mukul@cybage.com</EmailAddress>
      <Option>9</Option>
    </AcknowledgementsTo>
    <ClientName>Client_Nitin</ClientName>
    <SiteName>Site_Nitin</SiteName>
    <ContentName>Great_5</ContentName>
    <Language>English</Language>
    <ConditionalContent>
      <EnableHTMLEditor>true</EnableHTMLEditor>
      <IF>FirstName</IF>
      <Condition>equals</Condition>
      <ConditionValue>Mukul</ConditionValue>
      <ContentBody>
        <TextBody>
          <BodyText>
            hi 1 in text part on 29 April.
            http://www.Great.com
          </BodyText>
        </TextBody>
        <AolBody>
          <BodyText>
            hi 1 in Aol part on 29 April.
          </BodyText>
        </AolBody>
        <HtmlBody>
          <BodyText>
          </BodyText>
        </HtmlBody>
      </ContentBody>
    </ConditionalContent>
    <Defaultcontent>
      <EnableHTMLEditor>true</EnableHTMLEditor>
      <TextBody>
        <BodyText>
          Great Default text part on 29 April.
          Visit again.
        </BodyText>
      </TextBody>
      <AolBody>
        <BodyText>
          Great Default Aolpart on 29 April.
        </BodyText>
      </AolBody>
      <HtmlBody>
        <BodyText>
          <a href="http://www.Great.com">Great Default Html part on 29 April.</a>

```

```
</BodyText>
</HtmlBody>
</Defaultcontent>
<TrackedUrls>
<URL>
<URLName>http://www.Great.com</URLName>
<URLLabel>GreatURL</URLLabel>
</URL>
</TrackedUrls>
</CreateConditionalContent>
</DMConnect>
```

CreateContent Request

A *CreateContent* request creates a new content segment of the type specified.

Elements in the request

The *CreateContent* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *CreateContent* request

Element Name	Description	Nesting
<CreateContent>	Contains the information associated with a content segment.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>CreateContent</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ContentType>	<p>The content segment type to create:</p> <ul style="list-style-type: none">• <i>StandardContent</i>• <i>ScriptContent</i>.	Nest within a <i>CreateContent</i> element.
<ClientName>	The client that contains the site that contains the content segments.	Nest within a <i>CreateContent</i> element.

TABLE: Description of the elements in a *CreateContent* request

Element Name	Description	Nesting
<SiteName>	The site that contains the content segment.	Nest within a <i>CreateContent</i> element.
<ContentName>	The content segment to create.	Nest within a <i>CreateContent</i> element.
<Language>	<p>Specifies the character set of the message's content. Only use if the site supports multiple character sets. Default is English (US-ASCII).</p> <p>Specify character sets, such as <i>Cyrillic (ISO-8859-5)</i>, instead of languages. Check the <i>DREAMmail</i> application for the exact spelling. For more information, see Language Support and Character Sets and Submitting multi-language content.</p>	Optional. Nest within a <i>CreateContent</i> element.
<ContentText>	<p>Delimits the content of the message. This can contain a script content segment. For more information about script content segments, see the <i>Advanced Messaging Guide</i>.</p> <p>Important: When including URLs in the segment, they must match the URLs in the <URLName> tags. Either do not use a punctuation mark at the end of the URL, or leave a space between the punctuation mark and the URL. If you place a punctuation mark at the end of a URL in the content segment, the request fails.</p> <p>Note: Content in a language other than English must be in a CDATA section or escaped. For more information, see Special Characters and Non-English Text.</p>	Nest within a <i>CreateContent</i> element.

TABLE: Description of the elements in a *CreateContent* request

Element Name	Description	Nesting
<TrackedUrls>	The URLs tracked within this content. Each URL is identified with its own <URL></URL> element.	Optional. Nest within a <i>CreateContent</i> element.
<URL>	The URL tracked.	Nest within a <i>TrackedUrls</i> element. You can include this more than once.
<URLName>	The name of the URL to track; for example <code>http://www.somedomain.com</code> .	Nest within a <i>URL</i> element.
<URLLabel>	Assigns a label to the URL to track.	Optional. Nest within a <i>URL</i> element.

Example of a *CreateContent* request: This content segment is referenced in the HTML and AOL messages in the *UpdateMailing* request (see [Example of an UpdateMailing request](#)).

```

<DMConnect>
  <CreateContent>
    <AcknowledgementsTo>
      <EmailAddress>deployment@Epsilon.com</EmailAddress>
      <Option>0</Option>
    </AcknowledgementsTo>
    <ContentType>ScriptContent</ContentType>
    <ClientName>DREAMmailTipster</ClientName>
    <SiteName>DMTips</SiteName>
    <ContentName>Tips_03162002</ContentName>
    <ContentText> <![CDATA[
      <If condition = "#City#='New York'">
        <font face = "arial"><a href = "http://DMtips3.com">New Yorkers,
        click here for an extra tip</a></font><br>
      </If>]]>
    </ContentText>
    <TrackedUrls>
      <URL>
        <URLName>http://www.DMtips3.com</URLName>
        <URLLabel>NY Tips</URLLabel>
      </URL>
    </TrackedUrls>
  </CreateContent>
</DMConnect>

```

CreateDuplicateMailing Request

A *CreateDuplicateMailing* request creates a copy of a specified message. All attributes, except status, of the source message are copied to the duplicate message. For standard messages, the status of the newly duplicated message is “To be sent;” for triggered messages the status is “Composing.”

Elements in the request

The *CreateDuplicateMailing* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *CreateDuplicateMailing* request

Element Name	Description	Nesting
<CreateDuplicateMailing>	Creates a duplicate of a specified message.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>CreateDuplicateMailing</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none"> • 0 — no notifications • 1 — only send confirmation acknowledgements • 2 — only send failure acknowledgements • 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client that contains the message that you are duplicating.	Nest within a <i>CreateDuplicateMailing</i> element.
<SiteName>	The site that contains the message that you are duplicating.	Nest within a <i>CreateDuplicateMailing</i> element.

TABLE: Description of the elements in a *CreateDuplicateMailing* request

Element Name	Description	Nesting
<CampaignName>	The campaign that contains the message that you are duplicating.	Nest within a <i>CreateDuplicateMailing</i> element.
<MailingName>	The message that you are duplicating.	Nest within a <i>CreateDuplicateMailing</i> element.
<NewMailingName>	The name of the newly duplicated message.	Nest within a <i>CreateDuplicateMailing</i> element.

Example of a *CreateDuplicateMailing* request:

```
<DMConnect>
  <CreateDuplicateMailing>
    <AcknowledgementsTo>
      <EmailAddress>deployment@Epsilon.com</EmailAddress>
      <Option>0</Option>
    </AcknowledgementsTo>
    <ClientName>Client1</ClientName>
    <SiteName>Site2</SiteName>
    <CampaignName>Campaign_04162005</CampaignName>
    <MailingName>mailing1</MailingName>
    <NewMailingName>mailing1_duplicated</NewMailingName>
  </CreateDuplicateMailing>
</DMConnect>
```

CreateList Request

A *CreateList* request creates a mailing list. You must use the *ImportList* request to add subscribers and their profile data to the list. For information, see [ImportList Request](#).

Elements in the request

The *CreateList* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *CreateList* request

Element Name	Description	Nesting
<CreateList>	Contains the information for the new mailing list.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>CreateList</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The type of acknowledgement that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: If you chose 0 as an option, you still must include content in the <EmailAddress> element in your request.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client in which the list is created.	Nest within a <i>CreateList</i> element.
<SiteName>	The site in which the list is created.	Nest within a <i>CreateList</i> element.
<ListName>	The mailing list to be created.	Nest within a <i>CreateList</i> element.

Example of a CreateList request:

```
<DMConnect>
  <CreateList>
    <AcknowledgementsTo>
      <EmailAddress>deployment@Epsilon.com</EmailAddress>
      <EmailAddress>hot_tip_editor@Epsilon.com</EmailAddress>
      <Option>9</Option>
    </AcknowledgementsTo>
    <ClientName>DREAMmailTipster</ClientName>
    <SiteName>DMTips</SiteName>
    <ListName>DMTips01List</ListName>
  </CreateList>
</DMConnect>
```

CreateMailing Request

A *CreateMailing* request creates a message of the type that you specify.

Elements in the request

The *CreateMailing* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *CreateMailing* request

Element Name	Description	Nesting
<CreateMailing>	Contains the information associated with a message.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>CreateMailing</i> element.
<EmailAddress>	The email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The type of acknowledgement that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: If you chose 0 as an option, you still must include content in the <EmailAddress> element in your request.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<MailingType>	<p>Specifies the message type to be created:</p> <ul style="list-style-type: none">• <i>StandardMailing</i>• <i>NotificationTriggerMailing</i>• <i>RecurringTriggerMailing</i>• <i>ApprovalMailing</i>.	Nest within a <i>CreateMailing</i> element.

TABLE: Description of the elements in a *CreateMailing* request

Element Name	Description	Nesting
<ClientName>	The client that contains the site where you want to create the message.	Nest within a <i>CreateMailing</i> element.
<SiteName>	The site that contains the campaign where you want to create the message.	Nest within a <i>CreateMailing</i> element.
<CampaignName>	The campaign where you want to create the message.	Nest within a <i>CreateMailing</i> element.
<MailingName>	The message you want to create. This name must be unique within a campaign.	Nest within a <i>CreateMailing</i> element.
<FromAddr>	From address for mailing.	Optional. This element must be nested within a <i>CreateMailing</i> .
<AliasName>	From Address for message level.	This element must be nested within FromAddr element.
<DisplayName>	Display name for From Address.	Optional. This element must be nested within a FromAddr element.
<ToList>	The set of mailing lists to which the message is sent.	Optional. Nest within a <i>CreateMailing</i> element. <i>Note: If the MailingType is StandardMailing or ApprovalMailing, then this is mandatory.</i>
<ListName>	A mailing list to which the message is to be sent.	Nest within a <i>ToList</i> element. You can include this more than once.
<Filter>	A SQL <i>where</i> clause that restricts the set of subscribers who receive the message.	Optional. Nest within a <i>CreateMailing</i> element.
<Suppression>	Exclude members who have received previous mailings or campaigns.	Optional. Nest within a <i>CreateMailing</i> element.

TABLE: Description of the elements in a *CreateMailing* request

Element Name	Description	Nesting
<SuppressCampaign>	Exclude members who have received previous mailings from a specified campaign.	Nest within a <i>Suppression</i> element.
<CampaignName>	Name of campaign to be suppressed.	Nest within a <i>SuppressCampaign</i> element.
<MailingName>	Name of mailing to be suppressed.	Optional. Nest within a <i>CampaignName</i> element.
<Subject>	The subject of the message. Important: You should always use escaped ASCII formatting for the subject in the event that special or reserved characters are used. For more information see Special Characters and Non-English Text .	Nest within a <i>CreateMailing</i> element.
<Language>	Specifies the character set of the message's content. Only use if the site supports multiple character sets. Default is English (US-ASCII). Specify character sets, such as <i>Cyrillic (ISO-8859-5)</i> , instead of languages. Check the <i>DREAMmail</i> application for the exact spelling. For more information, see Language Support and Character Sets and Submitting multi-language content .	Optional. Nest within a <i>CreateMailing</i> element.
<MessageContent>	Delimits the message content. Note: There must be content within at least one of <i>TextBody</i> , <i>HtmlBody</i> , or <i>AolBody</i> .	Nest within a <i>CreateMailing</i> element.
<TextBody>	Content for text-only email applications.	Optional. Nest within a <i>MessageContent</i> element.
<HtmlBody>	Content for HTML-capable email applications.	Optional. Nest within a <i>MessageContent</i> element.

TABLE: Description of the elements in a *CreateMailing* request

Element Name	Description	Nesting
<AolBody>	Content for AOL-capable email applications.	Optional. Nest within a <i>MessageContent</i> element.
<BodyText>	<p>The message content text.</p> <p>Important: You should always use escaped ASCII formatting for the content in the event that special or reserved characters are used. For more information, see Special Characters and Non-English Text.</p> <p>Important: When you include URLs in this element, the URLs must exactly match the way the URLs are specified in the <URLName> element. Do not place punctuation at the end of the URL in this element. Punctuation placed at the end of a URL causes the request to fail.</p>	Nest within a <i>TextBody</i> , <i>AolBody</i> , or <i>HtmlBody</i> element.
<TrackedUrls>	The URLs to track. Each URL is identified with its own <URL></URL> element.	Optional. Nest within a <i>TextBody</i> , <i>AolBody</i> , or <i>HtmlBody</i> element.
<URL>	The URL to track.	Nest within a <i>TrackedUrls</i> element. You can include this more than once.
<URLName>	The actual URL to track; for example <code>http://www.somedomain.com</code>	Nest within a <i>URL</i> element.
<URLLabel>	Assigns a label to the URL to track.	Optional. Nest within a <i>URL</i> element.
<HtmlOpenRateFlag>	Set this to <i>true</i> or <i>1</i> , to enable HTML open rate tracking. Any other value disables tracking. By default, tracking is disabled.	Optional. Nest within a <i>CreateMailing</i> element.
<AutosenseFlag>	Set this to <i>true</i> or <i>1</i> to enable autosensing. Any other value disables autosensing. By default, autosensing is disabled.	Optional. Nest within a <i>CreateMailing</i> element.

TABLE: Description of the elements in a *CreateMailing* request

Element Name	Description	Nesting
<RetryAutosenseFlag>	Verifies subscriber email applications previously marked as text-only. Set this to <i>true</i> or <i>1</i> , to verify if addresses marked text-only are HTML-capable. Any other value disables this action. By default, this is disabled.	Optional. Nest within a <i>CreateMailing</i> element.
<IncludeCampaignFooter Flag>	Set this to <i>true</i> or <i>1</i> to include the campaign footer within the text message body. Any other value excludes the footer. By default, the footer is excluded. <i>Note: This is also referred to as the Message Footer. For more information, see the DREAMmail. help system.</i>	Optional. Nest within a <i>CreateMailing</i> element.
<CookieOptionflag>	Specify when <i>DREAMmail</i> sends a cookie to the subscriber. If omitted, this defaults to 0. If HTML Open Rate tracking is enabled, the accepted values are: <ul style="list-style-type: none"> • 0 — set cookie on click-through • 1 — set cookie on message open • 2 — set cookie on both click-through and message open. If HTML Open Rate tracking is not enabled, you can only set this to 0.	Optional. Nest within a <i>CreateMailing</i> element.
<ChannelViewFlag>	Set this to <i>true</i> or <i>1</i> to use an automated process to extract message data (once deployment is complete) and report on it using the <i>ChannelView</i> analytics tool.	Optional. Nest within a <i>CreateMailing</i> element.

TABLE: Description of the elements in a *CreateMailing* request

Element Name	Description	Nesting
<SendMultipartFlag>	Set this to <i>true</i> or <i>1</i> to send all messages in multi-part format regardless of the individual recipient settings in <i>DREAMmail</i> . Used primarily for European ISPs that handle HTML email as a file attachment.	Optional. Nest within a <i>CreateMailing</i> element.
<ReplyToAddr>	Reply-To-Address for mailing.	Optional. This element must be nested within a <i>CreateMailing</i> . <i>Note: If using TrackUnsubscribes do not use ReplyToAddr element.</i>
<AliasName>	Reply-To-Address for message level.	Optional. This element must be nested within <i>ReplyToAddr</i> element. This element cannot be included if <i>TrackUnsubFromReplies</i> element is included.
<DisplayName>	Display name for <i>Reply_To_Address</i> .	Optional. This element must be nested within a <i>ReplyToAddr</i> element.
<TrackUnsubFromReplies >	To enable or disable the Track unsubscribes from replies. <ul style="list-style-type: none"> • 0 • 1 • True • False (Case insensitive)	Optional. This element must be nested within a <i>ReplyToAddr</i> element. This element cannot be included if <i>AliasName</i> element is included.

TABLE: Description of the elements in a *CreateMailing* request

Element Name	Description	Nesting
<TrackUnsubscribes>	Set for specifying tracking unsubscribe flags. Leave empty to inherit default campaign values	Optional. Nest within a <i>CreateMailing</i> element. Note: If using <i>ReplyToAddr</i> and/or <i>SubunsubAddr</i> elements, do not use <i>TrackUnsubscribes</i> .
<RepliesFlag>	Set to 1 or true to enable track unsubscribe requests from replies. Set to 0 or false to disable.	Optional. Nest within a <i>TrackUnsubscribes</i> element.
<UnsubscriptionAliasFlag>	Set to 1 or true to enable track unsubscribe requests from unsubscription address. Set to 0 or false to disable.	Optional. Nest within a <i>TrackUnsubscribes</i> element.
<ArmMessage>	This will be contains value that will be for Arm or Open message for Notification Triggered messages. <ul style="list-style-type: none"> • 1 • 0 • True • False (case insensitive)	Optional. This element must be nested within a <i>CreateMailing</i> . This element can only be included for Notification Triggered messages.
< SubunsubAddr>	It includes Sub, Unsub and TrackUnsub element tags.	Optional. This element must be nested within a <i>CreateMailing</i> . Note: If using <i>TrackUnsubscribes</i> do not use <i>SubunsubAddr</i> element.

TABLE: Description of the elements in a *CreateMailing* request

Element Name	Description	Nesting
< SubscribeAlias>	Subscription alias at message level. If not specified default will be taken from campaign level.	Optional. This element must be nested within SubunsubAddr element.
<UnsubscribeAlias>	Unsubscription alias at message level. If value not specified, default will be taken from campaign level.	Optional. This element must be nested within SubunsubAddr element.
<TrackUnsub>	To enable or disable the Track unsubscribes from Unsub address. If value not specified, default will be taken from campaign level. <ul style="list-style-type: none"> • 0 • 1 • True • False (Case insensitive)	Optional. This element must be nested within SubunsubAddr element.

Example of a CreateMailing request:

```

<DMConnect>
  <CreateMailing>
    <AcknowledgementsTo>
      <EmailAddress>deployment@Epsilon.com</EmailAddress>
      <EmailAddress>hot_tip_editor@Epsilon.com</EmailAddress>
      <Option>9</Option>
    </AcknowledgementsTo>
    <MailingType>StandardMailing</MailingType>
    <ClientName>DREAMmailTipster</ClientName>
    <SiteName>DMTips</SiteName>
    <CampaignName>MarchTips</CampaignName>
    <MailingName>DMTips_03162002</MailingName>
    <FromAddr>
      <AliasName>From_Addr</AliasName>
      <DisplayName>Name</DisplayName>
    </FromAddr>
    <ToList>
      <ListName>DMTipsList</ListName>

```

```

<ListName>DMTipsAccepters</ListName>
</ToList>
<Suppression>
  <SuppressCampaign>
    <CampaignName>suppress_campaign_1</CampaignName>
    <MailingName>suppress_mailing_1</MailingName>
    <MailingName>suppress_mailing_2</MailingName>
  </SuppressCampaign>
  <SuppressCampaign>
    <CampaignName>suppress_campaign_2</CampaignName>
  </SuppressCampaign>
</Suppression>
<Subject>Hot DM Tip for March 16</Subject>
<MessageContent>
  <TextBody>
    <BodyText> <![CDATA[
Dear #FirstName#,
These are your Hot DREAMmail Tips for March 16 in text format.
http://www.DMtips.com
http://www.DMtips2.com
Thanks,
The DREAMmailTipster]]>
    </BodyText>
    <TrackedUrls>
      <URL>
        <URLName>http://www.DMtips.com</URLName>
      </URL>
      <URL>
        <URLName>http://www.DMtips2.com</URLName>
      </URL>
    </TrackedUrls>
  </TextBody>
  <HtmlBody>
    <BodyText>
    <![CDATA[<HTML><body>
Dear #FirstName#,<br>
These are your Hot DREAMmail Tips for March 16 in html format.<br>
<a href="http://www.DMtips.com">Click Here for the first tip</a><br>
<a href="http://www.DMtips2.com">Click Here for the second tip</a><br>
Thanks,<br>
The DREAMmailTipster</body></HTML>
]]>
    </BodyText>
    <TrackedUrls>
      <URL>
        <URLName>http://www.DMtips.com</URLName>
      </URL>

```

```

        <URL>
        <URLName>http://www.DMtips2.com</URLName>
    </URL>
</TrackedUrls>
</HtmlBody>
<AolBody>
    <BodyText>
<![CDATA[<HTML><body>
Dear #FirstName#,<br>
These are your Hot DREAMmail Tips for March 16 in AOL format.<br>
<a href="http://www.DMtips.com">Click Here for the tip</a><br>
<a href="http://www.DMtips2.com">Click Here for the second tip</a><br>
Thanks,<br>
The DREAMmailTipster</body></HTML>
]]>
    </BodyText>
</TrackedUrls>
    <URL>
    <URLName>http://www.DMtips.com</URLName>
    </URL>
    <URL>
    <URLName>http://www.DMtips2.com</URLName>
    </URL>
</TrackedUrls>
</AolBody>
</MessageContent>
<HtmlOpenRateFlag>True</HtmlOpenRateFlag>
<AutosenseFlag>True</AutosenseFlag>
<RetryAutosenseFlag>True</RetryAutosenseFlag>
<IncludeCampaignFooterFlag>True</IncludeCampaignFooterFlag>
<ReplyToAddr>
    <AliasName>ReplyToAddress_Name</AliasName>
    <DisplayName>Name</DisplayName>
    <TrackUnsubFromReplies>0</TrackUnsubFromReplies>
</ReplyToAddr>
<SubunsubAddr>
    <SubscribeAlias>None</SubscribeAlias>
    <UnsubscribeAlias>Unsub_addr</UnsubscribeAlias>
    <TrackUnsub>true</TrackUnsub>
</SubunsubAddr>
<CookieOptionFlag>1</CookieOptionFlag>
<ChannelViewFlag>1</ChannelViewFlag>
<SendMultipartFlag>1</SendMultipartFlag>
<TrackUnsubscribes>
    <RepliesFlag>1</RepliesFlag>
    <UnsubscriptionAliasFlag>1</UnsubscriptionAliasFlag>
</TrackUnsubscribes>

```

```
    </CreateMailing>  
</DMConnect>
```

CreateMailingFromTemplate Request

A *CreateMailingFromTemplate* request creates a message of the type specified, based upon a message template. The template must already exist. For more information about message templates, see the [DREAMmail](#) help system.

Elements in the request

The *CreateMailingFromTemplate* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *CreateMailingFromTemplate* request

Element Name	Description	Nesting
<CreateMailingFromTemplate>	Contains the information associated with a message.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>CreateMailingFromTemplate</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<TemplateName>	The template used to create the specified message.	Nest within a <i>CreateMailingFromTemplate</i> element.

TABLE: Description of the elements in a *CreateMailingFromTemplate* request

Element Name	Description	Nesting
<MailingType>	The message type to create: <ul style="list-style-type: none"> <i>StandardMailing</i> <i>NotificationTriggerMailing</i> <i>RecurringTriggerMailing</i> <i>ApprovalMailing</i>. 	Nest within a <i>CreateMailingFromTemplate</i> element.
<ClientName>	The client that contains the site in which the message is created.	Nest within a <i>CreateMailingFromTemplate</i> element.
<SiteName>	The site in which the message is created.	Nest within a <i>CreateMailingFromTemplate</i> element.
<CampaignName>	The campaign that contains the message to be created.	Nest within a <i>CreateMailingFromTemplate</i> element.
<MailingName>	The message under which the mail is sent. This name must be unique within a campaign.	Nest within a <i>CreateMailingFromTemplate</i> element.
<FromAddr>	From address for mailing.	Optional. This element must be nested within a <i>CreateMailingFromTemplate</i> .
<AliasName>	From Address for message level.	This element must be nested within FromAddr element.
<DisplayName>	Display name for From Address.	Optional. This element must be nested within a FromAddr element.
<ToList>	The set of mailing lists to which the message is sent.	Optional. Nest within a <i>CreateMailingFromTemplate</i> element. <i>Note: If the MailingType is StandardMailing or ApprovalMailing, then this is mandatory.</i>
<ListName>	A mailing list to which the message is to be sent.	Optional. Nest within a <i>ToList</i> element. You can include this more than once.
<Filter>	A SQL <i>where</i> clause that restricts the set of subscribers who receive the message.	Optional. Nest within a <i>CreateMailingFromTemplate</i> element.

TABLE: Description of the elements in a *CreateMailingFromTemplate* request

Element Name	Description	Nesting
<Suppression>	Exclude members who have received previous mailings or campaigns.	Optional. Nest within a <i>CreateMailing</i> element.
<SuppressCampaign>	Exclude members who have received previous mailings from a specified campaign.	Nest within a <i>Suppression</i> element.
<CampaignName>	Name of campaign to be suppressed.	Nest within a <i>SuppressCampaign</i> element.
<MailingName>	Name of mailing to be suppressed.	Optional. Nest within a <i>CampaignName</i> element.
<TemplateVariables>	The set of variables used in the template.	Optional. Nest within a <i>CreateMailingFromTemplate</i> element.
<Variable>	The textual substitution for the variable in the template.	Nest within a <i>TemplateVariables</i> element. You can include this more than once.
<Name>	The variable to be substituted.	Nest within a <i>Variable</i> element.
<Value>	The value of the variable.	Nest within a <i>Variable</i> element.
<ReplyToAddr>	Reply-To-Address for mailing.	Optional. This element must be nested within a <i>CreateMailingFromTemplate</i> element. Note: If using <i>TrackUnsubscribes</i> do not use <i>ReplyToAddr</i> element.
<AliasName>	Reply-To-Address for message level.	Optional. This element must be nested within <i>ReplyToAddr</i> element.
<DisplayName>	Display name for Reply_To_Address.	Optional. This element must be nested within a <i>ReplyToAddr</i> element.

TABLE: Description of the elements in a *CreateMailingFromTemplate* request

Element Name	Description	Nesting
<TrackUnsubFromReplies >	To enable or disable the Track unsubscribes from replies. <ul style="list-style-type: none"> • 0 • 1 • True • False (Case insensitive)	Optional. This element must be nested within a <i>ReplyToAddr</i> element.
<TrackUnsubscribes>	Set for specifying tracking unsubscribe flags. Leave empty to inherit default campaign values	Optional. Nest within a <i>CreateMailingFromTemplate</i> element. Note: If using <i>ReplyToAddr</i> and/or <i>SubunsubAddr</i> elements, do not use <i>TrackUnsubscribes</i> .
<RepliesFlag>	Set to 1 or true to enable track unsubscribe requests from replies. Set to 0 or false to disable.	Optional. Nest within a <i>TrackUnsubscribes</i> element.
<UnsubscriptionAliasFlag>	Set to 1 or true to enable track unsubscribe requests from unsubscription address. Set to 0 or false to disable.	Optional. Nest within a <i>TrackUnsubscribes</i> element.
<ArmMessage>	This will be contains value that will be for Arm or Open message. <ul style="list-style-type: none"> • 1 • 0 • True • False (case insensitive)	Optional. This element must be nested within a <i>CreateMailingFromTemplate</i> element and is only used for Notification Triggered Messages.
< SubunsubAddr>	It includes Sub, Unsub and TrackUnsub element tags.	Optional. This element must be nested within a <i>CreateMailingFromTemplate</i> element.

TABLE: Description of the elements in a *CreateMailingFromTemplate* request

Element Name	Description	Nesting
< SubscribeAlias>	Subscription alias at message level. If not specified default will be taken from campaign level.	Optional. This element must be nested within <i>SubunsubAddr</i> element.
< UnsubscribeAlias >	Unsubscription alias at message level. If value not specified, default will be taken from campaign level.	Optional. This element must be nested within <i>SubunsubAddr</i> element.
< TrackUnsub >	To enable or disable the Track unsubscribes from Unsub address. If value not specified, default will be taken from campaign level. <ul style="list-style-type: none"> • 0 • 1 • True • False (Case insensitive)	Optional. This element must be nested within <i>SubunsubAddr</i> element.

Example of a CreateMailing FromTemplate request:

```

<DMConnect>
  <CreateMailingFromTemplate>
    <AcknowledgementsTo>
      <EmailAddress>deployment@Epsilon.com</EmailAddress>
      <Option>9</Option>
    </AcknowledgementsTo>
    <TemplateName>UKTips2002</TemplateName>
    <MailingType>StandardMailing</MailingType>
    <ClientName>DREAMmailTipster</ClientName>
    <SiteName>DMTips</SiteName>
    <CampaignName>MarchTips</CampaignName>
    <MailingName>DMTips_03162002</MailingName>
    <FromAddr>
      <AliasName>Alias_Name</AliasName>
      <DisplayName>Name</DisplayName>
    </FromAddr>
    <ToList>

```

```
<ListName>DMTipsList</ListName>
<ListName>DMTipsAccepters</ListName>
</ToList>
<Suppression>
  <SuppressCampaign>
    <CampaignName>suppress_campaign_1</CampaignName>
    <MailingName>suppress_mailing_1</MailingName>
    <MailingName>suppress_mailing_2</MailingName>
  </SuppressCampaign>
  <SuppressCampaign>
    <CampaignName>suppress_campaign_2</CampaignName>
  </SuppressCampaign>
</Suppression>
<TemplateVariables>
  <Variable>
    <Name>Tip_Heading</Name>
    <Value>Gold Skyrockets</Value>
  </Variable>
  <Variable>
    <Name>Tip_Contents</Name>
    <Value>
      Today, the price of gold rose by 10% in international markets.
    </Value>
  </Variable>
</TemplateVariables>
<ReplyToAddr>
  <AliasName>ReplyToAddress_Name</AliasName>
  <DisplayName>Name</DisplayName>
  <TrackUnsubFromReplies>0</TrackUnsubFromReplies>
</ReplyToAddr>
<TrackUnsubscribes>
  <RepliesFlag>1</RepliesFlag>
  <UnsubscriptionAliasFlag>1</UnsubscriptionAliasFlag>
</TrackUnsubscribes>
<SubunsubAddr>
  <SubscribeAlias>None</SubscribeAlias>
  <UnsubscribeAlias>Unsub_addr</UnsubscribeAlias>
  <TrackUnsub>true</TrackUnsub>
</SubunsubAddr>
</CreateMailingFromTemplate>
</DMConnect>
```

CreateRTMMailing

A *CreateRTMMailing* request creates a new *RTM* message of the type that you specify. For more information about *RTM* and a description of related elements, see the *Real-Time Messaging Guide*.

ExportReport Request

An *ExportReport* request creates a report and saves it in the *My Data* directory. You can access these reports using *DREAMmail* or *DARTmailConnect*.

Elements in the request

The *ExportReport* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in an *ExportReport* request

Element Name	Description	Nesting
<ExportReport>	Contains the information for the report to be exported.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	Email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within an <i>ExportReport</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this element more than once.
<Option>	<p>Use this element to specify the type of acknowledgement required. The accepted values are:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you set this to 0, you still must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.

TABLE: Description of the elements in an *ExportReport* request

Element Name	Description	Nesting
<ReportType>	The report type to export: <ul style="list-style-type: none"> • <i>DeliveryResponse</i> • <i>Bouncebacks</i> • <i>Clickthroughs</i> • <i>ListActivity</i> • <i>EmailAddressChanges</i> • <i>ExportList</i> • <i>ProfileChanges</i> • <i>RTMDelivered</i> • <i>RTMNotDelivered</i> • <i>RTMClicks</i> • <i>RTMHTMLOpens</i> • <i>RTMSpotlight</i> • <i>Unsubscriptions</i> 	Nest within an <i>ExportReport</i> element.
<ExportName>	The report name.	Nest within an <i>ExportReport</i> element.
<Parameters>	The report parameters.	Nest within an <i>ExportReport</i> element.
<Parameter>	A single parameter. Note: Each report has different mandatory and optional parameters. For information about each report, see Generating Standard Reports With DMConnect .	Nest within a <i>Parameters</i> element. You can include this more than once.
<Name>	The parameter name.	Nest within a <i>Parameter</i> element.
<Value>	The parameter value.	Nest within a <i>Parameter</i> element.

For the reports covered by this request, see [Generating Standard Reports With DMConnect](#).

GetAllExportNames Request

A *GetAllExportNames* request lists the reports to which the user has access. This request only lists created and currently available reports.

Elements in the request

The *GetAllExportNames* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetAllExportNames* request

Element Name	Description	Nesting
<GetAllExportNames>	Contains information about the report to be exported.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>GetAllExportNames</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.

Example of a GetAllExportNames request:

```
<DMConnect>
  <GetAllExportNames>
    <AcknowledgementsTo>
      <EmailAddress>Bob@Epsilon.com</EmailAddress>
      <EmailAddress>Dave@Epsilon.com</EmailAddress>
      <Option>9</Option>
    </AcknowledgementsTo>
  </GetAllExportNames>
</DMConnect>
```

GetExportedReport Request

A *GetExportedReport* request retrieves the report from *MyData*.

Elements in the request

The *GetExportedReport* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetExportedReport* request

Element Name	Description	Nesting
<GetExportedReport>	Contains information about the report to be exported.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses to which the notification request is sent.	Optional. Nest within a <i>GetExportedReport</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ExportName>	The report.	Nest within a <i>GetExportedReport</i> element.

Example of a GetExported Report request:

```
<DMConnect>
  <GetExportedReport>
    <AcknowledgementsTo>
```



```
<EmailAddress>Bob@Epsilon.com</EmailAddress>  
<EmailAddress>Dave@Epsilon.com</EmailAddress>  
<Option>9</Option>  
</AcknowledgementsTo>  
<ExportName>MayClickthroughReport</ExportName>  
</GetExportedReport>  
</DMConnect>
```

GetExportReportStatus Request

A *GetExportReportStatus* request returns information about whether the specified *ExportReport* request has completed its export process.

Elements in the request

The *GetExportReportStatus* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetExportReportStatus* request

Element Name	Description	Nesting
<GetExportReportStatus>	Contains information about the report to be exported.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>GetExportReportStatus</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ExportName>	The report.	Nest within a <i>GetExportReportStatus</i> element.

Example of a GetReportStatus request:

```
<DMConnect>
  <GetExportReportStatus>
    <AcknowledgementsTo>
```

```
<EmailAddress>Bob@Epsilon.com</EmailAddress>  
<EmailAddress>Dave@Epsilon.com</EmailAddress>  
<Option>9</Option>  
</AcknowledgementsTo>  
<ExportName>MayClickthroughReport</ExportName>  
</GetExportReportStatus>  
</DMConnect>
```

GetFtpImportStatus Request

A *GetFtpImportStatus* request retrieves the status of a single import initiated by the [TriggerFtpImport Request](#).

Elements in the request

The *GetFtpImportStatus* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetFtpImportStatus* request

Element Name	Description	Nesting
<GetFtpImportStatus>	Contains the information for querying the status of a single import.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification by email when the request succeeds or fails.	Optional. Nest within a <i>GetFtpImportStatus</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client.	Nest within a <i>GetFtpImportStatus</i> element.
<SiteName>	The site within the specified client.	Nest within a <i>GetFtpImportStatus</i> element.

TABLE: Description of the elements in a *GetFtpImportStatus* request

Element Name	Description	Nesting
<ImportName>	A user-assigned name for the import job. The must be unique for the client and site. This must match the <i>ImportName</i> element in the corresponding TriggerFtpImport Request .	Nest within a <i>GetFtpImportStatus</i> element.

Example of a *GetFtpImportStatus* request:

```
<DMConnect>
  <GetFtpImportStatus>
    <ClientName>Ensemble</ClientName>
    <SiteName>Site1</SiteName>
    <ImportName>Import1</ImportName>
  </GetFtpImportStatus>
</DMConnect>
```

GetFtpImportStatusRs Response

The *GetFtpImportStatusRs* response contains the output from the [GetFtpImportStatus Request](#).

Elements in the response

The *GetFtpImportStatusRs* response can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetFtpImportStatusRs* response

Element Name	Description	Nesting
<GetFtpImportStatusRs>	Contains the information returned by a <i>GetFtpImportStatus</i> request.	This is nested within a <i>DMResponse</i> element.
<ClientName>	The client.	This is nested within a <i>GetFtpImportStatusRs</i> element.
<SiteName>	The site within the specified client.	This is nested within a <i>GetFtpImportStatusRs</i> element.
<ImportName>	The FTP import that is being queried.	This is nested within a <i>GetFtpImportStatusRs</i> element.
<ImportState>	The current state of the import process: <ul style="list-style-type: none">• 1 — a running (in-progress) import• 2 — a paused import• 3 — a completed import.	This is nested within a <i>GetFtpImportStatusRs</i> element.
<Code>	The success or failure of the import. Zero (0) is a successful import. Otherwise, a user input validation error, security error, importing error or internal error code is returned. If failure occurs during the data hygiene phase, a code of 1309 is returned. (For error codes, see Types of DMConnect response messages .) The <i>Code</i> element is only displayed if <i>ImportState</i> equals 3.	This is nested within a <i>GetFtpImportStatusRs</i> element.

TABLE: Description of the elements in a *GetFtpImportStatusRs* response

Element Name	Description	Nesting
<Description>	Further information in case a FTP import failed.	Optional. Nest within a <i>GetFtpImportStatusRs</i> element.
<FinishedTime>	The import finish date and time. Format: yyyy-mm-dd hh:mm:ss. This is only displayed if <i>ImportState</i> equals 3.	Optional. Nest within a <i>GetFtpImportStatusRs</i> element.
<Imported>	The number of rows (records) imported. If <i>ImportState</i> equals 1, then this is the current number of imported rows. During the data hygiene phase, <i>Imported</i> equals 0.	Nest within a <i>GetFtpImportStatusRs</i> element.
<Rejected>	The number of rows (records) rejected. If <i>ImportState</i> equals 1, then this is the current number of rejected rows. <i>Note: Because rejection of records now occurs during the data hygiene phase, this number equals 0.</i>	Nest within a <i>GetFtpImportStatusRs</i> element.

Example of a *GetFtpImportStatusRs* response:

```

<DMResponse>
  <Code>1</Code>
  <Description>Your request was successfully received and
processed.</Description>
  <ResultData>
    <Explanation> Here is the response to your request </
Explanation>
    <GetFtpImportStatusRs>
      <ClientName>Ensemble</ClientName>
      <SiteName>Site1</SiteName>
      <ImportName>Import1</ImportName>
      <ImportState>3</ImportState>
      <Code>0</Code>
      <FinishedTime>2003-03-24 16:53:12</FinishedTime>
      <Imported>10000000</Imported>
      <Rejected>0</Rejected>
    </GetFtpImportStatusRs>
  </ResultData>
</DMResponse>

```

GetHygieneRuleNames Request

The *GetHygieneRuleNames* request returns the names of hygiene rules that have been created for a specific site.

Elements in the request

The *GetHygieneRuleNames* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetHygieneRuleNames* request

Element Name	Description	Nesting
<ClientName>	The client.	Nest within a <i>GetHygieneRuleNames</i> element.
<SiteName>	The site within the specified client.	Nest within a <i>GetHygieneRuleNames</i> element.

Example of a *GetHygieneRuleNames* request:

```
<DMConnect>
  <GetHygieneRuleNames>
    <ClientName>Zones</ClientName>
    <SiteName>Zones</SiteName>
  </GetHygieneRuleNames>
</DMConnect>
```


GetHygieneRuleNamesRs Response

The *GetHygieneRuleNamesRs* response contains the output of the [GetHygieneRuleNames Request](#).

Elements in the response

The *GetHygieneRuleNamesRs* response can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetHygieneRuleNamesRs* response

Element Name	Description	Nesting
<ClientName>	The client.	Nest within a <i>GetHygieneRuleNamesRs</i> element.
<SiteName>	The site within the specified client.	Nest within a <i>GetHygieneRuleNamesRs</i> element.
<HygieneRuleName>	The names of the hygiene rules for the site.	Nest within a <i>GetHygieneRuleNamesRs</i> element.

Example of a *GetHygieneRuleNamesRs* response:

```
<DMConnect>
  <GetHygieneRuleNamesRs>
    <ClientName>Zones</ClientName>
    <SiteName>Zones</SiteName>
    <HygieneRuleName>Region1_rule_1</HygieneRuleName>
    <HygieneRuleName>Region1_rule_2</HygieneRuleName>
    <HygieneRuleName>Region2_rule_1</HygieneRuleName>
  </GetHygieneRuleNamesRs>
</DMConnect>
```

GetJobStatus Request

A *GetJobStatus* request returns the status of any job in *DREAMmail*.

Elements in the request

The *GetJobStatus* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetJobStatus* request

Element Name	Description	Nesting
<GetJobStatus>	Contains information about the report to be exported.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>GetJobStatus</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<JobName>	The job for which you want the status.	Nest within a <i>GetJobStatus</i> element.

Example of a GetJobStatus request:

```
<DMConnect>
  <GetJobStatus>
    <AcknowledgementsTo>
      <EmailAddress>Bob@Epsilon.com</EmailAddress>
      <EmailAddress>Dave@Epsilon.com</EmailAddress>
      <Option>9</Option>
    </AcknowledgementsTo>
    <JobName>March_List_Imports</JobName>
  </GetJobStatus>
</DMConnect>
```

GetListNames Request

A *GetListNames* request returns the names of the mailing lists and list attributes for a specific site.

Elements in the request

The *GetListNames* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetListNames* request

Element Name	Description	Nesting
<GetListNames>	Contains the information for obtaining the mailing list names for a specific site.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification by email when the request succeeds or fails.	Optional. Nest within a <i>GetListNames</i> element. You can include this more than once.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
ClientName	The client.	Nest within a <i>GetListNames</i> element.
SiteName	The site within the specified client.	Nest within a <i>GetListNames</i> element.

Example of a *GetListNames* request:

```
<DMConnect>
  <GetListNames>
    <ClientName>Client1</ClientName>
    <SiteName>Site1</SiteName>
  </GetListNames>
</DMConnect>
```

GetListNamesRs Response

The *GetListNamesRs* response contains the output from the [GetTestMailingStatus Request](#).

Elements in the response

The *GetListNamesRs* response can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetListNamesRs* response

Element Name	Description	Nesting
<GetListNamesRs>	Contains the information returned by a <i>GetListName</i> request.	This is nested within a <i>DMResponse</i> element.
<ListName>	The list. This is repeated for every list under the client and site specified in the <i>GetListName</i> request. If this element is not present, then there are no lists under the specified site.	This is nested within a <i>GetListNamesRs</i> element.
<ListType>	Contains the type of the list. Type can be: <ul style="list-style-type: none">• Standard• Approval• Proof• Approval and Proof• Global• Suppression	This is nested within a <i>GetListNamesRs</i> element.
<ListSize>	Contains the total number of members in a list.	This is nested within a <i>GetListNamesRs</i> element.
<Mailable>	Contains the counts of the members in the list which are mailable.	This is nested within a <i>GetListNamesRs</i> element.
<Text>	Contains the count of text enabled members in the list which are mailable.	This is nested within a <i>Mailable</i> element.
<HTML>	Contains the count of HTML enabled members in the list which are mailable.	This is nested within a <i>Mailable</i> element.

TABLE: Description of the elements in a *GetListNamesRs* response

Element Name	Description	Nesting
<AOL>	Contains the count of AOL enabled members in the list which are mailable.	This is nested within a <i>Mailable</i> element.
<Total>	Contains the count of all members in the list which are mailable. (It is addition of Text + HTML + AOL + Unknown)	This is nested within a <i>Mailable</i> element.
<NonMailable>	Contains the counts of the members in the list which are not mailable.	This is nested within a <i>GetListNamesRs</i> element.
<Invalid>	Contains the count of invalid members in the list	This is nested within a <i>NonMailable</i> element.
<Unconfirmed>	Contains the count of unconfirmed members in the list	This is nested within a <i>NonMailable</i> element.
<Unknown>	Contains the count of unknown members in the list	This is nested within a <i>GetListNamesRs</i> element.

Example of a *GetListNamesRs* response:

```

<DMResponse><Code>1</Code><Description>Your request was successfully
received and processed.</Description><ResultData><Explanation> Here
is the response to your request </Explanation>
  <GetListNamesRs>
    <List>
      <ListName>100k</ListName>
      <ListType>Standard</ListType>
      <ListSize>100000</ListSize>
      <Mailable>
        <Text>1</Text>
        <HTML>0</HTML>
        <AOL>0</AOL>
        <Total>100000</Total>
      </Mailable>
      <NonMailable>
        <Invalid>0</Invalid>
        <Unconfirmed>0</Unconfirmed>
      </NonMailable>
      <Unknown>99999</Unknown>
    </List>
    <List>
      <ListName>PA</ListName>
      <ListType>Approval and Proof</ListType>

```

```
<ListSize>9</ListSize>
<Mailable>
  <Text>1</Text>
  <HTML>6</HTML>
  <AOL>0</AOL>
  <Total>7</Total>
</Mailable>
<NonMailable>
  <Invalid>2</Invalid>
  <Unconfirmed>0</Unconfirmed>
</NonMailable>
<Unknown>0</Unknown>
</List>  <List>
  <ListName>PerList_Sup</ListName>
  <ListType>Suppression</ListType>
  <ListSize>0</ListSize>
  <Mailable>
    <Text>0</Text>
    <HTML>0</HTML>
    <AOL>0</AOL>
    <Total>0</Total>
  </Mailable>
  <NonMailable>
    <Invalid>0</Invalid>
    <Unconfirmed>0</Unconfirmed>
  </NonMailable>
  <Unknown>0</Unknown>
</List>
</GetListNamesRs>
</ResultData></DMResponse>
```


GetMailingContent Request

A *GetMailingContent* request returns the content for a specific message.

Elements in the request

The *GetMailingContent* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetMailingContent* request

Element Name	Description	Nesting
<GetMailingContent>	Returns the content for a specific message.	Nest within a <i>DMConnect</i> element.
<ClientName>	The client that contains the message.	Nest within a <i>GetMailingContent</i> element.
<SiteName>	The site that contains the message.	Nest within a <i>GetMailingContent</i> element.
<CampaignName>	The campaign that contains the message.	Nest within a <i>GetMailingContent</i> element.
<MailingName>	The message that contains the content you want to obtain.	Nest within a <i>GetMailingContent</i> element.

Example of a *GetMailingContent* request:

```
<DMConnect>
  <GetMailingContent>
    <ClientName>Client1</ClientName>
    <SiteName>Site2</SiteName>
    <CampaignName>Campaign_04162005</CampaignName>
    <MailingName>mailing1</MailingName>
  </GetMailingContent>
</DMConnect>
```

GetMailingContentRs Response

A *GetMailingContentRs* response contains the output of the *GetMailingContent* request.

Elements in the response

The *GetMailingContentRs* response can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element is included.

TABLE: Description of the elements in a *GetMailingContentRs* request

Element Name	Description	Nesting
<GetMailingContentRs>	Contains the output returned by a <i>GetMailingContent</i> request.	Nested within a <i>DMResponse</i> element
<Subject>	The subject line of the message.	Nested within a <i>GetMailingContentRs</i> element.
<Language>	The character set of the message content. The default character set is English (US-ASCII).	Optional. Nested within a <i>GetMailingContentRs</i> element.
<AOLContent>	The AOL content of the message. This field is empty if no AOL content was included in the message.	Optional. Nested within a <i>GetMailingContentRs</i> element.
<HTMLContent>	The HTML content of the message. This field is empty if no HTML content was included in the message.	Optional. Nested within a <i>GetMailingContentRs</i> element.
<TextContent>	The text content of the message. This field is empty if no text content was included in the message.	Optional. Nested within a <i>GetMailingContentRs</i> element.

Example of a *GetMailingContentRs* response:

```
<DMResponse>
  <GetMailingContentRs>
    <Subject>Spring Tips</Subject>
    <HTMLContent>
      <BodyText>
        <![CDATA[<HTML><body>
          Dear #FirstName#,<br>
```

```
        These are your Hot DREAMmail Tips for March 16 in html
        format.<br>
        <a href="http://www.DMtips.com">Click Here for the first
        tip</a><br>
        <a href="http://www.DMtips2.com">Click Here for the second
        tip</a><br>
        Thanks,<br>
        The DREAMmailTipster</body></HTML>
    ]]>
</BodyText>
</HTMLContent>
</GetMailingContentRs>
</DMResponse>
```

GetMailingStatus Request

A *GetMailingStatus* request returns the status of a specific message.

Elements in the request

The *GetMailingStatus* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetMailingStatus* request

Element Name	Description	Nesting
<GetMailingStatus>	Contains the information for obtaining the message status.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>GetMailingStatus</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client.	Nest within a <i>GetMailingStatus</i> element.
<SiteName>	The site within the specified client.	Nest within a <i>GetMailingStatus</i> element.
<CampaignName>	The mailing campaign within the specified client and site.	Nest within a <i>GetMailingStatus</i> element.
<MailingName>	The message.	Nest within a <i>GetMailingStatus</i> element.

Example of a GetMailingStatus request:

```
<DMConnect>
  <GetMailingStatus>
    <ClientName>CompanyABC</ClientName>
    <SiteName>Region1</SiteName>
    <CampaignName>TestCampaign</CampaignName>
    <MailingName>SimpleTest4</MailingName>
  </GetMailingStatus>
</DMConnect>
```

GetMailingStatusRs Response

The *GetMailingStatusRs* response contains the output from the [GetMailingStatus Request](#).

Elements in the response

The *GetMailingStatusRs* response can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetMailingStatusRs* response

Element Name	Description	Nesting
<GetMailingStatusRs>	Contains the information returned by a <i>GetMailingStatus</i> request.	Nested within a <i>DMResponse</i> element.
<ClientName>	The client.	Nested within a <i>GetMailingStatusRs</i> element.
<SiteName>	The site within the specified client.	Nested within a <i>GetMailingStatusRs</i> element.
<CampaignName>	The mailing campaign within the specified client and site.	Nested within a <i>GetMailingStatusRs</i> element.
<MailingName>	The message.	Nested within a <i>GetMailingStatusRs</i> element.

TABLE: Description of the elements in a *GetMailingStatusRs* response

Element Name	Description	Nesting
<MailingState>	<p>The specified mailing state. Use these values for standard and approval mailings, unless specified otherwise:</p> <ul style="list-style-type: none">• 5 — Closed (trigger only)• 4 — Armed (trigger only)• 3 — Approval pending (approval only)• 2 — Composing (approval and trigger only)• 1 — Running (sending)• 0 — Ready-to-send• -1 — Finished sending• -2 — Aborted• -3 — Paused• -4 — Error• -5 — Blocked• -6 — Archived• -7 — Archiving• -8 — Archiving Errors	Nested within a <i>GetMailingStatusRs</i> element.
<RecipientsTotal>	Total number of subscribers that this message is targeting.	Nested within a <i>GetMailingStatusRs</i> element.
<MailsPushed>	Total number of emails sent, regardless of whether they are delivered. This is lower than <i>RecipientsTotal</i> until the mailing is finished, at which time the numbers are equal.	Nested within a <i>GetMailingStatusRs</i> element.

TABLE: Description of the elements in a *GetMailingStatusRs* response

Element Name	Description	Nesting
<Code>	<p>The success or failure of the message deployment. Zero (0) is successful deployment. Otherwise, a user input validation error, security error, mailing deployment, or internal error code is returned. (For list or error codes, see Types of DmConnect response messages.)</p> <p>This is only displayed only for the following states:</p> <ul style="list-style-type: none"> Finished Error Archiving Error Abort. 	Optional. Nested within a <i>GetMailingStatusRs</i> element.
<Description>	Further information in case a mailing fails to be sent.	Optional. Nested within a <i>GetMailingStatusRs</i> element.
<FinishedTime>	<p>The date and time when the mailing finished. Format: yyyy-mm-dd hh:mm:ss.</p> <p>This is only displayed if the <i>MailingStatus</i> element equals -1.</p>	Optional. If this is included in the response, the <i>ErrorReport</i> element is not included. Nested within a <i>GetMailingStatusRs</i> element.
<ErrorReport>	A composite element that consists of the <i>Type</i> , <i>Name</i> , <i>Format</i> , and <i>DetailDescription</i> elements.	Optional. This is only included in the response if the <i>GetMailingStatus</i> request is unsuccessful. Nested within a <i>GetMailingStatusRs</i> element.
<Type>	Type of <i>DREAMmail</i> entity in which the message was detected (for example, message).	Optional. Nested within an <i>ErrorReport</i> element.
<Name>	<i>DREAMmail</i> entity in which the message was detected (for example, message).	Optional. Nested within an <i>ErrorReport</i> element.
<Format>	Format of the message.	Optional. Nested within an <i>ErrorReport</i> element.

TABLE: Description of the elements in a *GetMailingStatusRs* response

Element Name	Description	Nesting
<DetailDescription>	Sample line of content from the message where the error was detected.	Optional. Nested within an <i>ErrorReport</i> element.

Example of a *GetMailingStatusRs* response:

```
<DMResponse>
  <Code>1</Code>
  <Description>Your request was successfully received and
processed.</Description>
  <ResultData>
    <Explanation> Here is the response to your request </
Explanation>
    <GetMailngStatusRs>
      <ClientName>CompanyABC</ClientName>
      <SiteName>Site1</SiteName>
      <CampaignName>TestCampaign</CampaignName>
      <MailingName>SimpleTest4</MailingName>
      <MailingState>-1</MailingState>
      <RecipientsTotal>25</RecipientsTotal>
      <MailsPushed>25</MailsPushed>
      <Code>0</Code>
      <FinishedTime>2003-07-25 17:03:51</FinishedTime>
    </GetMailngStatusRs>
  </ResultData>
</DMResponse>
```

GetMessageSize Request

The *GetMessageSize* request returns the size of a specific message.

Elements in the request

The *GetMessageSize* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetMessageSize* request

Element Name	Description	Nesting
<GetMessageSize>	Contains the information for obtaining the size of a specific message.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>GetMessageSize</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client.	Nest within a <i>GetMessageSize</i> element.
<SiteName>	The site within the specified client.	Nest within a <i>GetMessageSize</i> element.
<CampaignName>	The campaign that contains the message.	Nest within a <i>GetMessageSize</i> element.

TABLE: Description of the elements in a *GetMessageSize* request

Element Name	Description	Nesting
<MailingName>	The message. The name must be unique within the specified campaign.	Nest within a <i>GetMessageSize</i> element.

Example of a GetMessageSize request:

```
<DMConnect>
  <GetMessageSize>
    <AcknowledgementsTo>
      <EmailAddress>yjo@dme.Epsilon.net</EmailAddress>
    </AcknowledgementsTo>
    <ClientName>Zones</ClientName>
    <SiteName>Zones</SiteName>
    <CampaignName>yjTestCampaign</CampaignName>
    <MailingName>wrongProfileField_message</MailingName>
  </GetMessageSize>
</DMConnect>
```

GetMessageSizeRs Response

A *GetMessageSizeRs* response contains the output from the [GetMessageSize Request](#).

Elements in the response

The *GetMessageSizeRs* response can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetMessageSizeRs* response

Element Name	Description	Nesting
<GetMessageSizeRs>	The information returned by a <i>GetMessageSize</i> request.	Nested within a <i>DMResponse</i> element.
<ClientName>	The client specified in the <i>GetMessageSize</i> request.	Nested within a <i>GetMessageSizeRs</i> element.
<SiteName>	The site specified in the <i>GetMessageSize</i> request.	Nested within a <i>GetMessageSizeRs</i> element.
<CampaignName>	The campaign specified in the <i>GetMessageSize</i> request.	Nested within a <i>GetMessageSize</i> element.
<MailingName>	The message specified in the <i>GetMessageSize</i> request.	Nested within a <i>GetMessageSizeRs</i> element.
<TextSize>	Text content size in kilobytes. Only the integer is displayed.	Nested within a <i>GetMessageSizeRs</i> element.
<AOLSize>	AOL content size in kilobytes. Only the integer is displayed.	Nested within a <i>GetMessageSizeRs</i> element.
<HTMLSize>	HTML content size in kilobytes. Only the integer is displayed.	Nested within a <i>GetMessageSizeRs</i> element.

Example of a GetMessageSizeRs response:

```
<DMResponse>
  <Code>1</Code>
  <Description>Your request was successfully received and processed.</Description>
  <ResultData>
    <Explanation> Here is the response to your request </Explanation>
    <GetMailingSizeRs>
      <ClientName>Zones</ClientName>
      <SiteName>Zones</SiteName>
      <CampaignName>yjTestCampaign</CampaignName>
      <MessageName>wrongProfileField_message</MessageName>
      <TextSize>0.010</TextSize>
```

```
        <AOLSize>0.000</AOLSize>
        <HTMLSize>0.000</HTMLSize>
    </GetMailngSizeRs>
</ResultData>
</DMResponse>
```

GetObjectNames Request

A *GetObjectNames* request returns the names of *DREAMmail* objects such as clients, sites, campaigns, messages, and mailing lists.

Elements in the request

The *GetObjectNames* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetObjectNames* request

Element Name	Description	Nesting
<GetObjectNames>	Contains the information for querying object names in <i>DREAMmail</i> .	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>GetObjectNames</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Optional. Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.

TABLE: Description of the elements in a *GetObjectNames* request

Element Name	Description	Nesting
<Depth>	<p>Depth of the DREAMmail hierarchy to return:</p> <ul style="list-style-type: none"> • <i>Client</i>: only returns client names. • <i>Site</i>: only returns client and site names. • <i>Campaign</i>: returns client, site, campaign, and template names. • <i>Mailing</i>: returns client, site, campaign, template, and mailing names. • <i>Addresses</i>: returns the Alias and Display Names for From and Reply-To addresses at the lowest hierarchical level of the <ClientName>, <SiteName>, or <CampaignName>. • <i>ContentList</i>: returns all content segment Names and Type at the lowest hierarchical level of the <ClientName> or <SiteName> (for <CampaignName> as well) element. <p>Note: For this request, a user's viewing privileges must match the level of depth that the user specifies.</p> <p>If this element is not present, <i>Depth</i> is determined by the scope of the element that indicates the hierarchical level of the object: <ClientName>, <SiteName>, or <CampaignName>.</p>	Optional. Nest within a <i>GetObjectNames</i> element.
<ClientName>	The client. If this is not included, <i>DMConnect</i> returns all client names to which the user has access.	Optional. Nest within a <i>GetObjectNames</i> element.

TABLE: Description of the elements in a *GetObjectNames* request

Element Name	Description	Nesting
<SiteName>	The site within the client. Only include this if you have included <i>ClientName</i> in the request. If this is not included, <i>DMConnect</i> returns all sites the user can access under the specified client.	Optional. Nest within a <i>GetObjectNames</i> element.
<CampaignName>	The mailing campaign within the specified client and site. Only include this if you have included <i>Sitename</i> in the request. If this element is not included, <i>DMConnect</i> returns all campaigns under the specified site.	Optional. Nest within a <i>GetObjectNames</i> element.

Example of a *GetObjectNames* request:

```
<DMConnect>
  <GetObjectNames>
    <Depth>CAMPAIGN</Depth>
    <ClientName>CompanyABC</ClientName>
  </GetObjectNames>
</DMConnect>
```


GetObjectNamesRs Response

The *GetObjectNamesRs* response contains the output from the [GetObjectNames Request](#).

Elements in the response

The *GetObjectNamesRs* response can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetObjectNamesRs* response

Element Name	Description	Nesting
<GetObjectNamesRs>	Contains the information returned by a <i>GetObjectNames</i> request.	Nested within a <i>DMResponse</i> element.
<ClientTree>	Composite element that contains a hierarchy of objects under a client. <i>DREAMmail</i> returns one instance for every client the user can access. If this element is not present, the user cannot access any clients on this server.	Nested within a <i>GetObjectNamesRs</i> element.
<ClientName>	The client	Nested within a <i>ClientTree</i> element.
<SiteTree>	Composite element that contains a hierarchy of objects under a site. <i>DREAMmail</i> returns one instance for every site the user can access within the specified client. If this is not present, the user cannot access any sites within the client.	Nested within a <i>ClientTree</i> element.
<SiteName>	Site name.	Nested within a <i>SiteTree</i> element.
<CampaignTree>	Composite element that contains a hierarchy of objects under a campaign. <i>DREAMmail</i> returns one instance for every campaign within the site. If this element is not present, no campaigns exist in the specified site.	Nested within a <i>SiteTree</i> element.
<CampaignName>	Campaign name.	Nested within a <i>CampaignTree</i> element.

TABLE: Description of the elements in a *GetObjectNamesRs* response

Element Name	Description	Nesting
<MailingName>	Mailings names for the specified campaign. If this is not present there are no mailings for the campaign when the <i>GetObjectNames</i> request was submitted.	Nested within a <i>CampaignTree</i> element.
<Mailing Type>	The type of mailing. This is returned with every <i>MailingName</i> element. <i>Types</i> include: <ul style="list-style-type: none"> • <i>StandardMailing</i> • <i>ApprovalMailing</i> • <i>NotificationTriggerMailing</i> • <i>RecurringTriggerMailing</i> Note: Recurring triggers can only be sent to recipients already in the DREAMmail database, whereas notification mailings can be sent to users who are not in the DREAMmail database.	Nested within a <i>CampaignTree</i> element.
<TemplateName>	Message names in the specific site. This is zero (0) if no templates have been created yet.	Nested within a <i>SiteTree</i> element.
<ContentTree>	Contains the information of the each Content separately. This tag is repeated for every content in a Site.	This is nested within <i>SiteTree</i> element.
<ContentName>	Contains the name of the Content.	This is nested within <i>ContentTree</i> element.
<ContentType>	Contains the type of the Content.	This is nested within <i>ContentTree</i> element.
<FromAddress>	Contains the name of FromAddress separately. This tag is repeated for every FromAddress in the Site corresponding to the Client/Site/Campaign level.	This is nested within <i>SiteTree</i> element.

TABLE: Description of the elements in a *GetObjectNamesRs* response

Element Name	Description	Nesting
<ReplyToAddress>	Contains the name of ReplyToAddress separately. This tag is repeated for every ReplyToAddress in the Site corresponding to the Client/Site/Campaign level.	This is nested within <i>SiteTree</i> element.
<AliasName>	Contains the alias name (without the domain for FromAddress)	This is nested within a <i>FromAddress</i> or <i>ReplyToAddress</i> element.
<DisplayName>	The display name for the alias	This is nested within a <i>FromAddress</i> or <i>ReplyToAddress</i> element.

Example of a *GetObjectNamesRs* response:

```

<DMResponse>
  <Code>1</Code>
  <Description>Your request was successfully received and
processed.</Description>
  <ResultData>
    <Explanation> Here is the response to your request </
Explanation>
    <GetObjectNamesRs>
      <ClientTree>
        <ClientName>CompanyABC</ClientName>
        <SiteTree>
          <SiteName>Region1</SiteName>
          <CampaignTree>
            <CampaignName>ClcConfirmationsCampaign</
CampaignName>
          </CampaignTree>
          <CampaignTree>
            <CampaignName>TestCampaign</CampaignName>
          </CampaignTree>
          <CampaignTree>
            <CampaignName>TrainingCampaign</CampaignName>
          </CampaignTree>
          <TemplateName>TestTemplate</TemplateName>
        </SiteTree>
        <SiteTree>
          <SiteName>Region2</SiteName>
          <CampaignTree>
            <CampaignName>ClcConfirmationsCampaign</
CampaignName>

```

```
        </CampaignTree>
    </SiteTree>
    <SiteTree>
        <SiteName>Region3</SiteName>
        <CampaignTree>
            <CampaignName>ClcConfirmationsCampaign</
CampaignName>
        </CampaignTree>
        <CampaignTree>
            <CampaignName>SaTestMailing</CampaignName>
        </CampaignTree>
        <CampaignTree>
            <CampaignName>TestCamp2</CampaignName>
        </CampaignTree>
    </SiteTree>
</ClientTree>
</GetObjectNamesRs>
</ResultData>
</DMResponse>
```

GetProfileFields Request

A *GetProfileFields* request lists the system and user-defined profile fields at the client, site or email address level. Examples include *EmailAddress*, *FirstName*, *LastName*, and user-defined profile fields. The following table contains the rules for specifying client and site names in the request.

TABLE: Rules for specifying level in the request

Rule	Result
Specify <i>Clientname</i> element only	Request returns all profile fields for the client and for all of the sites under the client.
Specify <i>Clientname</i> and <i>Sitename</i> elements	Request returns all profile fields for the client and for the specified site.
Specify <i>Clientname</i> ; include the <i>SiteName</i> element but leave it blank	Request returns only the profile fields for the client.
Specify <i>Clientname</i> , <i>Sitename</i> and <i>EmailAddress</i> elements	Request returns only the profile fields for the specified email address.

Elements in the request

The *GetProfileFields* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetProfileFields* request

Element Name	Description	Nesting
<GetProfileFields>	Contains the information about retrieving profile fields from an email address, site or client.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses to which the notification message is sent.	Optional. Nest within a <i>GetProfileFields</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.

TABLE: Description of the elements in a *GetProfileFields* request

Element Name	Description	Nesting
<Option>	The acknowledgement type that you want to receive: <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client whose profile fields are being queried.	Nest within a <i>GetProfileFields</i> element.
<SiteName>	The site whose profile fields are being queried. If the request only specifies the client name, it returns all profile fields for the client and for all of the sites under the client	Optional. Nest within a <i>GetProfileFields</i> element.
<EmailAddress>	The email address whose profile fields are being queried.	Optional. Nest within a <i>GetProfileFields</i> element.

Example of a *GetProfileFields* request:

```
<DMConnect>
  <GetProfileFields>
    <ClientName>CompanyABC</ClientName>
    <SiteName>Region1</SiteName>
  </GetProfileFields>
</DMConnect>
```

GetProfileFieldsRs Response

The *GetProfileFieldsRs* response contains the output from the [GetProfileFields Request](#).

Elements in the response

The *GetProfileFieldsRs* response can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetProfileFieldsRs* response

Element Name	Description	Nesting
<GetProfileFieldsRs>	Contains the information returned by a <i>GetProfileFields</i> request.	Nested within a <i>DM Response</i> element.
<ClientFields>	A composite element that consists of the <i>ClientName</i> and <i>FieldDfn</i> elements.	Nested within a <i>GetProfileFieldsRs</i> element.
<ClientName>	The client whose profile fields are being queried.	Nested within a <i>ClientFields</i> element. You can include this more than once.
<FieldDfn>	A composite element that contains the <i>Name</i> , <i>FieldType</i> , and <i>Length</i> elements.	Nested within a <i>ClientFields</i> element.
<Name>	Profile field name.	Nested within a <i>FieldDfn</i> element.
<Value>	Profile field value. This value is only returned for EmailAddress level.	Nested within a <i>FieldDfn</i> element.
<FieldType>	The data type of the profile field: <ul style="list-style-type: none">• <i>DATE</i> — format: yyyy-mm-dd• <i>INT</i> — integer• <i>STRING</i> — a string of characters of variable length.	Nested within a <i>FieldDfn</i> element.
<Length>	Profile field length. Only used if the profile field type is <i>STRING</i> .	Nested within a <i>FieldDfn</i> element. This is mandatory when <i>FieldType</i> is <i>STRING</i> , otherwise it is not included.

TABLE: Description of the elements in a *GetProfileFieldsRs* response

Element Name	Description	Nesting
<SiteFields>	A composite element that consists of the <i>SiteName</i> and <i>FieldDfn</i> elements.	Optional. Nested within a <i>GetProfileFieldsRs</i> element.
<SiteName>	The site whose profile fields are being queried.	Nested within a <i>SiteFields</i> element.
<FieldDfn>	Same as <FieldDfn> explained above.	Nested within a <i>SiteFields</i> element.
<Subscribed_Lists>	Contains all subscribed lists for each site. This tag is repeated for every Site coming from request. Only returned for EmailAddress level.	This is nested within a <i>SiteFields</i> element.
<UnSubscribed_Lists>	Contains all Unsubscribed lists for each site. This tag is repeated for every Site coming from request. Only returned for EmailAddress level.	This is nested within a <i>SiteFields</i> element.
<ListName>	Contains the Name of individual list. This tag is repeated for every Subscribed_Lists and Un Subscribed_Lists tags for every site. Only returned for EmailAddress level.	This is nested within a Subscribed_Lists and UnSubscribed_Lists element.

GetProfileFieldsRs response:

```

<DMResponse>
  <Code>1</Code>
  <Description>Your request was successfully received and processed.</Description>
  <ResultData>
    <Explanation> Here is the response to your request </Explanation>
    <GetProfileFieldsRs>
      <ClientFields>
        <ClientName>CompanyABC</ClientName>
        <FieldDfn>
          <Name>EmailAddr</Name>
          <FieldType>STRING</FieldType>
          <Length>255</Length>
        </FieldDfn>
        <FieldDfn>

```



```
<Name>FirstName</Name>
<FieldType>STRING</FieldType>
<Length>255</Length>
</FieldDfn>
<FieldDfn>
  <Name>LastName</Name>
  <FieldType>STRING</FieldType>
  <Length>255</Length>
</FieldDfn>
<FieldDfn>
  <Name>Platform</Name>
  <FieldType>STRING</FieldType>
  <Length>255</Length>
</FieldDfn>
<FieldDfn>
  <Name>ClientUnsubscribe</Name>
  <FieldType>INT</FieldType>
</FieldDfn>
<FieldDfn>
  <Name>ValidEmailAddr</Name>
  <FieldType>INT</FieldType>
</FieldDfn>
<FieldDfn>
  <Name>HtmlCapable</Name>
  <FieldType>INT</FieldType>
</FieldDfn>
<FieldDfn>
  <Name>ClcStatus</Name>
  <FieldType>INT</FieldType>
</FieldDfn>
</ClientFields>
<SiteFields>
  <SiteName>Site1</SiteName>
  <FieldDfn>
    <Name>ShoeSize</Name>
    <FieldType>INT</FieldType>
  </FieldDfn>
  <FieldDfn>
    <Name>ShoeColor</Name>
    <FieldType>STRING</FieldType>
    <Length>400</Length>
  </FieldDfn>
  <FieldDfn>
    <Name>PurchaseDate</Name>
    <FieldType>DATE</FieldType>
  </FieldDfn>
</SiteFields>
```

```
        </GetProfileFieldsRs>
      </ResultData>
</DMResponse>
```

GetTemplateVariables Request

A *GetTemplateVariables* request returns the variables of a specific template.

Elements in the request

The *GetTemplateVariables* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetTemplateVariables* request

Element Name	Description	Nesting
<GetTemplateVariables>	Contains the information associated with a content segment.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification by email when the request succeeds or fails.	Optional. Nest within a <i>GetTemplateVariables</i> element. You can include this more than once.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client that contains the template.	Nest within a <i>GetTemplateVariables</i> element.
<SiteName>	The site that contains the template.	Nest within a <i>GetTemplateVariables</i> element.

TABLE: Description of the elements in a *GetTemplateVariables* request

Element Name	Description	Nesting
<TemplateName>	The template that contains the variables that you want to obtain.	Nest within a <i>GetTemplateVariables</i> element.

Example of a *GetTemplateVariables* request:

```
<DMConnect>
  <GetTemplateVariables>
    <AcknowledgementsTo>
      <EmailAddress>deployment@Epsilon.com</EmailAddress>
      <Option>0</Option>
    </AcknowledgementsTo>
    <ClientName>client1</ClientName>
    <SiteName>site3</SiteName>
    <TemplateName>Template_eastern</TemplateName>
  </GetTemplateVariables>
</DMConnect>
```

GetTemplateVariablesRs Response

The *GetTemplateVariablesRs* response contains the output from the [GetTemplateVariables Request](#).

Elements in the response

The *GetTemplateVariablesRs* response can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetTemplateVariablesRs* response

Element Name	Description	Nesting
<GetTemplateVariablesRs>	Contains the information returned by a <i>GetTemplateVariables</i> request.	Nested within a <i>DM Response</i> element.
<TemplateVariableAndDefaults>	Contains nested pairs of template variable names and default values.	Nested within a <i>GetTemplateVariablesRs</i> element.
<TemplateVariable>	Contains a template variable name and a default value.	Nested within a <i>TemplateVariableAndDefaults</i> element.
<VariableName>	Name of the variable in the template.	Nested within a <i>TemplateVariable</i> element.
<DefaultValue>	The default value of the variable.	Nested within a <i>TemplateVariable</i> element.

Example of a *GetTemplateVariablesRs* response:

```
<DMResponse>
  </GetTemplateVariablesRs>
    <TemplateVariablesAndDefaults>
      <TemplateVariable>
        <VariableName>NYC_greeting</VariableName>
        <DefaultValue>Dear New York Resident,</DefaultValue>
      </TemplateVariable>
      <TemplateVariable>
        <VariableName>Boston_greeting</VariableName>
        <DefaultValue>Dear Boston Resident,</DefaultValue>
      </TemplateVariable>
    </TemplateVariablesAndDefaults>
  </GetTemplateVariablesRs>
</DMResponse>
```

GetTestMailingStatus Request

A *GetTestMailingStatus* request returns the status of a specific test message.

Elements in the request

The *GetTestMailingStatus* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetTestMailingStatus* request

Element Name	Description	Nesting
<GetTestMailingStatus>	Contains the information for checking the test message status.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>GetTestMailingStatus</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client.	Nest within a <i>GetTestMailingStatus</i> element.
<SiteName>	The site within the specified client.	Nest within a <i>GetTestMailingStatus</i> element.
<CampaignName>	The mailing campaign within the specified client and site.	Nest within a <i>GetTestMailingStatus</i> element.

TABLE: Description of the elements in a *GetTestMailingStatus* request

Element Name	Description	Nesting
<MailingName>	The message.	Nest within a <i>GetTestMailingStatus</i> element.

Example of a *GetTestMailingStatus* request:

```
<DMConnect>
  <GetTestMailingStatus>
    <AcknowledgementsTo>
      <EmailAddress>yjo@dme.Epsilon.net</EmailAddress>
    </AcknowledgementsTo>
    <ClientName>Zones</ClientName>
    <SiteName>Zones</SiteName>
    <CampaignName>yjTestCampaign</CampaignName>
    <MailingName>proofMessage_fixError_Proof_agai</MailingName>
  </GetTestMailingStatus>
</DMConnect>
```

GetTestMailingStatusRs Response

The *GetTestMailingStatusRs* response contains the output from the [GetTestMailingStatus Request](#).

Elements in the response

The *GetTestMailingStatusRs* response can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *GetTestMailingStatusRs* response

Element Name	Description	Nesting
<GetTestMailingStatusRs>	Information returned by a <i>GetTestMailingStatus</i> request.	Nested within a <i>DMResponse</i> element.
<ClientName>	The client.	Nested within a <i>GetMailingStatusRs</i> element.
<SiteName>	The site within the specified client.	Nested within a <i>GetMailingStatusRs</i> element.
<CampaignName>	The mailing campaign within the specified client and site.	Nested within a <i>GetMailingStatusRs</i> element.
<MailingName>	The message.	Nested within a <i>GetMailingStatusRs</i> element.
<MailsPushed>	Total number of emails sent, regardless of whether they are delivered. This is lower than <i>RecipientsTotal</i> until the mailing is finished, at which time the numbers are equal.	Nested within a <i>GetMailingStatusRs</i> element.

TABLE: Description of the elements in a *GetTestMailingStatusRs* response

Element Name	Description	Nesting
<Code>	<p>Success or failure of the deployment of the message. Zero (0) is a successful deployment. Otherwise, a user input validation error, security error, mailing deployment, or internal error code is returned. (For a list of error codes, see Types of DMConnect response messages.)</p> <p>This element is only displayed for the following states:</p> <ul style="list-style-type: none"> • Finished • Error • Archiving • Error • Abort. 	Optional. Nested within a <i>GetMailingStatusRs</i> element.
<Description>	Further information in case a test mailing fails to be sent.	Optional. Nested within a <i>GetMailingStatusRs</i> element.
<ErrorReport>	A composite element that consists of the <i>Type</i> , <i>Name</i> , <i>Format</i> , and <i>DetailDescription</i> elements.	<p>Optional. This is only included in the response if the <i>GetTestMailingStatus</i> request is unsuccessful.</p> <p>Nested within a <i>GetTestMailingStatusRs</i> element.</p>
<Type>	Type of <i>DREAMmail</i> entity in which the message was detected (for example, message).	Optional. Nested within an <i>ErrorReport</i> element.
<Name>	<i>DREAMmail</i> entity in which the message was detected (for example, message).	Optional. Nested within an <i>ErrorReport</i> element.
<Format>	Format of the message.	Optional. Nested within an <i>ErrorReport</i> element.
<DetailDescription>	Sample line of content from the message where the error was detected.	Optional. Nested within an <i>ErrorReport</i> element.

Example of a GetTestMailingStatusRs response:

```
<DMResponse>
  <Code>1</Code>
  <Description>Your request was successfully received and processed.</
Description>
  <ResultData>
    <Explanation> Here is the response to your request </Explanation>
    <GetTestMailngStatusRs>
      <ClientName>Zones</ClientName>
      <SiteName>Zones</SiteName>
      <CampaignName>yjTestCampaign</CampaignName>
      <MailingName>proofMessage_fixError_Proof_agai</MailingName>
      <Code>2487</Code>
      <Description>Wrong Operator for Switch statement: "</
Description>
      <ErrorReport>
        <Type>Message</Type>
        <Name>proofMessage_fixError_Proof_agai</Name>
        <Format>HTML</Format>
        <DetailDescription><![CDATA[<Switch type="string"
value=EmailAddr">    ]]></DetailDescription>
      </ErrorReport>
    </GetTestMailngStatusRs>
  </ResultData>
</DMResponse>
```

GetTrackedURL Request

A *GetTrackedURL* request retrieves all tracked URLs in a message.

Elements in the request

The *GetTrackedURL* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains which elements must be included.

TABLE: Description of the elements of a *GetTrackedURL* request

Element Name	Description	Nesting
<GetTrackedURL>	Contains the information associated with the tracked URLs within a message.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails	Optional. Nest within a <i>GetTrackedURL</i> element.
<EmailAddress>	The email addresses that receive a notification when the request succeeds or fails.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	The acknowledgement type that you want to receive: <ul style="list-style-type: none">• 0 – no notifications• 1 – only send confirmation acknowledgements• 2 – only send failure acknowledgements• 9 – send all acknowledgements <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client that contains the site where you want to retrieve the tracked URLs of a message.	Nest within a <i>GetTrackedURL</i> element.
<SiteName>	The site within the specified client.	Nest within a <i>GetTrackedURL</i> element.
<CampaignName>	The campaign that contains the message.	Nest within a <i>GetTrackedURL</i> element.

Element Name	Description	Nesting
<MailingName>	The message containing the tracked URLs you want to obtain.	Nest within a <i>GetTrackedURL</i> element.

Example of a GetTrackedURL request:

```
<DMConnect>
  <GetTrackedURL>
    <AcknowledgementsTo><EmailAddress>user@yourcompany.com</
    EmailAddress>
    </AcknowledgementsTo>
    <ClientName>Client1</ClientName>
    <SiteName>Site1</SiteName>
    <CampaignName>Campaign1</CampaignName>
    <MailingName>Message1</MailingName>
  </GetTrackedURL>
</DMConnect>
```

GetTrackedURLRs Response

A *GetTrackedURLRs* response contains the output from the [GetTrackedURL Request](#).

Elements in the response

The *GetTrackedURLRs* response can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains which elements must be included.

TABLE: Description of the elements of a *GetTrackedURLRs* response

Element Name	Description	Nesting
<GetTrackedURLRs>	Contains the information returned by a GetTrackedURL request.	Nested within a <i>DMResponse</i> element.
<URLInfo>	The URLs tracked within the message. Each URL is identified with its own <URLInfo></URLInfo> element.	Nested within a <i>GetTrackedURLRs</i> element.
<URLName>	The tracked URL in the message; for example: http://www.fallriversports.com	Nested within a <i>URLInfo</i> element.
<URLVariant>	The variant (personalization) of the URL.	Nested within a <i>URLInfo</i> element.
<URLLabel>	The label (if applicable) given to the URL.	Nest within a <i>URLInfo</i> element.
<URLRevision>	The revision number of the URL.	Nested within a <i>URLInfo</i> element.
<URLFormat>	The body formats (Text, AOL, HTML) of the message where the URL is located.	Nested within a <i>URLInfo</i> element.
<URLLocation>	The locations of the URL in the message (Content, Offer or Body).	Nested within a <i>URLInfo</i> element.

GetTrackedURLRs response:

```
<DMResponse>
  <Code>1</Code>
  <Description>Your request was successfully received and processed.</
Description>
  <ResultData>
    <Explanation>Here is the response to your request </Explanation>
    <GetTrackedURLRs>
      <URLInfo>
        <URLName>http://www.fallriversports.com</URLName>
        <URLVariant></URLVariant>
        <URLLabel>Home Page</URLLabel>
        <URLRevision>1</URLRevision>
        <URLFormat>TEXT</URLFormat>
        <URLLocation>OFFER</URLLocation>
      </URLInfo></GetTrackedURLRs>
    </ResultData>
  </DMConnect>
```

ImportList Request

An *ImportList* request adds a set of subscribers and their profile data to a list.

Elements in the request

The *ImportList* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *ImportList* request

Element Name	Description	Nesting
<ImportList>	Contains the information for the import to be performed.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification by email when the request succeeds or fails.	Optional. Nest within an <i>ImportList</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<JobName>	<p>The job for which you want the status.</p> <p>If you do not specify a JobName, a numeric JobName is returned with the receipt message.</p>	Optional. Nest within a <i>ImportList</i> element.
<ClientName>	The client that contains the customer profiles to be updated.	Nest within an <i>ImportList</i> element.

TABLE: Description of the elements in a *ImportList* request

Element Name	Description	Nesting
<SiteName>	The site that contains the list to be updated.	Nest within an <i>ImportList</i> element.
<ListName>	The list to which the imported subscriber profiles are added.	Nest within an <i>ImportList</i> element.
<ImportData>	The profile data to be updated.	Nest within an <i>ImportList</i> element.
<Row> or <R>	The data for a single row. The number of rows cannot be greater than the maximum specified in configuration.	Nest within an <i>ImportData</i> element. You can include this more than once.
<MemberIdentifier>	The email address of the subscriber to be updated. <i>Note: If the row includes <C> <Name> EmailAddr</Name> <Value> ... </Value> </C>, the value in the Column element changes the subscriber's email address.</i>	Nest within a <i>Row</i> element. you can only include one instance of this element per row element.
<Column> or <C>	The name and value for a single column within a row. The number of columns cannot be greater than the maximum specified in configuration.	Nest within a <i>Row</i> element. You can include this more than once.
<Name>	The column name.	Nest within a <i>Column</i> element.
<Value>	The value of the column.	Nest within a <i>Column</i> element.

Example of a *ImportList* request:

```

<DMConnect>
  <ImportList>
    <AcknowledgementsTo>
      <EmailAddress>deployment@Epsilon.com</EmailAddress>
      <EmailAddress>hot_tip_editor@Epsilon.com</EmailAddress>
      <Option>9</Option>
    </AcknowledgementsTo>
    <ClientName>DREAMmailTipster</ClientName>
    <SiteName>DMTips</SiteName>
    <ListName>DMAcceptors</ListName>
  </ImportList>
</DMConnect>

```



```
<ImportData>
  <Row>
    <MemberIdentifier>Kim@Tantex.com</MemberIdentifier>
    <Column>
      <Name>FavColor</Name>
      <Value>Blue</Value>
    </Column>
    <Column>
      <Name>Sex</Name>
      <Value>Male</Value>
    </Column>
  </Row>
  <Row>
    <MemberIdentifier><![CDATA[s.o'neill@tantex.com]]></MemberIdentifier>
    <Column>
      <Name>FavColor</Name>
      <Value>Green</Value>
    </Column>
    <Column>
      <Name>Sex</Name>
      <Value>Male</Value>
    </Column>
  </Row>
</ImportData>
</ImportList>
</DMConnect>
```

ManageLists Request

A *ManageLists* request will enable Activation/Deactivation of lists for a particular client and site.

Elements in the request

The *ManageLists* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *ManageLists* request

Element Name	Description	Nesting
<ManageLists>	Contains the information for managing the list statuses.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>ManageLists</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client that contains the site that contains the list.	Nest within a <i>ManageLists</i> element.
<SiteName>	The site that contains the campaign that contains the list.	Nest within a <i>ManageLists</i> element.
<ActivateLists>	It includes list names to make activate.	Optional. This element must be nested within a <i>ManageLists</i> element.

TABLE: Description of the elements in a *ManageLists* request

Element Name	Description	Nesting
<ListName>	The name of the list to activate or deactivate.	This element must be nested within an <i>ActivateLists</i> or <i>DeactivateLists</i> element.
<DeactivateLists>	It includes list names to make inactive.	Optional. This element must be nested within a <i>ManageLists</i> element.

Example of a *ManageLists* request:

```
<DMConnect>
  <ManageLists>
    <ClientName>MyClient</ClientName>
    <SiteName>MySite</SiteName>
    <ActivateLists>
      <ListName>mylist1</ListName>
      <ListName>mylist2</ListName>
    </ActivateLists>
    <DeactivateLists>
      <ListName>mylist3</ListName>
      <ListName>mylist4</ListName>
    </DeactivateLists>
  </ManageLists>
</DMConnect>
```

SendMailingForApproval Request

A *SendingMailForApproval* request sends a specified message out for approval to the site's approval list. For information about approving messages, see the [DREAMmail](#) help system.

Elements in the request

The *SendingMailForApproval* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *SendMailingForApproval* request

Element Name	Description	Nesting
<SendMailingForApproval>	Contains the information for the message to be sent.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>SendMailingForApproval</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client that contains the site that contains the message.	Nest within a <i>SendMailingForApproval</i> element.
<SiteName>	The site that contains the campaign that contains the message.	Nest within a <i>SendMailingForApproval</i> element.

TABLE: Description of the elements in a *SendMailingForApproval* request

Element Name	Description	Nesting
<CampaignName>	The campaign that contains the message.	Nest within a <i>SendMailingForApproval</i> element.
<MailingName>	The message under which the mail is sent. This name must be unique within a campaign.	Nest within a <i>SendMailingForApproval</i> element.

Example of a *SendMailingForApproval* request:

```
<DMConnect>
  <SendMailingForApproval>
    <AcknowledgementsTo>
      <EmailAddress>deployment@Epsilon.com</EmailAddress>
      <EmailAddress>hot_tip_editor@Epsilon.com</EmailAddress>
      <Option>9</Option>
    </AcknowledgementsTo>
    <ClientName>DREAMmailTipster</ClientName>
    <SiteName>DMTips</SiteName>
    <CampaignName>MarchTips</CampaignName>
    <MailingName>DMTips_03162002</MailingName>
  </SendMailingForApproval>
</DMConnect>
```

StartMailing Request

A *StartMailing* request sends a specified message.

Note: You can only use this request for messages specified as either *StandardMailing* or *ApprovalMailing*.

Elements in the request

The *StartMailing* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *StartMailing* request

Element Name	Description	Nesting
<StartMailing>	Contains the information for the message to be sent.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>StartMailing</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	The acknowledgement type that you want to receive: <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must still include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client that contains the site that contains the message.	Nest within a <i>StartMailing</i> element.
<SiteName>	The site that contains the campaign that contains the message.	Nest within a <i>StartMailing</i> element.

TABLE: Description of the elements in a *StartMailing* request

Element Name	Description	Nesting
<CampaignName>	The campaign that contains the message.	Nest within a <i>StartMailing</i> element.
<MailingName>	The message under which the mail is sent. This name must be unique within a campaign.	Nest within a <i>StartMailing</i> element.
<SendRate>	<p>The rate at which messages are sent:</p> <ul style="list-style-type: none"> • 0 — ASAP (over 1,000,000 emails per hour) • 1 — 500,000 emails per hour • 2 — 250,000 emails per hour • 3 — 100,000 emails per hour • 4 — 75,000 emails per hour • 5 — 50,000 emails per hour • 6 — 25,000 emails per hour • 7 — 15,000 emails per hour • 8 — 4,000 emails per hour • 9 — 1,000 emails per hour. 	Nest within a <i>StartMailing</i> element.

Example of a *StartMailing* request:

```

<DMConnect>
  <StartMailing>
    <AcknowledgementsTo>
      <EmailAddress>deployment@Epsilon.com</EmailAddress>
      <EmailAddress>hot_tip_editor@Epsilon.com</EmailAddress>
      <Option>9</Option>
    </AcknowledgementsTo>
    <ClientName>DREAMmailTipster</ClientName>
    <SiteName>DMTips</SiteName>
    <CampaignName>MarchTips</CampaignName>
    <MailingName>DMTips_03162002</MailingName>
    <SendRate>3</SendRate>
  </StartMailing>
</DMConnect>

```

TestMailing Request

A *TestMailing* request sends a specific message to the email addresses on the test list. You can use this request for both the validation phase, where you check a message for grammatical and technical errors, and the test phase, where you send the message to a randomly selected group of subscribers to test the response to a message. For more information, see the [DREAMmail](#) help system.

Elements in the request

The *TestMailing* request can contain the elements listed in this table, in hierarchical order. The *Nesting* column explains the hierarchy in which you should include the element.

TABLE: Description of the elements in a *TestMailing* request

Element Name	Description	Nesting
<TestMailing>	Contains all the information for the message to be tested.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>TestMailing</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The type of acknowledgement that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: If you chose 0 as an option, you still must include content in the <EmailAddress> element in your request.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client that contains the site that contains the message.	Nest within a <i>TestMailing</i> element.

TABLE: Description of the elements in a *TestMailing* request

Element Name	Description	Nesting
<SiteName>	The site that contains the campaign that contains the message.	Nest within a <i>TestMailing</i> element.
<CampaignName>	The campaign that contains the message.	Nest within a <i>TestMailing</i> element.
<MailingName>	The message under which the mail is sent. This name must be unique within a campaign.	Nest within a <i>TestMailing</i> element.
<TestList>	The list to which the message is sent.	Nest within a <i>TestMailing</i> element.
<ProofFormat>	Leave empty to send all proof formats.	Optional. Nest within a <i>TestMailing</i> element.
<Format>	The format for which you want to receive a proof: Can be ALL, AOL, HTML, or Text This is case sensitive.	Optional. Nest within a <i>ProofFormat</i> element.

Example of a *TestMailing* request:

```

<DMConnect>
  <TestMailing>
    <AcknowledgementsTo>
      <EmailAddress>deployment@Epsilon.com</EmailAddress>
      <EmailAddress>hot_tip_editor@Epsilon.com</EmailAddress>
      <Option>9</Option>
    </AcknowledgementsTo>
    <ClientName>DREAMmailTipster</ClientName>
    <SiteName>DMTips</SiteName>
    <CampaignName>MarchTips</CampaignName>
    <MailingName>DMTips_03162002</MailingName>
    <ProofFormat>
      <Format>ALL</Format>
      <Format>AOL</Format>
      <Format>HTML</Format>
      <Format>Text</Format>
    </ProofFormat>
    <TestList>DMTipsTestList</TestList>
  </TestMailing>
</DMConnect>

```

TriggerFtpImport Request

A *TriggerFtpImport* request initiates a single or “one-time” FTP import. This request supplies parameters required by the import process, such as the structure of the input file, the file where the rejected records are stored, the hygiene rule to be applied to the file, the mapping between the input file columns, and [DREAMmail](#) profile fields.

FTP Import overview

This section describes how you use *DARTmailConnect* to perform a single FTP import.

Stage	Description
1	<p>In DREAMmail, enable the <i>Enable FTP Import</i> checkbox.</p> <p>This is in the <i>Options</i> tab under <i>Site Setup > General</i>. You only need to do this once and it initiates single and recurring FTP imports. (For more information, see the DREAMmail help system.)</p> <p>If you enabled the FTP import feature in a release previous to DREAMmail 4.0, you must first disable the <i>Enable FTP Import</i> checkbox, save the change, and then re-enable the checkbox to apply the new settings.</p> <p>Note: When re-enabling the <i>Enable FTP Import</i> checkbox, enter the user name and password that you were using before the release of DREAMmail 4.0.</p>
2	<p>Using a SFTP client, connect to the SFTP file server using the user name and password that you entered in Stage 1.</p>
3	<p>After accessing the SFTP file server, upload the import file to the <i>\OneTimeFTP</i> directory.</p>
4	<p>Create and execute a XML script in <i>DARTmailConnect</i> that uses the <i>TriggerFtpImport</i> request to trigger the import.</p> <p>(See Example of a TriggerFtpImport request.)</p>
5	<p>After the script has been executed, the uploaded file is automatically moved to the <i>\OneTimeFTP\3_ToBeImported</i> folder.</p> <p>Do not move or delete files in the <i>\OneTimeFTP\3_ToBeImported</i> folder using any other external processes. Manually moving the files from the <i>\OneTimeFTP\3_ToBeImported</i> folder causes the import job to fail.</p>
6	<p>After the import job completes successfully, the original list import file is moved to the <i>\OneTimeFTP\4_PostImported</i> folder.</p> <p>If the job fails, the file is moved to the <i>5_Rejected</i> folder. The files in <i>\OneTimeFTP\3_ToBeImported</i> are automatically removed.</p>

Stage	Description
7	<p>Rejected records are written to the <code>\OneTimeFTP\6_Reports\Rejected_<SourceFileName>.<Timestamp></code> file.</p> <p>You can have rejected records even if the import job has completed successfully. A record is rejected due to an invalid email address, an incorrect format, or because the record was suppressed by the suppression file. (For more information about suppression files, see the <i>DREAMmail</i> help system.)</p> <p>Note: <i>DREAMmail</i> considers an empty row in the import file to contain an invalid email address. Therefore this record is written to the <i>Rejected</i> folder. To avoid this, do not enter a carriage return after the final record in your import file.</p>
8	<p>Records with email addresses that have been successfully corrected by a data hygiene rule are written to the <code>\OneTimeFTP\6_Reports\Corrected_<SourceFileName>.<Timestamp></code> file.</p>
9	<p>If a job fails due to errors, <i>DREAMmail</i> moves the processed import file to the <code>\OneTimeFTP\5_Rejected</code> folder where it remains for seven days.</p> <p>Use the rejected data file to determine the records that contain invalid data.</p> <p>If more than one import job uses the same rejected file name, the file will contain only the rejected records from the last import job. Therefore, you should use a different rejected file name for each import job.</p>

Elements in the request

The *TriggerFtpImport* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *TriggerFtpImport* request

Element Name	Description	Nesting
<TriggerFtpImport>	Contains the information for triggering an import.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>TriggerFtpImport</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.

TABLE: Description of the elements in a *TriggerFtplImport* request

Element Name	Description	Nesting
<Option>	<p>The type of acknowledgement that you want to receive:</p> <ul style="list-style-type: none"> • 0 — no notifications • 1 — only send confirmation acknowledgements • 2 — only send failure acknowledgements • 9 — send all acknowledgements. <p>Note: If you chose 0 as an option, you still must include content in the <EmailAddress> element in your request.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client.	Nest within a <i>TriggerFtplImport</i> element.
<SiteName>	The site within the specified client.	Nest within a <i>TriggerFtplImport</i> element.
<ImportName>	A user-assigned name for the import job. The must be unique for the client and site.	Nest within a <i>TriggerFtplImport</i> element.
<ImportAction>	<p>Contains the <i>ListName</i> element and mutually exclusive empty elements that show the import type:</p> <ul style="list-style-type: none"> • ClientUnsubscribe • SiteUnsubscribe • ListUnsubscribe • ListSubscribe. 	Nest within a <i>TriggerFtplImport</i> element.
<ListName>	List targeted by the import.	Nest within an <i>ImportAction</i> element.
<InputFile>	File to import, as it appears in the <i>OneTimeFtplImport</i> directory. (Do not include \OneTimeFtplImport\ as part of the file name.)	Nest within a <i>TriggerFtplImport</i> element.

TABLE: Description of the elements in a *TriggerFtplImport* request

Element Name	Description	Nesting
<FieldSeparator>	The separator character for the fields in a subscriber's record. This includes: <ul style="list-style-type: none"> • , (comma) • (pipe) • (double pipe) • Tab. 	Nest within a <i>TriggerFtplImport</i> element.
<FieldMap>	A composite element that defines the mapping between the columns in the input file and the profile fields.	Nest within a <i>TriggerFtplImport</i> element.
<Source>	The input column. This element is mapped to the <i>Destination</i> element.	Nest within a <i>FieldMap</i> element.
<Destination>	The <i>DREAMmail</i> profile field that the <i>Source</i> element is mapped to.	Nest within a <i>FieldMap</i> element.
<RejectsFile>	The reject file name, as it appears in the <i>OneTimeFtplImport</i> directory. (Do not include <i>\OneTimeFtplImport\</i> as part of the file name.) <i>Note: This element is no longer used as of DREAMmail 4.3.</i>	Nest within a <i>TriggerFtplImport</i> element.
<EmailAddress>	Email addresses to which notification is sent regarding the progress of the import job.	Optional. Nest within a <i>TriggerFtplImport</i> element.
<ApplyHygieneRule>	Name of the hygiene rule to be applied to the input file.	Optional. Nest within a <i>TriggerFtplImport</i> element
<SuppressionFile>	Name of the file used to purge records from the input file.	Optional. Nest within a <i>TriggerFtplImport</i> element

Example of a *TriggerFtplImport* request:

```

<DMConnect>
  <TriggerFtplImport>
    <ClientName>CompanyABC</ClientName>
    <SiteName>Region1</SiteName>
    <ImportName>ImportOne</ImportName>
    <ImportAction>
      <ListUnsubscribe/>

```

```
        <ListName>TestList</ListName>
    </ImportAction>
    <InputFile>input.csv</InputFile>
    <FieldSeparator>|</FieldSeparator>
    <FieldMap>
        <Source>Email</Source>
        <Destination>EmailAddress</Destination>
    </FieldMap>
    <ApplyHygieneRule>Region1_rule_2</ApplyHygieneRule>
</TriggerFtpImport>
</DMConnect>
```

TriggerMailingToEmailAddresses Request

A *TriggerMailingToEmailAddresses* request sends a triggered message to selected email addresses using either a DREAMmail From Address or a Referral From Address. A Referral From Address allows you to send a message that appears to be coming from another person, rather than your company or brand, and is commonly used with Forward-to-a-Friend or agent model solutions.

Note: The *TriggerMailingToEmailAddresses* request only works on a notification or recurring triggered message. For more information, see the *DREAMmail* help system.

The *TriggerMailingToList* request uses the same set of variables for an entire list, while the *TriggerMailingToEmailAddresses* uses a specific set of variables for each address. If you send a number of trigger messages from different lists using the same variables, it is more efficient to create a new list and use the *TriggerMailingToList* request, rather than the *TriggerMailingToEmailAddresses* request.

Important Considerations for using a Referral From Address: Using the Referral From Address to modify the From Address to one that does not use the mailing domain set up in DREAMmail may impact your deliverability. Most ISPs use authentication techniques to verify that a message received is from the address and domain indicated in the message header. When these do not match, there is a risk that the message may be caught in the recipient's junk box or filtered by the ISP.

In addition, privacy issues exist when using a Referral From Address. From a CAN-SPAM perspective, it is considered best practice that the message, when looked at as a whole, is not "deceptive." Therefore, when using a Referral From Address, the subject line of the message should be clear and the message should clearly indicate it has been forwarded by a friend or sent by an agent.

Note: The Referral From Address allows 64 characters for the alias and 253 characters for the domain, for a total maximum length of 255 characters.

Referral Addresses are validated on submission through DMConnect to ensure proper syntax compliance, including alias, @ sign, domain and top level domain (alias@domain.com). Outbound validation is not performed for messages using the Referral From address. The From Address provided via the API post will not be validated against any internal list, including the:

1. Mail Transfer Agent Global Suppression List
2. Mail Transfer Agent Invalids

Note: Bounces resulting from messages sent using a Referral From Address are not reported on in DREAMmail. Any bounces or replies resulting from the message will be sent to the address indicated in the From Address.

Elements in the request

The *TriggerMailingToEmailAddresses* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *TriggerMailingToEmailAddresses* request

Element Name	Description	Nesting
<TriggerMailingToEmailAddresses>	Contains the information for the message request to be triggered to a set of lists.	Nest within a <i>DMConnect</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>TriggerMailingToEmailAddresses</i> element.
<Option>	<p>The type of acknowledgement that you want to receive:</p> <ul style="list-style-type: none"> • 0 — no notifications • 1 — only send confirmation acknowledgements • 2 — only send failure acknowledgements • 9 — send all acknowledgements. <p>Note: If you chose 0 as an option, you still must include content in the <EmailAddress> element in your request.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client that contains the trigger message.	Nest within a <i>TriggerMailingToEmailAddresses</i> element.

TABLE: Description of the elements in a *TriggerMailingToEmailAddresses* request

Element Name	Description	Nesting
<SiteName>	The site that contains the trigger message.	Nest within a <i>TriggerMailingToEmailAddresses</i> element.
<CampaignName>	The campaign that contains the trigger message.	Nest within a <i>TriggerMailingToEmailAddresses</i> element.
<MailingName>	The trigger message. This name must be unique within a campaign.	Nest within a <i>TriggerMailingToEmailAddresses</i> element.
<ReferralAddr>	Allows user to specify any From address to be used in the deployment. <i>Note: If the "Track unsubscriptions from replies" option is selected for the specified message, the ReferralAddr may not be used.</i>	Optional. Nest within a <i>TriggerMailingToEmailAddresses</i> element.
<TriggerFromAddr>	The From Address value for the message level. This value does not have to match the existing alias or domain. <i>Note: The TriggerFromAddr value overrides the From Address and Reply-To Address (if selected) values set within the message.</i>	Nest within a <i>ReferralAddr</i> element.
<DisplayName>	Display name for the From Address.	Optional. Nest within a <i>ReferralAddr</i> element.
<ToEmailAddresses>	The addresses to which the message is triggered.	Nest within a <i>TriggerMailingToEmailAddresses</i> element.
<TriggerEmailAddress>	The information required to send the message to a single email address.	Nest within a <i>ToEmailAddresses</i> element. You can include this more than once.
<EmailAddress>	An email address to which a message is triggered.	Nest within a <i>TriggerEmailAddress</i> element.

TABLE: Description of the elements in a *TriggerMailingToEmailAddresses* request

Element Name	Description	Nesting
<EventVariables>	The event variables that have a value defined for the trigger, which is to be applied to the email address identified in the <i>EmailAddress</i> element. Note: When triggering messages to multiple email address, event variables must be specified in the same order for all email address.	Optional. Nest within a <i>TriggerEmailAddress</i> element. Note: Event variables must not contain other event variables or tags for other event variables. Messages with this nesting syntax will result in errors.
<Variable>	One of the event variables for the email address identified in the <i>EmailAddress</i> element.	Nest within an <i>EventVariables</i> element. You can include this more than once.
<Name>	The event variable.	Nest within a <i>Variable</i> element.
<Value>	The value of the event variable. This can contain a script content segment. For more information about script content segments, see the <i>Advanced Messaging Guide</i> . Note: Content in a language other than English must be placed in a CDATA section or be escaped. For more information, see Special Characters and Non-English Text .	Nest within a <i>Variable</i> element.
<TrackedUrls>	The URLs tracked for the email address identified within the <i>EmailAddress</i> element. Each URL is identified with its own <URL></URL> element.	Optional. Nest within a <i>TriggerEmailAddress</i> element.
<URL>	The URL that is tracked.	Nest within a <i>TrackedUrls</i> element. You can include this more than once.
<URLName>	The name of the URL to track; for example, http://www.somedomain.com.	Nest within a <i>URL</i> element.

TABLE: Description of the elements in a *TriggerMailingToEmailAddresses* request

Element Name	Description	Nesting
<URLLabel>	Assigns a label to the URL to track.	Optional. Nest within a <i>URL</i> element.

Example of a *TriggerMailingToEmailAddresses* request:

```

<DMConnect>
  <TriggerMailingToEmailAddresses>
    <AcknowledgementsTo>
      <EmailAddress>deployment@Epsilon.com</EmailAddress>
      <EmailAddress>hot_tip_editor@Epsilon.com</EmailAddress>
      <Option>9</Option>
    </AcknowledgementsTo>
    <ClientName>DREAMmailTipster</ClientName>
    <SiteName>DMTips</SiteName>
    <CampaignName>Stock_Alert</CampaignName>
    <MailingName>Stock_Alert01122002</MailingName>
    <ReferralAddr>
      <TriggerFromAddr>Jsmith@smithco.com</TriggerFromAddr>
      <DisplayName>John Smith</DisplayName>
    </ReferralAddr>
    <ToEmailAddresses>
      <TriggerEmailAddress>
        <EmailAddress>Rick@Tantex.com</EmailAddress>
        <EventVariables>
          <Variable>
            <Name>Gold_Stocks</Name>
            <Value><![CDATA[<HTML><body>
              Gold Prices Skyrocket<br>
              <a href="http://www.DMtips7.com">Click here for details</a><br>
              <body><HTML>]]>
            </Value>
          </Variable>
        </EventVariables>
        <TrackedUrls>
          <URL>
            <URLName>http://www.DMtips7.com</URLName>
            <URLLabel>Gold Tips</URLLabel>
          </URL>
        </TrackedUrls>
      </TriggerEmailAddress>
      <TriggerEmailAddress>
        <EmailAddress>Rob@Bobc.com</EmailAddress>
        <EventVariables>
          <Variable>

```

```
<Name>Tech_Stocks</Name>
<Value><![CDATA[<HTML><body>
Tech stocks rebound!<br>
<a href = "http:www.DMtips10.com">Click here for details</a>
<br><body><HTML>]]>
</Value>
</Variable>
</EventVariables>
<TrackedUrls>
<URL>
<URLName>http://www.DMtips10.com</URLName>
<URLLabel>Tech Tips</URLLabel>
</URL>
</TrackedUrls>
</TriggerEmailAddress>
</ToEmailAddresses>
</TriggerMailingToEmailAddresses>
</DMConnect>
```

TriggerMailingToList Request

A *TriggerMailingToList* request sends an armed, or active, trigger message to a set of mailing lists using either a DREAMmail From Address or a Referral From Address. A Referral From Address allows you to send a message that appears to be coming from another person, rather than your company or brand, and is commonly used with Forward-to-a-Friend or agent model solutions.

Note: The *TriggerMailingToList* request works only on notification or recurring triggered messages. For more information, see the [DREAMmail help system](#).

Important: The *TriggerMailingToList* request uses the same set of variables for an entire list, while the *TriggerMailingToEmailAddresses* uses a specific set of variables for each address. If you are sending out a number of triggered messages from different lists using the same variables, it is more efficient to create a new list and use the *TriggerMailingToList* request, rather than the *TriggerMailingToEmailAddresses* request.

Important Considerations for using a Referral From Address: Using the Referral From Address to modify the From Address to one that does not use the mailing domain set up in DREAMmail may impact your deliverability. Most ISPs use authentication techniques to verify that a message received is from the address and domain indicated in the message header. When these do not match, there is a risk that the message may be caught in the recipient's junk box or filtered by the ISP.

In addition, privacy issues exist when using a Referral From Address. From a CAN-SPAM perspective, it is considered best practice that the message, when looked at as a whole, is not "deceptive." Therefore, when using a Referral From Address, the subject line of the message should be clear and the message should clearly indicate it has been forwarded by a friend or sent by an agent.

Note: The Referral From Address allows 64 characters for the alias and 253 characters for the domain, for a total maximum length of 255 characters.

Referral Addresses are validated on submission through DMConnect to ensure proper syntax compliance, including alias, @ sign, domain and top level domain (alias@domain.com). Outbound validation is not performed for messages using the Referral From address. The From Address provided via the API post will not be validated against any internal list, including the:

1. Mail Transfer Agent Global Suppression List
2. Mail Transfer Agent Invalids

Note: Bounces resulting from messages sent using a Referral From Address are not reported on in DREAMmail. Any bounces or replies resulting from the message will be sent to the address indicated in the From Address.

Elements in the request

The *TriggerMailingToList* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *TriggerMailingToList* request

Element Name	Description	Nesting
<TriggerMailingToList>	Contains the information for the message request to be triggered to a set of lists.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses to which the notification message is sent.	Optional. Nest within a <i>TriggerMailingToList</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The type of acknowledgement that you want to receive:</p> <ul style="list-style-type: none"> • 0 — no notifications • 1 — only send confirmation acknowledgements • 2 — only send failure acknowledgements • 9 — send all acknowledgements. <p>Note: If you chose 0 as an option, you still must include content in the <EmailAddress> element in your request.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client that contains the trigger message.	Nest within a <i>TriggerMailingToList</i> element.

TABLE: Description of the elements in a *TriggerMailingToList* request

Element Name	Description	Nesting
<SiteName>	The site that contains the trigger message.	Nest within a <i>TriggerMailingToList</i> element.
<CampaignName>	The campaign that contains the trigger message.	Nest within a <i>TriggerMailingToList</i> element.
<MailingName>	The trigger message. This name must be unique within a campaign.	Nest within a <i>TriggerMailingToList</i> element.
<ReferralAddr>	Allows user to specify any From address to be used in the deployment. Note: If the "Track unsubscriptions from replies" option is selected for the specified message, the ReferralAddr may not be used.	Optional. Nest within a <i>TriggerMailingToList</i> element.
<TriggerFromAddr>	The From Address value for the message level. This value does not have to match the existing alias or domain. Note: The TriggerFromAddr value overrides the From Address and Reply-To Address (if selected) values set within the message.	Nest within a <i>ReferralAddr</i> element.
<DisplayName>	Display name for the From Address.	Optional. Nest within a <i>ReferralAddr</i> element.
<ToList>	The set of mailing lists to which the message is sent.	Nest within a <i>TriggerMailingToList</i> element.
<ListName>	A list to which the message is to be sent.	Nest within a <i>ToList</i> element. You can include this more than once.

TABLE: Description of the elements in a *TriggerMailingToList* request

Element Name	Description	Nesting
<EventVariables>	The event variables that have a value defined for the trigger.	Optional. Nest within a <i>TriggerMailingToList</i> element. Note: Event variables must not contain other event variables or tags for other event variables. Messages with this nesting syntax will result in errors.
<Variable>	Indicates one of the event variables for the message.	Nest within an <i>EventVariables</i> element. You can include this more than once.
<Name>	The event variable.	Nest within a <i>Variable</i> element.
<Value>	The value of the event variable. This can contain a script content segment. For more information about script content segments, see the <i>Advanced Messaging Guide</i> . Note: Content in a language other than English must be placed in a CDATA section or be escaped. For more information, see Special Characters and Non-English Text .	Nest within a <i>Variable</i> element.
<TrackedUrls>	The URLs that are tracked. Each URL is identified with its own <URL></URL> element.	Optional. Nest within a <i>TriggerMailingToList</i> element.
<URL>	The URL that is tracked.	Nest within a <i>TrackedUrls</i> element. You can include this more than once.
<URLName>	The name of the URL to track; for example, http://www.somedomain.com.	Nest within a <i>URL</i> element.
<URLLabel>	Assigns a label to the URL to track.	Optional. Nest within a <i>URL</i> element.

Example of a TriggerMailing ToList request:

```
<DMConnect>
  <TriggerMailingToList>
    <AcknowledgementsTo>
      <EmailAddress>deployment@Epsilon.com</EmailAddress>
      <Option>9</Option>
    </AcknowledgementsTo>
    <ClientName>DREAMmailTipster</ClientName>
    <SiteName>DMTips</SiteName>
    <CampaignName>Stock_Alert</CampaignName>
    <MailingName>Stock_Alert01122002</MailingName>
    <ReferralAddr>
      <TriggerFromAddr>jsmith@smithco.com</TriggerFromAddr>
      <DisplayName>John Smith</DisplayName>
    </ReferralAddr>
    <ToList>
      <ListName>DMTipsList</ListName>
      <ListName>DMTipsAccepters</ListName>
    </ToList>
    <EventVariables>
      <Variable>
        <Name>Gold_Stocks</Name>
        <Value><![CDATA[ <HTML><body>
          Gold Prices Skyrocket!<br>
          <a href="http://www.DMtips7.com">Click here for details
          </a><br>
          </body></HTML>]]>
        </Value>
      </Variable>
    </EventVariables>
    <TrackedUrls>
      <URL>
        <URLName>http://www.DMtips7.com</URLName>
        <URLLabel>Gold Tips</URLLabel>
      </URL>
    </TrackedUrls>
  </TriggerMailingToList>
</DMConnect>
```

Unsubscribe Request

An *Unsubscribe* request removes email addresses from mailing lists at the client, site, or list levels.

Elements in the request

The *Unsubscribe* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *Unsubscribe* request

Element Name	Description	Nesting
<Unsubscribe>	Contains the information about the unsubscriptions to be performed.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within an <i>Unsubscribe</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	The acknowledgement type that you want to receive: <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<JobName>	The job name for which you want the status. If you do not specify a JobName, a numeric JobName is returned with the receipt message.	Optional. Nest within an <i>Unsubscribe</i> element.
<ClientName>	The client.	Nest within an <i>Unsubscribe</i> element.

TABLE: Description of the elements in a *Unsubscribe* request

Element Name	Description	Nesting
<UnsubscribeData>	The subscribers to be unsubscribed.	Nest within an <i>Unsubscribe</i> element.
<UnsubscribeRequest>	The data for each unsubscription request. An <i>UnsubscribeRequest</i> must include one of: <ul style="list-style-type: none"> <i>ClientUnsubscription</i> <i>SiteUnsubscription</i> <i>ListUnsubscription</i>. 	Nest within an <i>UnsubscribeData</i> element. You can include this more than once.
<ClientUnsubscription>	Performs an unsubscription request for the client. The subscriber is removed from all sites and lists within the client.	Optional. Nest within an <i>UnsubscribeRequest</i> element.
<SiteUnsubscription>	Performs an unsubscription request for the site. The subscriber is removed from all lists associated with the site.	Optional. Nest within an <i>UnsubscribeRequest</i> element.
<ListUnsubscription>	Performs an unsubscription request for the list. The subscriber is only removed from the specific list.	Optional. Nest within an <i>UnsubscribeRequest</i> element.
<SiteName>	The site that contains the list to update or the site to update.	Nest within a <i>SiteUnsubscription</i> or <i>ListUnsubscription</i> element.
<ListName>	The list from which you want to unsubscribe the email addresses.	Nest within a <i>ListUnsubscription</i> element.
<MemberIdentifier>	The email address to unsubscribe.	Nest within a <i>ClientUnsubscription</i> , <i>SiteUnsubscription</i> , or <i>ListUnsubscription</i> element.

Example of a Unsubscribe request:

```

<DMConnect>
  <Unsubscribe>
    <AcknowledgementsTo>
      <EmailAddress>deployment@Epsilon.com</EmailAddress>
      <EmailAddress>hot_tip_editor@Epsilon.com</EmailAddress>
    <Option>9</Option>
  </Unsubscribe>
</DMConnect>

```

```
</AcknowledgementsTo>
<ClientName>DREAMmailTipster</ClientName>
<UnsubscribeData>
  <UnsubscribeRequest>
    <ClientUnsubscription>
      <MemberIdentifier>RickD@Tantex.com</MemberIdentifier>
    </ClientUnsubscription>
  </UnsubscribeRequest>
  <UnsubscribeRequest>
    <ClientUnsubscription>
      <MemberIdentifier><![CDATA[Rob.O'Neill@Bobc.com]]>
    </MemberIdentifier>
    </ClientUnsubscription>
  </UnsubscribeRequest>
  <UnsubscribeRequest>
    <SiteUnsubscription>
      <SiteName>DMTips</SiteName>
      <MemberIdentifier>Dave@Tantex.com</MemberIdentifier>
    </SiteUnsubscription>
  </UnsubscribeRequest>
  <UnsubscribeRequest>
    <SiteUnsubscription>
      <SiteName>DMTips</SiteName>
      <MemberIdentifier>Rick@Bobc.com</MemberIdentifier>
    </SiteUnsubscription>
  </UnsubscribeRequest>
  <UnsubscribeRequest>
    <ListUnsubscription>
      <SiteName>DMTips</SiteName>
      <ListName>DMTipsList</ListName>
      <MemberIdentifier>Chris@Bobc.com</MemberIdentifier>
    </ListUnsubscription>
  </UnsubscribeRequest>
</UnsubscribeData>
</Unsubscribe>
</DMConnect>
```

UpdateConditionalContent Request

An *UpdateConditionalContent* request can update new content by overwriting existing contents.

Elements in the request

The *UpdateConditionalContent* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *UpdateConditionalContent* request

Element Name	Description	Nesting
<UpdateConditionalContent>	Contains the information associated with a content segment.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within a <i>UpdateConditionalContent</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	The acknowledgement type that you want to receive: <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<ClientName>	The client that contains the site that contains the content segments.	Nest within a <i>UpdateConditionalContent</i> element.
<SiteName>	The site that contains the content segment.	Nest within a <i>UpdateConditionalContent</i> element.

TABLE: Description of the elements in a *UpdateConditionalContent* request

Element Name	Description	Nesting
<ContentName>	The content segment to create.	Nest within a <i>UpdateConditionalContent</i> element.
<Language>	<p>Specifies the character set of the message's content. Only use if the site supports multiple character sets. Default is English (US-ASCII).</p> <p>Specify character sets, such as <i>Cyrillic (ISO-8859-5)</i>, instead of languages. Check the <i>DREAMmail</i> application for the exact spelling. For more information, see Language Support and Character Sets and Submitting multi-language content.</p>	Optional. Nest within a <i>UpdateConditionalContent</i> element.
<ConditionalContent>	Includes information pertaining to specific condition which include the condition itself, the profile field to which condition will be applied, contents to be included etc.	This element must be nested within an <i>UpdateConditionalContent</i> element.
< IF>	It includes profile field to which condition will be applied.	This element must be nested within an <i>ConditionalContent</i> element.

TABLE: Description of the elements in a *UpdateConditionalContent* request

Element Name	Description	Nesting
< Condition >	It includes the condition operation. Text operators: contains, equals, does not equal, comes before, comes before or equals, comes after, comes after or equals. Date/Numeric Operators in UpdateConditionalContent are as follows: <ul style="list-style-type: none"> • equals • notequals • lessthan • lessthanorequalto • greaterthan • greaterthanorequalto 	This element must be nested within a <i>ConditionalContent</i> element.
<ConditionValue>	It include condition data for the conditional profile field.	This element must be nested within a <i>ConditionalContent</i> element.
< ContentBody>	It include body to be included for a successful specific condition.	This element must be nested within a <i>ConditionalContent</i> element.
<TextBody>	Content for text-only email applications.	Optional. Nest within a <i>ContentBody</i> element.
<HtmlBody>	Content for HTML-capable email applications.	Optional. Nest within a <i>ContentBody</i> element.
<AolBody>	Content for AOL-capable email applications.	Optional. Nest within a <i>ContentBody</i> element.

TABLE: Description of the elements in a *UpdateConditionalContent* request

Element Name	Description	Nesting
<BodyText>	<p>The message content text.</p> <p>Important: You should always use escaped ASCII formatting for the content in the event that special or reserved characters are used. For more information, see Special Characters and Non-English Text.</p> <p>Important: When you include URLs in this element, the URLs must exactly match the way the URLs are specified in the <URLName> element. Do not place punctuation at the end of the URL in this element. Punctuation placed at the end of a URL causes the request to fail.</p>	Nest within a <i>TextBody</i> , <i>AolBody</i> , or <i>HtmlBody</i> element.
<DefaultContent>	Includes information pertaining to the default condition which includes the condition itself, profile field to which condition will be applied. This element is exactly the same as the <i>ConditionalContent</i> element in structure.	This element must be nested within an <i>UpdateConditionalContent</i> element.
<TrackedUrls>	The URLs to track. Each URL is identified with its own <URL></URL> element.	Optional. This element must be nested within an <i>UpdateConditionalContent</i> element.
<URL>	The URL to track.	Nest within a <i>TrackedUrls</i> element. You can include this more than once.
<URLName>	The actual URL to track; for example <code>http://www.somedomain.com</code>	Nest within a <i>URL</i> element.
<URLLabel>	Assigns a label to the URL to track.	Optional. Nest within a <i>URL</i> element.

Example of a UpdateConditionalContent request:

```
<?xml version="1.0" encoding="UTF-8"?>
<DMConnect>
  <UpdateConditionalContent>
    <AcknowledgementsTo>
      <EmailAddress>mukulk@cybage.com</EmailAddress>
      <Option>9</Option>
    </AcknowledgementsTo>
    <ClientName>Client_John</ClientName>
    <SiteName>Site_John</SiteName>
    <ContentName>Great_5</ContentName>
    <Language>English</Language>
    <ConditionalContent>
      <EnableHTMLEditor>true</EnableHTMLEditor>
      <IF>FirstName</IF>
      <Condition>equals</Condition>
      <ConditionValue>John</ConditionValue>
      <ContentBody>
        <TextBody>
          <BodyText>hi 1 in text part on 29 April. http://
            www.Great.com
          </BodyText>
        </TextBody>
        <AolBody>
          <BodyText>hi 1 in Aol part on 29 April.
        </BodyText>
        </AolBody>
        <HtmlBody>
          <BodyText>
        </BodyText>
        </HtmlBody>
      </ContentBody>
    </ConditionalContent>
    <Defaultcontent>
      <EnableHTMLEditor>true</EnableHTMLEditor>
      <ContentBody>
        <TextBody>
          <BodyText>Great Default text part on 29 April. Visit again.
        </BodyText>
        </TextBody>
        <AolBody>
          <BodyText>Great Default Aolpart on 29 April.
        </BodyText>
        </AolBody>
        <HtmlBody>
          <BodyText>Great Default Html part on 29 April.
        </BodyText>
      </ContentBody>
    </Defaultcontent>
  </UpdateConditionalContent>
</DMConnect>
```

```
        </HtmlBody>
      </ContentBody>
    </Defaultcontent>
    <TrackedUrls>
      <URL>
        <URLName>http://www.Great.com
        </URLName>
        <URLLabel>GreatURL</URLLabel>
      </URL>
    </TrackedUrls>
  </UpdateConditionalContent>
</DMConnect>
```

UpdateContent Request

An *UpdateContent* request modifies an existing content segment of the type specified.

Elements in the request

The *UpdateContent* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in an *UpdateContent* request

Element Name	Description	Nesting
<UpdateContent>	Contains the information associated with a content segment.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within an <i>UpdateContent</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.

TABLE: Description of the elements in an *UpdateContent* request

Element Name	Description	Nesting
<ContentType>	The type of the content segment to update. This is set when the content is created and cannot be updated. However, you must include it in this request. Possible values are: <ul style="list-style-type: none">• <i>StandardContent</i>• <i>ScriptContent</i>.	Nest within an <i>UpdateContent</i> element.
<ClientName>	The client that contains the site.	Nest within an <i>UpdateContent</i> element.
<SiteName>	The site that contains the content segment.	Nest within an <i>UpdateContent</i> element.
<ContentName>	The content segment to update.	Nest within an <i>UpdateContent</i> element.
<Language>	Specifies the character set of the message's content. Only use if the site supports multiple character sets. Default is English (US-ASCII). Specify character sets, such as <i>Cyrillic (ISO-8859-5)</i> , instead of languages. Check the <i>DREAMmail</i> application for the exact spelling. For more information, see Language Support and Character Sets and Submitting multi-language content .	Optional. Nest within an <i>UpdateContent</i> element.

TABLE: Description of the elements in an *UpdateContent* request

Element Name	Description	Nesting
<ContentText>	<p>Delimits the message content. This can contain a script content segment. For more information, see the <i>Advanced Messaging Guide</i>.</p> <p>Important: When including URLs in the segment, they must match the URLs in the <URLName> tags. Either do not use a punctuation mark at the end of the URL, or leave a space between the punctuation mark and the URL. If you place a punctuation mark at the end of a URL in the content segment, the request fails.</p> <p>Note: Place content in a language other than English in a CDATA section or escape it. For more information, see Special Characters and Non-English Text.</p>	Nest within an <i>UpdateContent</i> element.
<TrackedUrls>	The URLs tracked within this content. Each URL is identified with its own <URL></URL> element.	Optional. Nest within an <i>UpdateContent</i> element.
<URL>	The URL to track.	Nest within a <i>TrackedUrls</i> element. You can include this more than once.
<URLName>	The actual URL to track; for example http://www.somedomain.com.	Nest within a <i>URL</i> element.
<URLLabel>	Assigns a label to the URL to track.	Optional. Nest within a <i>URL</i> element.

Example of an UpdateContent request: The *UpdateContent* request adds a specific URL for subscribers living in Los Angeles to the *CreateContent* request example (see [Example of a CreateContent request](#)).

```
<DMConnect>
  <UpdateContent>
```

```

<AcknowledgementsTo>
  <EmailAddress>deployment@Epsilon.com</EmailAddress>
  <EmailAddress><![CDATA[s.o'neill@Epsilon.net]]>
</EmailAddress>
  <Option>9</Option>
</AcknowledgementsTo>
<ContentType>ScriptContent</ContentType>
<ClientName>DREAMmailTipster</ClientName>
<SiteName>DMTips</SiteName>
<ContentName>Tips_03162002</ContentName>
<ContentText><![CDATA[ <font face="Arial">
<Switch type = "string" value = #City#>

  <Case match="New York">
    <font face = "arial"><a href = "http://DMtips3.com">New Yorkers,
    click here for third tip.</font></a>
  </Case>

  <Case match="Los Angeles">
    <font face = "arial"><a href = "http://DMtips4.com">Click here for
    third tip. </font></a>
  </Case>

  <Default>
</Switch>
    ]]>
</ContentText>
<TrackedUrls>
  <URL>
    <URLName>http://www.DMtips3.com</URLName>
    <URLLabel>NY Tips</URLLabel>
  </URL>
  <URL>
    <URLName>http://www.DMtips4.com</URLName>
    <URLLabel>LA Tips</URLLabel>
  </URL>
</TrackedUrls>
</UpdateContent>
</DMConnect>

```

UpdateMailing Request

An *UpdateMailing* request updates an existing message of the specified type. The content in this request completely replaces the content in the existing message. Therefore, to update one paragraph you must resubmit the entire message.

Deselecting tracked URLs

You can stop tracking a previously tracked URL by omitting the URL within the `<TrackedUrls>` element in the *UpdateMailing* request. For more information, see the `<TrackedUrls>` element in [Table](#) below.

Elements in the request

The *UpdateMailing* request can contain the elements listed in the table below, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in an *UpdateMailing* request

Element Name	Description	Nesting
<code><UpdateMailing></code>	Contains the information associated with a message.	Nest within a <i>DMConnect</i> element.
<code><AcknowledgementsTo></code>	Email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within an <i>UpdateMailing</i> element.
<code><EmailAddress></code>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<code><Option></code>	<p>The type of acknowledgement required:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: If you set this to 0, you must still include content in the <code><EmailAddress></code> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.

TABLE: Description of the elements in an *UpdateMailing* request

Element Name	Description	Nesting
<MailingType>	The type of the message to be updated. This is set when the message is created and cannot be updated. However, you must include it in this request. Possible values are: <ul style="list-style-type: none"> • <i>StandardMailing</i> • <i>NotificationTriggerMailing</i> • <i>RecurringTriggerMailing</i> • <i>ApprovalMailing</i>. 	Nest within an <i>UpdateMailing</i> element.
<ClientName>	The client that contains the site that contains the message that you want to modify.	Nest within an <i>UpdateMailing</i> element.
<SiteName>	The site that contains the campaign that contains the message that you want to modify.	Nest within an <i>UpdateMailing</i> element.
<CampaignName>	The campaign that contains the message that you want to modify.	Nest within an <i>UpdateMailing</i> element.
<MailingName>	The message you want to modify.	Nest within an <i>UpdateMailing</i> element.
<FromAddr>	From address for mailing.	Optional. This element must be nested within an <i>UpdateMailing</i> .
<AliasName>	From Address for message level.	This element must be nested within FromAddr element.
<DisplayName>	Display name for From Address.	Optional. This element must be nested within a FromAddr element.
<ToList>	The set of mailing lists to which the message is sent.	Optional for Standard and Approval mailing. Nest within an <i>UpdateMailing</i> element.
<ListName>	A mailing list to which the message is to be sent.	Nest within a <i>ToList</i> element. You can include this more than once.

TABLE: Description of the elements in an *UpdateMailing* request

Element Name	Description	Nesting
<Filter>	A SQL <i>where</i> clause that restricts the set of subscribers who receive the message.	Optional. Nest within an <i>UpdateMailing</i> element.
<Suppression>	Exclude members who have received previous mailings or campaigns.	Optional. Nest within a <i>CreateMailing</i> element.
<SuppressCampaign>	Exclude members who have received previous mailings from a specified campaign.	Nest within a <i>Suppression</i> element.
<CampaignName>	Name of campaign to be suppressed.	Nest within a <i>SuppressCampaign</i> element.
<MailingName>	Name of mailing to be suppressed.	Optional. Nest within a <i>CampaignName</i> element.
<Subject>	The subject of the message. Important: You should always use escaped ASCII formatting for the subject in the event that special or reserved characters are used. For more information see Special Characters and Non-English Text .	Nest within an <i>UpdateMailing</i> element.
<Language>	Specifies the character set of the message's content. Only use if the site supports multiple character sets. Default is English (US-ASCII). Specify character sets, such as <i>Cyrillic (ISO-8859-5)</i> , instead of languages. Check the <i>DREAMmail</i> application for the exact spelling. For more information, see Language Support and Character Sets and Submitting multi-language content .	Optional. Nest within an <i>UpdateMailing</i> element.

TABLE: Description of the elements in an *UpdateMailing* request

Element Name	Description	Nesting
<MessageContent>	Delimits the content of the message. Note: There must be content within at least one of <TextBody>, <HtmlBody>, or <AolBody>.	Nest within an <i>UpdateMailing</i> element.
<TextBody>	The message for text-only email applications.	Optional. Nest within a <i>MessageContent</i> element.
<HtmlBody>	The message for HTML-capable email applications.	Optional. Nest within a <i>MessageContent</i> element.
<AolBody>	The message for AOL- capable email applications.	Optional. Nest within a <i>MessageContent</i> element.
<BodyText>	The text for the specified message content type. Important: You should always use escaped ASCII formatting for the content in the event that special or reserved characters are used. For more information, see Special Characters and Non-English Text . Important: When you include URLs in this element, the URLs must exactly match the way the URLs are specified in the <URLName> element. Do not place punctuation at the end of the URL in this element. Punctuation placed at the end of a URL causes the request to fail.	Nest within a <i>TextBody</i> , <i>AolBody</i> or <i>HtmlBody</i> element.
<TrackedUrls>	The URLs tracked within this message. They must be nested in the appropriate message-body type. Each URL is identified with its own <URL></URL> element. Note: To stop tracking a URL that was previously tracked, omit the <i>URL</i> within this element.	Optional. Nest within a <i>TextBody</i> , <i>AolBody</i> or <i>HtmlBody</i> element.

TABLE: Description of the elements in an *UpdateMailing* request

Element Name	Description	Nesting
<URL>	The URL to track.	Nest within a <i>TrackedUrls</i> element. You can include this more than once.
<URLName>	The actual URL to track; for example <code>http://www.somedomain.com</code> .	Nest within a <i>URL</i> element.
<URLLabel>	Assigns a label to the URL to track.	Optional. Nest within a <i>URL</i> element.
<ResetSettings>	Set this to <i>false</i> or zero to preserve the current state of the <i>HTMLOpenRateFlag</i> , <i>AutoSenseFlag</i> , <i>RetryAutoSenseFlag</i> , and <i>IncludeCampaignFooterFlag</i> . If this is set to <i>true</i> , then all four flag values will be reset to <i>false</i> .	Optional. Nest within an <i>UpdateMailing</i> element.
<HtmlOpenRateFlag>	Set this to <i>true</i> or <i>1</i> to enable HTML open rate tracking. Any other value disables tracking. By default, tracking is disabled.	Optional. Nest within an <i>UpdateMailing</i> element.
<AutosenseFlag>	Set this to <i>true</i> or <i>1</i> to enable autosensing. Any other value disables autosensing. By default, autosensing is disabled.	Optional. Nest within an <i>UpdateMailing</i> element.
<RetryAutosenseFlag>	Verifies subscriber email applications previously marked as text-only. Set this to <i>true</i> or <i>1</i> , to verify if addresses marked text-only are HTML-capable. Any other value disables this action. By default, this is disabled.	Optional. Nest within an <i>UpdateMailing</i> element.
<IncludeCampaignFooter Flag>	Set this to <i>true</i> or <i>1</i> to include the campaign footer within the text message body. Any other value excludes the footer. By default, the footer is excluded. <i>Note: This is also referred to as the Message Footer. For more information, see the DREAMmail help system.</i>	Optional. Nest within an <i>UpdateMailing</i> element.

TABLE: Description of the elements in an *UpdateMailing* request

Element Name	Description	Nesting
<CookieOptionflag>	<p>Specify when <i>DREAMmail</i> sends a cookie to the subscriber. If omitted, this defaults to 0.</p> <p>If HTML Open Rate tracking is enabled, the accepted values are:</p> <ul style="list-style-type: none"> • 0 — set cookie on click-through • 1 — set cookie on message open • 2 — set cookie on both click-through and message open. <p>If HTML Open Rate tracking is not enabled, you can only set this to 0.</p>	Optional. Nest within an <i>UpdateMailing</i> element.
<ChannelViewFlag>	Set this to <i>true</i> or 1 to use an automated process to extract message data (once deployment is complete) and report on it using the <i>ChannelView</i> analytics tool.	Optional. Nest within a <i>UpdateMailing</i> element.
<SendMultipartFlag>	Set this to <i>true</i> or 1 to send all messages in multi-part format regardless of the individual recipient settings in <i>DREAMmail</i> . Used primarily for European ISPs that handle HTML email as a file attachment.	Optional. Nest within a <i>UpdateMailing</i> element.
<ReplyToAddr>	Reply-To-Address for mailing.	<p>Optional. This element must be nested within an <i>UpdateMailing</i>.</p> <p>Note: If using <i>TrackUnsubscribe</i>s do not use <i>ReplyToAddr</i> element.</p>

TABLE: Description of the elements in an *UpdateMailing* request

Element Name	Description	Nesting
<AliasName>	Reply-To-Address for message level.	Optional. This element must be nested within <i>ReplyToAddr</i> element. This element cannot be included if <i>TrackUnsubFromReplies</i> element is included.
<DisplayName>	Display name for Reply_To_Address.	Optional. This element must be nested within a <i>ReplyToAddr</i> element.
<TrackUnsubFromReplies >	To enable or disable the Track unsubscribes from replies. <ul style="list-style-type: none"> • 0 • 1 • True • False (Case insensitive)	Optional. This element must be nested within a <i>ReplyToAddr</i> element. This element cannot be included if <i>AliasName</i> element is included.
<TrackUnsubscribes>	Set for specifying tracking unsubscribe flags. Leave empty to inherit default campaign values	Optional. Nest within a <i>UpdateMailing</i> element. Note: If using <i>ReplyToAddr</i> and/or <i>SubunsubAddr</i> elements, do not use <i>TrackUnsubscribes</i> .
<RepliesFlag>	Set to 1 or true to enable track unsubscribe requests from replies. Set to 0 or false to disable.	Optional. Nest within a <i>TrackUnsubscribes</i> element.
<UnsubscriptionAliasFlag>	Set to 1 or true to enable track unsubscribe requests from unsubscription address. Set to 0 or false to disable.	Optional. Nest within a <i>TrackUnsubscribes</i> element.

TABLE: Description of the elements in an *UpdateMailing* request

Element Name	Description	Nesting
<ArmMessage>	This will be contains value that will be for Arm or Open message for Notification Triggered messages. <ul style="list-style-type: none">• 1• 0• True• False (case insensitive)	Optional. This element must be nested within an <i>UpdateMailing</i> . This element can only be included for Notification Triggered messages.
< SubunsubAddr>	It includes Sub, Unsub and TrackUnsub element tags.	Optional. This element must be nested within an <i>UpdateMailing</i> . Note: If using TrackUnsubscribes do not use SubunsubAddr element.
< SubscribeAlias>	Subscription alias at message level. If not specified default will be taken from campaign level.	Optional. This element must be nested within SubunsubAddr element.
< UnsubscribeAlias >	Unsubscription alias at message level. If value not specified, default will be taken from campaign level.	Optional. This element must be nested within SubunsubAddr element.

TABLE: Description of the elements in an *UpdateMailing* request

Element Name	Description	Nesting
< TrackUnsub >	<p>To enable or disable the Track unsubscribes from Unsub address.</p> <p>If value not specified, default will be taken from campaign level.</p> <ul style="list-style-type: none"> • 0 • 1 • True • False <p>(Case insensitive)</p>	Optional. This element must be nested within SubunsubscribeAddress element.

Example of an UpdateMailing request: This UpdateMailing request adds a content segment and give labels to the URLs in the message created in the *CreateMailing* example (see [Example of a CreateMailing request](#)).

```

<DMConnect>
  <UpdateMailing>
    <AcknowledgementsTo>
      <EmailAddress>deployment@Epsilon.com</EmailAddress>
      <EmailAddress>hot_tip_editor@Epsilon.com</EmailAddress>
      <Option>9</Option>
    </AcknowledgementsTo>
    <MailingType>StandardMailing</MailingType>
    <ClientName>DREAMmailTipster</ClientName>
    <SiteName>DMTips</SiteName>
    <CampaignName>MarchTips</CampaignName>
    <MailingName>DMTips_03162002</MailingName>
    <FromAddr>
      <AliasName>Alias_Name</AliasName>
      <DisplayName>Name</DisplayName>
    </FromAddr>
    <ToList>
      <ListName>DMTipsList</ListName>
      <ListName>DMTipsAccepters</ListName>
    </ToList>
    <Suppression>
      <SuppressCampaign>
        <CampaignName>suppress_campaign_1</CampaignName>
        <MailingName>suppress_mailing_1</MailingName>
        <MailingName>suppress_mailing_2</MailingName>
      </SuppressCampaign>
    </Suppression>
  </UpdateMailing>
</DMConnect>

```

```

    <SuppressCampaign>
      <CampaignName>suppress_campaign_2</CampaignName>
    </SuppressCampaign>
  </Suppression>
  <Subject>Hot DM Tip for March 16 2002</Subject>
  <MessageContent>
    <TextBody>
      <BodyText> <![CDATA[
Dear #FirstName#,
These are your Hot DREAMmail Tips for March 16 in text format.
http://www.DMtips.com
http://www.DMtips2.com
#CONTENT:Tips_03162002T#
Thanks,
The DREAMmailTipster]]>
      </BodyText>
      <TrackedUrls>
        <URL>
          <URLName>http://www.DMtips.com</URLName>
          <URLLabel>Tip_1</URLLabel>
        </URL>
        <URL>
          <URLName>http://www.DMtips2.com</URLName>
          <URLLabel>Tip_2</URLLabel>
        </URL>
      </TrackedUrls>
    </TextBody>
    <HtmlBody>
      <BodyText>
        <![CDATA[ <HTML><body>
Dear #FirstName#,<br>
These are your Hot DREAMmail Tips for March 16 in HTML format.<br>
<a href="http://www.DMtips.com">Click Here for the first tip</a><br>
<a href="http://www.DMtips2.com">Click Here for the second tip</a><br>
#CONTENT:Tips_03162002#
Thanks,<br>
The DREAMmailTipster</body></HTML> ]]>
      </BodyText>
      <TrackedUrls>
        <URL>
          <URLName>http://www.DMtips.com</URLName>
          <URLLabel>Tip_1</URLLabel>
        </URL>
        <URL>
          <URLName>http://www.DMtips2.com</URLName>
          <URLLabel>Tip_2</URLLabel>
        </URL>
      </TrackedUrls>
    </HtmlBody>
  </MessageContent>
</DMConnectRequest>

```



```

        </TrackedUrls>
    </HtmlBody>
    <AolBody>
        <BodyText>
<![CDATA[<HTML><body>
Dear #FirstName#,<br>
These are your Hot DREAMmail Tips for March 16 in AOL format.<br>
<a href="http://www.DMtips.com">Click Here for the first tip</a><br>
<a href="http://www.DMtips2.com">Click Here for the second tip</a><br>
#CONTENT:Tips_03162002#
Thanks,<br>
The DREAMmailTipster</body></HTML>
]]>
        </BodyText>
        <TrackedUrls>
            <URL>
                <URLName>http://www.DMtips.com</URLName>
                <URLLabel>Tip_1</URLLabel>
            </URL>
            <URL>
                <URLName>http://www.DMtips2.com</URLName>
                <URLLabel>Tip_2</URLLabel>
            </URL>
        </TrackedUrls>
    </AolBody>
</MessageContent>
<ResetSettings>False</ResetSettings>
<HtmlOpenRateFlag>True</HtmlOpenRateFlag>
<AutosenseFlag>True</AutosenseFlag>
<RetryAutosenseFlag>True</RetryAutosenseFlag>
<IncludeCampaignFooterFlag>True</IncludeCampaignFooterFlag>
<CookieOptionflag>1</CookieOptionflag>
<ChannelViewFlag>1</ChannelViewFlag>
<SendMultipartFlag>1</SendMultipartFlag>
<ReplyToAddr>
    <AliasName>ReplyToAddress_Name</AliasName>
    <DisplayName>Name</DisplayName>
    <TrackUnsubFromReplies>0</TrackUnsubFromReplies>
</ReplyToAddr>
<TrackUnsubscribes>
    <RepliesFlag>1</RepliesFlag>
    <UnsubscriptionAliasFlag>1</UnsubscriptionAliasFlag>
</TrackUnsubscribes>
<SubunsubAddr>
    <SubscribeAlias>None</SubscribeAlias>
    <UnsubscribeAlias>Unsub_addr</UnsubscribeAlias>
    <TrackUnsub>true</TrackUnsub>

```

```
        </SubunsubAddr>  
    </UpdateMailing>  
</DMConnect>
```

UpdateProfile Request

An *UpdateProfile* request modifies the profile fields that belong to the specified email addresses. You can use this to change email addresses in these profiles.

Elements in the request

The *UpdateProfile* request can contain the elements listed in the following table, shown in hierarchical order. The *Nesting* column explains the hierarchy in which the element must be included.

TABLE: Description of the elements in a *UpdateProfile* request

Element Name	Description	Nesting
<UpdateProfile>	Contains the information associated with the profile update to be performed.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification by email when the request succeeds or fails.	Optional. Nest within an <i>UpdateProfile</i> element.
<EmailAddress>	An email address to which the notification message is sent.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	<p>The acknowledgement type that you want to receive:</p> <ul style="list-style-type: none">• 0 — no notifications• 1 — only send confirmation acknowledgements• 2 — only send failure acknowledgements• 9 — send all acknowledgements. <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element.
<JobName>	The job for which you want the status. If you do not specify a JobName, a numeric JobName is returned with the receipt message.	Optional. Nest within a <i>UpdateProfile</i> element.
<ClientName>	The client that contains the customer profiles to be updated.	Nest within an <i>UpdateProfile</i> element.

TABLE: Description of the elements in a *UpdateProfile* request

Element Name	Description	Nesting
<SiteName>	The site that contains the customer profiles to be updated.	Nest within an <i>UpdateProfile</i> element.
<ImportData>	The profile data to be updated.	Nest within an <i>UpdateProfile</i> element.
<Row> or <R>	The data for a single row.	Nest within an <i>ImportData</i> element. You can include this more than once.
<MemberIdentifier>	The email address of the subscriber to be updated. <i>Note: If the row includes <C> <Name> EmailAddr </Name> <Value> ... </Value> </C>, the value in the Column element changes the subscriber's email address.</i>	Optional. Nest within a <i>Row</i> element. You can only include one instance of this element per row element.
<Column> or <C>	The data and identifier of a single column within a row.	Nest within a <i>Row</i> element. You can include this more than once.
<Name>	The column name.	Nest within a <i>Column</i> element.
<Value>	The value of the column.	Nest within a <i>Column</i> element.

Example of a *UpdateProfile* request:

```

<DMConnect>
  <UpdateProfile>
    <ClientName>DREAMmailTipster</ClientName>
    <SiteName>DMTips</SiteName>
    <ImportData>
      <Row>
        <MemberIdentifier>Richard@BobC.com</MemberIdentifier>
        <Column>
          <Name>EmailAddr</Name>
          <Value>Rick@Tantex.com</Value>
        </Column>
        <Column>
          <Name>City</Name>
          <Value>Toronto</Value>
        </Column>
      </Row>
    </ImportData>
  </UpdateProfile>
</DMConnect>

```

```
    </ImportData>  
  </UpdateProfile>  
</DMConnect>
```

UpdateRTMMailing

The *UpdateRTMMailing* request updates an existing *RTM* message. The content in this request replaces completely the content in the existing message. This means that if you want to update one paragraph in a message, you must submit the entire message, not just the one paragraph.

For more information about *RTM* and a description of related elements, see the *Real-Time Messaging Guide*.

UpdateTrackedURL Request

An *UpdateTrackedURL* request is used to modify the redirect location of your tracked message URLs, including URLs in the message itself or in any content segments and offer tables nested within the message. This request allows you to modify a tracked URL in any message with a status of sending, armed, aborted, paused, archived, closed, finished, or error.

Elements in the request

The *UpdateTrackedURL* request can contain the elements listed in the following table, shown in hierarchical order. The Nesting column explains which elements must be included.

TABLE: Description of the elements of an *UpdateTrackedURL* response

Element Name	Description	Nesting
<UpdateTrackedURL>	Contains the information associated with a message URL to be updated.	Nest within a <i>DMConnect</i> element.
<AcknowledgementsTo>	The email addresses that receive a notification when the request succeeds or fails.	Optional. Nest within an <i>UpdateTrackedURL</i> element.
<EmailAddress>	The email addresses that receive a notification when the request succeeds or fails.	Nest within an <i>AcknowledgementsTo</i> element. You can include this more than once.
<Option>	The acknowledgement type that you want to receive: <ul style="list-style-type: none">• 0 – no notifications• 1 – only send confirmation acknowledgements• 2 – only send failure acknowledgements• 9 – send all acknowledgements <p>Note: Even if you enter 0, you must include content in the <EmailAddress> element.</p>	Optional. Nest within an <i>AcknowledgementsTo</i> element
<ClientName>	The client that contains the site where you want to retrieve the tracked URLs of a message.	Nest within an <i>UpdateTrackedURL</i> element.

Element Name	Description	Nesting
<SiteName>	The site within the specified client.	Nest within an <i>UpdateTrackedURL</i> element.
<CampaignName>	The campaign that contains the message.	Nest within an <i>UpdateTrackedURL</i> element.
<MailingName>	The message containing the tracked URLs you want to update.	Nest within an <i>UpdateTrackedURL</i> element.
<UpdateURLs>	The parent tag of the URLs to update. Each URL is identified with its own <UpdateURL></UpdateURL> element.	Nest within an <i>UpdateTrackedURL</i> element.
<UpdateURL>	The URL to update.	Nest within an <i>UpdateURLs</i> element. You can include this element more than once.
<URLLink>	The URL within the message to update; for example <code>http://www.fallriversports.com/shoe</code>	Nest within an <i>UpdateURL</i> element.
<URLLabel>	The label (if applicable) given to the URL.	Optional. Nest within an <i>UpdateURL</i> element.
<UpdatedURLLink>	The updated URL to track; for example <code>http://www.fallriversports.com/home</code>	Nest within an <i>UpdateURL</i> element.
<UpdatedURLLabel>	The label (if applicable) given to the updated URL.	Optional. Nest within an <i>UpdateURL</i> element.

Example of an UpdateTrackedURL Request:

```

<DMConnect>
  <UpdateTrackedURL>
    <AcknowledgementsTo>
      <EmailAddress>user@yourcompany.com</EmailAddress>
      <Option>9</Option>
    </AcknowledgementsTo>
    <ClientName>Client1</ClientName>
    <SiteName>Site1</SiteName>
    <CampaignName>Campaign1</CampaignName>
    <MailingName>Message1</MailingName>
    <UpdateURLs>

```



```
<UpdateURL><URLLink><![CDATA[http://www.fallriversports.com/
shoe]]></URLLink><URLLabel>shoe offer</URLLabel>
<UpdatedURLLink><![CDATA[http://www.fallriversports.com/home]]></
UpdatedURLLink>
<UpdatedURLLabel>Home page</UpdatedURLLabel></UpdateURL>
</UpdateURLs>
</UpdateTrackedURL>
</DMConnect>
```

GENERATING STANDARD REPORTS WITH *DMCONNECT*

This section describes examples of the reports that you can access using *DMConnect*.

This section discusses the following topics:

- [DeliveryResponse Report Request](#)
- [Bouncebacks Report Request](#)
- [Clickthroughs Report Request](#)
- [ListActivity Report Request](#)
- [EmailAddressChanges Report Request](#)
- [ExportList Report Request](#)
- [ProfileChanges Report Request](#)
- [RTMDelivered Report Request](#)
- [RTMNotDelivered Report Request](#)
- [RTMClicks Report Request](#)
- [RTMHTMLOpens Report Request](#)
- [RTMSpotlight Report Request](#)
- [Unsubscription Report Request](#)

You generate all of these reports using the *ExportReport* request. Most of the examples in this section show the *ExportReport* request you must submit for the report and the content you must enter in the *Name* element.

You can pull *RTM* reports on an hourly basis. They only include data for a 24-hour period. This is because these reports can contain a large amount of data.

DeliveryResponse Report Request

The *DeliveryResponse* report request provides delivery and response information for all campaigns, by message for a specific site and time frame.

For a description of the elements that can be used, see [ExportReport Request](#).

Parameters for DeliveryResponse report request

The table below shows the mandatory parameters for this request.

TABLE: Description of the parameters in a *DeliveryResponse* report request

Parameter Name	Description	Nesting
ClientName	The name of the client.	This must be nested within a <i>Name</i> element.
SiteName	The name of the site from which you want the report data.	This must be nested within a <i>Name</i> element.
FromDate	The first date of the interval for which you want the report data. For information about acceptable date formats, see FromDate and EndDate Parameters .	This must be nested within a <i>Name</i> element.
EndDate	The last date of the interval for which you want the report data. For information about acceptable date formats, see FromDate and EndDate Parameters .	This must be nested within a <i>Name</i> element.

Note: The *FromDate* and *EndDate* parameters are inclusive. This means that the date range that you specify includes the data of both the starting and the ending dates that you set.

Example of a DeliveryResponse request

```
<DMConnect>
  <ExportReport>
    <AcknowledgementsTo>
      <EmailAddress>Bob@Epsilon.com</EmailAddress>
      <EmailAddress>Dave@Epsilon.com</EmailAddress>
    </AcknowledgementsTo>
    <ReportType>DeliveryResponse</ReportType>
    <ExportName>2002Q1Delivery</ExportName>
    <Parameters>
      <Parameter>
        <Name>ClientName</Name>
        <Value>BobsSports</Value>
```

```

</Parameter>
<Parameter>
  <Name>SiteName</Name>
  <Value>Volleyball</Value>
</Parameter>
<Parameter>
  <Name>FromDate</Name>
  <Value>2002-01-01</Value>
</Parameter>
<Parameter>
  <Name>EndDate</Name>
  <Value>2002-03-30</Value>
</Parameter>
</Parameters>
</ExportReport>
</DMConnect>

```

Output for DeliveryResponse report request

The fields in this report are explained in the table below.

TABLE: Delivery and Response summary

Field Name	Description
ClientName	The name of client that contains the selected site.
SiteName	The name of the selected site.
Campaign Name	The names of all campaigns for the given site.
Mailing Name	The names of the messages for the campaigns in the site.
# Delivered	The number of messages that were delivered
% Delivered	The number of delivered messages as a percentage of total messages that were sent.
# Not Delivered	The number of messages that were not delivered.
Click-Throughs	The number of click-throughs and pass alongs.
% Click-Throughs	The number of click-throughs as a percentage of messages that were delivered.
Distinct Click-Throughs	The number of unique users who clicked through on each message in the campaign.
Pass Alongs	The number of unique users who clicked through on any message as a result of having it forwarded to them.
# HTML Opened	The number of unique users who previewed or opened an HTML format email.

TABLE: Delivery and Response summary

Field Name	Description
% HTML Opened	The number of unique users who previewed or opened an HTML format message as a percentage of HTML messages messages that were delivered.
Ecommerce #	The number of all ecommerce transactions recorded.
Ecommerce %	The number of all ecommerce transactions recorded as a percentage of messages that were delivered.
Ecommerce Value	The total value of the ecommerce transactions recorded.
Text Deliv.	The number of text format messages that were delivered.
AOL Deliv.	The number of AOL format messages that were delivered.
HTML Deliv.	The number of HTML format messages that were delivered.
Autosense Deliv.	The number of autosense format messages that were delivered.
Total	The number of delivered messages.
Total BB	The number of bounce backs.
Hard BB	The number of hard bounce backs.
Soft BB	The number of soft bounce backs.
Invalid Address	The number of messages not sent because addresses were incomplete or incorrect.
Deliv. Error	The number of messages that could not be delivered due to network errors.
Total	The total number of messages not delivered.
Start Time	The time and date that the first message was sent. When you open the report in a spreadsheet, this cell is identified as a date type that you can format according to your preferences.
EndTime	The time and date that the last message was sent. When you open the report in a spreadsheet, this cell is identified as a date type that you can format according to your preferences.

Bouncebacks Report Request

The *Bouncebacks* report request provides information about bounce backs for a specific site over a specified period, sorted by campaign and message.

For a description of the elements that can be used, see [ExportReport Request](#).

Parameters for Bouncebacks report request

The table below shows the mandatory parameters for this report.

TABLE: Description of the parameters in a *Bouncebacks* report request

Parameter Name	Description	Nesting
ClientName	The name of the client.	This must be nested within a <i>Name</i> element.
SiteName	The name of the site from which you want the report data.	This must be nested within a <i>Name</i> element.
FromDate	The first date of the interval for which you want the report data. For information about acceptable date formats, see FromDate and EndDate Parameters .	This must be nested within a <i>Name</i> element.
EndDate	The last date of the interval for which you want the report data. For information about acceptable date formats, see FromDate and EndDate Parameters .	This must be nested within a <i>Name</i> element.

Note: The *FromDate* and *EndDate* parameters are inclusive. This means that the date range that you specify includes the data of both the starting and the ending dates that you set.

The table below shows parameters that are optional for this report:

TABLE: Description of the optional parameters in a *Bouncebacks* report request

Parameter Name	Description	Nesting
CampaignName	The name of the campaign for which you want the report data.	This must be nested within a <i>Name</i> element.
MailingName	The name of the message for which you want the report data.	This must be nested within a <i>Name</i> element.

Example of a Bouncebacks request

```
<DMConnect>
  <ExportReport>
    <AcknowledgementsTo>
      <EmailAddress>Bob@Epsilon.com</EmailAddress>
      <EmailAddress>Dave@Epsilon.com</EmailAddress>
    </AcknowledgementsTo>
    <ReportType>Bouncebacks</ReportType>
    <ExportName>BouncebackReport</ExportName>
    <Parameters>
      <Parameter>
        <Name>ClientName</Name>
        <Value>BobsSports</Value>
      </Parameter>
      <Parameter>
        <Name>SiteName</Name>
        <Value>hockey</Value>
      </Parameter>
      <Parameter>
        <Name>CampaignName</Name>
        <Value>FallCampaign</Value>
      </Parameter>
      <Parameter>
        <Name>MailingName</Name>
        <Value>New for Fall 2002</Value>
      </Parameter>
      <Parameter>
        <Name>FromDate</Name>
        <Value>2002-07-01</Value>
      </Parameter>
      <Parameter>
        <Name>EndDate</Name>
        <Value>2002-08-31</Value>
      </Parameter>
    </Parameters>
  </ExportReport>
</DMConnect>
```

Output for a Bounceback report request

The fields in this report are explained in the table below.

TABLE: Bouncebacks report

Field Name	Description
Client Name	The name of client that contains the selected site.
Site Name	The name of the selected site.

TABLE: Bouncebacks report

Field Name	Description
Campaign Name	The names of all campaigns for the given site.
Mailing Name	The names of the messages for the campaigns in the site.
Bounceback Type	The type of bounce back that occurred: hard or soft.
Date / Time of Bounceback	The date and time that the message was sent. When you open the report in a spreadsheet, this cell is identified as a date type that you can format according to your preference.
Email Address	The email addresses that returned bounce backs.

Clickthroughs Report Request

The *ClickThroughs* report request provides information about click-throughs over a specified period.

For a description of the elements that can be used, see [ExportReport Request](#).

Parameters for Clickthroughs report request

The table below shows the mandatory parameters for this report.

TABLE: Description of the parameters in a *ClickThrough* report request

Parameter Name	Description	Nesting
ClientName	The name of the client.	This must be nested within a <i>Name</i> element.
SiteName	The name of the site from which you want the report data.	This must be nested within a <i>Name</i> element.
FromDate	The first date of the interval for which you want the report data. For information about acceptable date formats, see FromDate and EndDate Parameters .	This must be nested within a <i>Name</i> element.
EndDate	The last date of the interval for which you want the report data. For information about acceptable date formats, see FromDate and EndDate Parameters .	This must be nested within a <i>Name</i> element.

Note: The *FromDate* and *EndDate* parameters are inclusive. This means that then date range that you specify includes the data of both the starting and the ending dates that you set.

The table below shows the parameters that are optional for this report.

TABLE: Description of the optional parameters in a *Clickthroughs* report request

Parameter Name	Description	Nesting
CampaignName	The name of the specific campaign for which you want the report data.	This must be nested within a <i>Name</i> element.
MailingName	The name of the specific message for which you want the report data.	This must be nested within a <i>Name</i> element.

Example of a Clickthroughs request

```
<DMConnect>
  <ExportReport>
    <AcknowledgementsTo>
      <EmailAddress>Bob@Epsilon.com</EmailAddress>
      <EmailAddress>Dave@Epsilon.com</EmailAddress>
    </AcknowledgementsTo>
    <ReportType>Clickthroughs</ReportType>
    <ExportName>ClickthroughReport</ExportName>
    <Parameters>
      <Parameter>
        <Name>ClientName</Name>
        <Value>BobsSports</Value>
      </Parameter>
      <Parameter>
        <Name>SiteName</Name>
        <Value>Hockey</Value>
      </Parameter>
      <Parameter>
        <Name>CampaignName</Name>
        <Value>Fall Campaign</Value>
      </Parameter>
      <Parameter>
        <Name>MailingName</Name>
        <Value>New for Fall 2002</Value>
      </Parameter>
      <Parameter>
        <Name>FromDate</Name>
        <Value>2002-07-01</Value>
      </Parameter>
      <Parameter>
        <Name>EndDate</Name>
        <Value>2002-07-31</Value>
      </Parameter>
    </Parameters>
  </ExportReport>
</DMConnect>
```

Output for Clickthroughs report request

The fields in this report are explained in the table below.

TABLE: Clickthroughs report

Field Name	Description
Client Name	The name of client that contains the selected site.
Site Name	The name of the selected site.

TABLE: Clickthroughs report

Field Name	Description
Campaign Name	The names of all campaigns for the given site.
Mailing Name	The names of the messages for the campaigns in the site.
Email Addr	The email address of the subscribers who clicked through a given URL from any message within the selected site. Pass along click-throughs are displayed with the original subscriber's address followed by (<i>pass along</i>).
URL	The URLs that were tracked for the given list.
URL Label	The label (if applicable) given to the URL.
Hit From	The source of the click-through (text, AOL, HTML format, or autosense message).
Date / Time	The date and time of the click-through. When you open the report in a spreadsheet, this cell is identified as a date type that you can format according to your preference.

ListActivity Report Request

The *ListActivity* report request provides information about subscribers who subscribed or unsubscribed during the period specified by the *FromDate* and *EndDate* parameters.

This report includes subscriptions that were made via email, *List Generator*, and *Subscriber Manager* and unsubscriptions that were made using *DMConnect*, email, *List Generator*, *Subscriber Manager*, and internal *DREAMmail* tools such as the *Abuse Tool* or the *DREAMmail Browser Tool*. Subscriptions and unsubscriptions that were made by any other means (for example, FTP) are not included in this report.

For a description of the elements that can be used, see [ExportReport Request](#).

Parameters for ListActivity report request

The table below shows the mandatory parameters for this report.

TABLE: Description of the parameters in a *ListActivity* report request

Parameter Name	Description	Nesting
ActivityType	This indicates which activity is being carried out: <ul style="list-style-type: none"> <i>Subscription</i> <i>Unsubscription</i>. 	This must be nested within a <i>Name</i> element.
ClientName	The name of the client.	This must be nested within a <i>Name</i> element.
SiteName	The name of the site from which you want the report data.	This must be nested within a <i>Name</i> element.
ListName	The name of the specific mailing list for which you want the report data.	This must be nested within a <i>Name</i> element.
FromDate	The first date of the interval for which you want the report data. For information about acceptable date formats, see FromDate and EndDate Parameters .	This must be nested within a <i>Name</i> element.
EndDate	The last date of the interval for which you want the report data. For information about acceptable date formats, see FromDate and EndDate Parameters .	This must be nested within a <i>Name</i> element.

TABLE: Description of the parameters in a *ListActivity* report request

Parameter Name	Description	Nesting
IncludeField	<p>The profile fields that you want to include in your report. The accepted values are:</p> <ul style="list-style-type: none"> Names of valid profile fields <i>ALL</i> – includes all profile fields <i>NONE</i> – excludes all profile fields <p>Note: The values <i>ALL</i> and <i>NONE</i> override any fields that you have listed. For example, if you specify the <i>FirstName</i> field and <i>NONE</i>, no fields are listed in the report. If you specify <i>FirstName</i> and <i>ALL</i>, then all fields are listed, and <i>FirstName</i> appears only once.</p>	This must be nested within a <i>Name</i> element.

Note: The *FromDate* and *EndDate* parameters are inclusive. This means that the date range that you specify includes the data of both the starting and the ending dates that you set.

Parameters for profile fields can be included in a request one of three ways. The following three examples show the possible solutions.

Example 1:

```
<Parameters>
  <Parameter>
    <Name>IncludeField</Name>
    <Value>Field1</Value>
  </Parameter>
  ...
  <Parameter>
    <Name>IncludeField</Name>
    <Value>Fieldn</Value>
  </Parameter>
</Parameters>
```

Example 2:

```
<Parameters>
  <Parameter>
    <Name>IncludeField</Name>
    <Value>ALL</Value>
  </Parameter>
</Parameters>
```

Example 3:

```
<Parameters>
  <Parameter>
    <Name>IncludeField</Name>
    <Value>NONE</Value>
  </Parameter>
</Parameters>
```

**Example of a
ListActivity
request**

```
<DMConnect>
  <ExportReport>
    <AcknowledgementsTo>
      <EmailAddress>Bob@Epsilon.com</EmailAddress>
      <EmailAddress>Dave@Epsilon.com</EmailAddress>
    </AcknowledgementsTo>
    <ReportType>ListActivity</ReportType>
    <ExportName>Subscriptions</ExportName>
    <Parameters>
      <Parameter>
        <Name>ActivityType</Name>
        <Value>Subscription</Value>
      </Parameter>
      <Parameter>
        <Name>ClientName</Name>
        <Value>BobsSports</Value>
      </Parameter>
      <Parameter>
        <Name>SiteName</Name>
        <Value>Hockey</Value>
      </Parameter>
      <Parameter>
        <Name>FromDate</Name>
        <Value>2002-01-01</Value>
      </Parameter>
      <Parameter>
        <Name>EndDate</Name>
        <Value>2002-03-31</Value>
      </Parameter>
      <Parameter>
        <Name>IncludeField</Name>
        <Value>ALL</Value>
      </Parameter>
    </Parameters>
  </ExportReport>
</DMConnect>
```

Output for a ListActivity report request

The fields in this report are explained in the table below.

TABLE: ListActivity report table

Field Name	Description
Client Name	The name of the client that contains the mailing lists.
Site Name	The name of the site that contains the mailing lists.
List Name	The names of the mailing lists used at a given site.
EmailAddr	The email address that subscribed or unsubscribed.
Date/Time	The date and time of the subscription or unsubscription. When you open the report in a spreadsheet, this cell is identified as a date type that you can format according to your preference.
Custom Profile Field	The name of all the profile fields that are updated at the site level.

EmailAddressChanges Report Request

The *EmailAddressChanges* report request provides the information about subscribers who changed their email addresses during a specified period.

For a description of the elements that can be used, see [ExportReport Request](#).

Parameters for EmailAddressChanges report request

The table below shows the mandatory parameters for this report.

TABLE: Description of the parameters in a *EmailAddressChanges* report request

Parameter Name	Description	Nesting
ClientName	The name of the client.	This must be nested within a <i>Name</i> element.
SiteName	The name of the site from which you want the report data.	This must be nested within a <i>Name</i> element.
FromDate	The first date of the interval for which you want the report data. For information about acceptable date formats, see FromDate and EndDate Parameters .	This must be nested within a <i>Name</i> element.
EndDate	The last date of the interval for which you want the report data. For information about acceptable date formats, see FromDate and EndDate Parameters .	This must be nested within a <i>Name</i> element.

Note: The *FromDate* and *EndDate* parameters are inclusive. This means that the date range that you specify includes the data of both the starting and the ending dates that you set.

Example of an EmailAddressChanges request

```
<DMConnect>
  <ExportReport>
    <AcknowledgementsTo>
      <EmailAddress>Bob@Epsilon.com</EmailAddress>
      <EmailAddress>Dave@Epsilon.com</EmailAddress>
    </AcknowledgementsTo>
    <ReportType>EmailAddressChanges</ReportType>
    <ExportName>Address Changes</ExportName>
    <Parameters>
      <Parameter>
        <Name>ActivityType</Name>
        <Value>Subscription</Value>
      </Parameter>
    </Parameters>
  </ExportReport>
</DMConnect>
```



```
</Parameter>
<Parameter>
  <Name>ClientName</Name>
  <Value>BobsSports</Value>
</Parameter>
<Parameter>
  <Name>SiteName</Name>
  <Value>Hockey</Value>
</Parameter>
<Parameter>
  <Name>FromDate</Name>
  <Value>2002-01-01</Value>
</Parameter>
<Parameter>
  <Name>EndDate</Name>
  <Value>2002-03-31</Value>
</Parameter>
</ExportReport>
</DMConnect>
```

Output for EmailAddress Change report

The fields in this report are explained in the table below.

TABLE: EmailAddressChange report table

Field Name	Description
Client Name	The name of the client.
Site Name	The name of the site that contains the subscribers.
Old Email	The subscriber's old email address.
New Email	The subscriber's new email address.
Date / Time	The date and time that the change was made. When you open the report in a spreadsheet, this cell is identified as a date type that you can format according to your preference.

ExportList Report Request

The *Export List* report request provides information about the specified list.

For a description of the elements that can be used, see [ExportReport Request](#).

Additional Elements in the Export List report request

The table below shows the additional elements for this report.

TABLE: Description of the additional elements in a *Export List* report request

Parameter Name	Description	Nesting
<ExportFileDetails>	The details of the file in which exported report will be saved in the <i>My Data</i> directory.	Optional. This element must be nested within an <i>ExportReport</i> element.
<ExportFileName>	The name of the file in which exported report will be saved in the <i>My Data</i> directory. If not specified default file name will be same as the name of Export job.	This element must be nested within an <i>ExportFileDetails</i> element.
<ExportFileFormat>	The format of the file in which exported report will be saved in the <i>My Data</i> directory: <ul style="list-style-type: none"> • CSV • Tab • Pipe • Double Pipe If not specified default format is CSV.	This element must be nested within an <i>ExportFileDetails</i> element.
<Compression>	The compression of the file in which exported report will be saved in the <i>My Data</i> directory: <ul style="list-style-type: none"> • Zip If not specified default the compression is Zip.	This element must be nested within an <i>ExportFileDetails</i> element.

Parameters for ExportList report request

The table below shows the mandatory parameters for this report.

TABLE: Description of the parameters in a *ExportList* report request

Parameter Name	Description	Nesting
ClientName	The name of the client.	This must be nested within a <i>Name</i> element.

TABLE: Description of the parameters in a *ExportList* report request

Parameter Name	Description	Nesting
SiteName	The name of the site from which you want the report data.	This must be nested within a <i>Name</i> element.
ListName	The name of the list from which you want the report data.	This must be nested within a <i>Name</i> element.

Example of a ExportList request

```

<DMConnect>
<ExportReport>
<AcknowledgementsTo>
<EmailAddress>Bob@doubleclick.com</EmailAddress>
<EmailAddress>Dave@doubleclick.com</EmailAddress>
</AcknowledgementsTo>
<ReportType>ExportList</ReportType>
<ExportName>2002Q1Delivery345612</ExportName>
  <Parameters>
    <Parameter>
      <Name>ClientName</Name>
      <Value>AClient_1</Value>
    </Parameter>
    <Parameter>
      <Name>SiteName</Name>
      <Value>AClient_1_S1</Value>
    </Parameter>
    <Parameter>
      <Name>ListName</Name>
      <Value>BBList</Value>
    </Parameter>
  </Parameters>
<ExportFileDetails>
<ExportFileName>export_file_name_1</ExportFileName>
<ExportFileFormat>CSV</ExportFileFormat>
  <Compression>Zip</Compression>
</ExportFileDetails>
</ExportReport>
</DMConnect>

```

Output for ExportList report request

The fields in this report are explained in the table below.

TABLE: ExportList report output table

Field Name	Descriptions
MemberID	The member id of the member.
EmailAddr	The member's email address.
Invalid	This indicates whether the address has been marked as invalid.
Client System Profile Fields	These are the various system profile fields set up at the client level.
Client Custom Profile Fields	These are the various custom profile fields set up at the client level.
Site System Profile Fields	These are the various system profile fields set up at the site level.
Site Custom Profile Fields	These are the various custom profile fields set up at the site level.

ProfileChanges Report Request

The *ProfileChanges* report request provides information about subscribers who changed their profiles during a specified period.

For a description of the elements that can be used, see [ExportReport Request](#).

Parameters for ProfileChanges report request

The table below shows the mandatory parameters for this report.

TABLE: Description of the parameters in a *ProfileChanges* report request

Parameter Name	Description	Nesting
ClientName	The name of the client.	This must be nested within a <i>Name</i> element.
SiteName	The name of the site from which you want the report data.	This must be nested within a <i>Name</i> element.
FromDate	The first date of the interval for which you want the report data. For information about acceptable date formats, see FromDate and EndDate Parameters .	This must be nested within a <i>Name</i> element.
EndDate	The last date of the interval for which you want the report data. For information about acceptable date formats, see FromDate and EndDate Parameters .	This must be nested within a <i>Name</i> element.
IncludeField	<p>The profile fields that you want to include in your report. The accepted values are:</p> <ul style="list-style-type: none">Names of valid profile fields<i>ALL</i> – includes all profile fields<i>NONE</i> – excludes all profile fields <p>Note: The values <i>ALL</i> and <i>NONE</i> override any fields that you have listed. For example, if you specify the <i>FirstName</i> field and <i>NONE</i>, no fields are listed in the report. If you specify <i>FirstName</i> and <i>ALL</i>, then all fields are listed, and <i>FirstName</i> appears only once.</p>	This must be nested within a <i>Name</i> element.

Note: The *FromDate* and *EndDate* parameters are inclusive. This means that the date range that you specify includes the data of both the starting and the ending dates that you set.

Parameters for profile fields can be included in a request one of three ways. The following three examples show the possible solutions.

Example 1:

```
<Parameters>
  <Parameter>
    <Name>IncludeField</Name>
    <Value>Field1</Value>
  </Parameter>
  ...
  <Parameter>
    <Name>IncludeField</Name>
    <Value>Fieldn</Value>
  </Parameter>
```

Example 2:

```
<Parameters>
  <Parameter>
    <Name>IncludeField</Name>
    <Value>ALL</Value>
  </Parameter>
</Parameters>
```

Example 3:

```
<Parameters>
  <Parameter>
    <Name>IncludeField</Name>
    <Value>NONE</Value>
  </Parameter>
</Parameters>
```

**Example of a
ProfileChanges
request**

```
<DMConnect>
  <ExportReport>
    <AcknowledgementsTo>
      <EmailAddress>Bob@Epsilon.com</EmailAddress>
      <EmailAddress>Dave@Epsilon.com</EmailAddress>
    </AcknowledgementsTo>
    <ReportType>ProfileChanges</ReportType>
    <ExportName>Profile Changes</ExportName>
    <Parameters>
      <Parameter>
        <Name>ClientName</Name>
        <Value>BobsSports</Value>
```

```

</Parameter>
<Parameter>
  <Name>SiteName</Name>
  <Value>Hockey</Value>
</Parameter>
<Parameter>
  <Name>FromDate</Name>
  <Value>2002-01-01</Value>
</Parameter>
<Parameter>
  <Name>EndDate</Name>
  <Value>2002-03-30</Value>
</Parameter>
<Parameter>
  <Name>IncludeField</Name>
  <Value>ALL</Value>
</Parameter>
</Parameters>
</ExportReport>
</DMConnect>

```

Output for ProfileChanges report

The fields in this report are explained in the table below.

TABLE: ProfileChanges report output table

Field Name	Descriptions
Client Name	The name of the client
Site Name	The name of the site that contains the subscribers.
Email Address	The subscriber's email address.
Invalid	This indicates whether the address has been marked as invalid.
HTML Capable	This indicates whether the address has been marked as being HTML capable.
Unsubscribed	This indicates whether the subscriber is marked as subscribed or unsubscribed.
CLC Status	This indicates whether <i>Closed Loop Confirmation</i> is being used.
Custom Profile Fields	These are the various custom profile fields set up at the site level.
Date / Time	The time and date that the changes were made. When you open the report in a spreadsheet, this cell is identified as a date type that you can format according to your preference.

RTMDelivered Report Request

The *RTMDelivered* report request provides information about the *RTM* transactions that resulted in messages being delivered during a specified period. For a description of the elements that can be used, see [ExportReport Request](#).

Parameters for RTMDelivered report request

The table below shows the parameters for this report. All these elements are nested within the *Name* element.

TABLE: Description of the parameters in an *RTMDelivered* report request

Parameter Name	Description
ClientName	The name of the client.
SiteName	The site from which you want the report data.
CampaignName	Optional. The campaign from which you want the report data.
MessageName	Optional. The <i>RTM</i> message you want to report on.
Start Date Time	The first time and date for which you want to report data. This is in the format YYYY/MM/DD HH:MM:SS or YYYY-MM-DD HH:MM:SS.
Duration	The length of time you want to report on. This must be a whole hour between 1 and 24.

Output for RTMDelivered report

This table explains the output of this report.

TABLE: *RTMDelivered* report output table

Field Name	Descriptions
Client Name	The client from which the data was pulled.
Site Name	The site from which the data was pulled.
Campaign Name	The campaign for the transaction.
Mailing Name	The <i>RTM</i> Message for the transaction.
Email Address	The email address the <i>RTM</i> message was sent to.
Transaction ID	The transaction ID for the transaction.
Date/Time	The date and time the <i>RTM</i> message was delivered.

RTMNotDelivered Report Request

The *RTMNotDelivered* report request provides information about the *RTM* transactions that take place during a 24-hour period. It provides the number of messages that were sent as a result of the transactions and the number of messages that were not sent due to the suppression of recipients' email addresses in the *DREAMmail* database. For a description of the elements that can be used, see [ExportReport Request](#).

Parameters for RTMNotDelivered report request

The table shows the parameters for this report. All these elements must be nested within the *Name* element.

TABLE: Description of the parameters in an *RTMNotDelivered* report request

Parameter Name	Description
ClientName	The name of the client.
SiteName	The site from which you want the report data.
CampaignName	Optional. The campaign from which you want the report data.
MessageName	Optional. The <i>RTM</i> message you want to report on.
Start Date Time	The first time and date for which you want to report data. This is in the format YYYY/MM/DD HH:MM:SS or YYYY-MM-DD HH:MM:SS.
Duration	The length of time you want to report on. This must be a whole hour between 1 and 24.

Output for RTMNotDelivered report

This table explains the output of this report.

TABLE: *RTMNotDelivered* report output table

Field Name	Descriptions
Triggered	The sum of the successful <i>RTM</i> transactions and <i>RTM</i> transactions that were unsuccessful due to the suppression of recipients' email addresses in the <i>DREAMmail</i> database.
Pushed	The number of successful <i>RTM</i> transactions.
Delivered	The number of times that an <i>RTM</i> message was delivered.
Invalid Addresses	The number of times that an <i>RTM</i> message could not be delivered because the email address was incomplete or syntactically invalid.
Bouncebacks	The number of times that <i>RTM</i> messages bounced (hard or soft).

TABLE: *RTMNotDelivered* report output table

Field Name	Descriptions
Suppressed	The number of unsuccessful <i>RTM</i> transactions due to the suppression of recipients' email addresses in the <i>DREAMmail</i> database.
Unsubscribed	The number of unsuccessful <i>RTM</i> transactions due to recipients having unsubscribed from the <i>DREAMmail</i> site or client that is sending the message.
Invalid	The number of unsuccessful <i>RTM</i> transactions due to the email addresses of the recipients being identified as invalid in the <i>DREAMmail</i> database.
Deferral Passed Time	The number of unsuccessful <i>RTM</i> transactions due to the <i>RTM</i> message not being sent because the XML message request had specified not to send the message after the waiting period had passed.
Other Reason	The number of unsuccessful <i>RTM</i> transactions due to reasons other than the suppression of the message, unsubscription of the subscriber, invalid message, or expiration of the deferral time.

RTMClicks Report Request

The *RTMClicks* report request provides information about clicks that occurred on *RTM* messages for during a specified period. For a description of the elements that can be used, see [ExportReport Request](#).

Parameters for RTMClicks report request

The table shows the parameters for this report. All of these parameters must be nested within a *Name* element.

TABLE: Description of the parameters in an *RTMClicks* report request

Parameter Name	Description
ClientName	The name of the client.
SiteName	The site from which you want the report data.
CampaignName	Optional. The campaign from which you want the report data.
MessageName	Optional. The <i>RTM</i> message you want to report on.
Start Date Time	The first time and date for which you want to report data. This is in the format YYYY/MM/DD HH:MM:SS or YYYY-MM-DD HH:MM:SS.
Duration	The length of time you want to report on. This must be a whole hour between 1 and 24.

Output for RTMClicks report

This table explains the output of this report.

TABLE: RTMClicks report output table

Field Name	Descriptions
Client Name	The client from which the data was pulled.
Site Name	The site from which the data was pulled.
Campaign Name	The campaign for the transaction.
Mailing Name	The <i>RTM</i> Message for the transaction.
Email Address	The email address the <i>RTM</i> message was sent to.
Transaction ID	The transaction ID for the transaction.
URL	The URL clicked on.
URL Label	The label for the URL clicked on.
Hit From	The message format for which the click occurred.

TABLE: RTMClicks report output table

Field Name	Descriptions
Pass Along	This shows if the click was from the original recipient of the message or was from another (pass along) recipient. 1 indicates this was from a pass along.
Date/Time	The date and time the clickthrough occurred.

RTMHTMLOpens Report Request

The *RTMHTMLOpens* report request provides information about HTML opens that occurred on *RTM* messages during a specified period. For a description of the elements that can be used, see [ExportReport Request](#).

Parameters for RTMHTMLOpens report request

The table shows the parameters for this report. All these parameters must be nested within the *Name* element.

TABLE: Description of the parameters in an *RTMHTMLOpens* report request

Parameter Name	Description
ClientName	The name of the client.
SiteName	The site from which you want the report data.
CampaignName	Optional. The campaign from which you want the report data.
MessageName	Optional. The <i>RTM</i> message you want to report on.
Start Date Time	The first time and date for which you want to report data. This is in the format YYYY/MM/DD HH:MM:SS or YYYY-MM-DD HH:MM:SS.
Duration	The length of time you want to report on. This must be a whole hour between 1 and 24.

Output for RTMHTMLOpens report

This table explains the output of this report.

TABLE: RTMHTMLOpens report output table

Field Name	Descriptions
Client Name	The client from which the data was pulled.
Site Name	The site from which the data was pulled.
Campaign Name	The campaign for the transaction.
Mailing Name	The <i>RTM</i> Message for the transaction.
Email Address	The email address the <i>RTM</i> message was sent to.
Transaction ID	The transaction ID for the transaction.
Format	The format of the opened message, for example, HTML or AOL.
Date/Time	The date and time the HTML open occurred.

RTMSpotlight Report Request

The *RTMSpotlight* report request provides information about *Spotlight* transactions that occurred on *RTM* messages during a specified period.

For a description of the elements that can be used, see [ExportReport Request](#).

Parameters for RTMSpotlight report request

This table shows the parameters for this report. All these parameters must be nested within the *Name* element.

TABLE: Description of the parameters in an *RTMSpotlight* report request

Parameter Name	Description
ClientName	The name of the client.
SiteName	The site from which you want the report data.
CampaignName	Optional. The campaign from which you want the report data.
MessageName	Optional. The <i>RTM</i> message you want to report on.
FromDate	The date and time for which you want to report data. This is in the format YYYY/MM/DD HH:MM:SS or YYYY-MM-DD HH:MM:SS.
Duration	The length of time you want to report on. This must be a whole hour between 1 and 24.

Output for RTMSpotlight report

This table explains the output of this report.

TABLE: RTMSpotlight report output table

Field Name	Descriptions
Client Name	The client from which the data was pulled.
Site Name	The site from which the data was pulled.
Campaign Name	The campaign for the transaction.
Mailing Name	The <i>RTM</i> Message for the transaction.
Email Address	The email address the <i>RTM</i> message was sent to.
Transaction ID	The transaction ID for the transaction.
Type	The <i>Spotlight</i> transaction type; event or purchase.
Category	The category for the <i>Spotlight</i> transaction.

TABLE: RTMSpotlight report output table

Field Name	Descriptions
Subcategory	The subcategory for the <i>Spotlight</i> transaction.
Quantity	The quantity for the <i>Spotlight</i> transaction.
Amount	The amount for the <i>Spotlight</i> transaction.
OrderID	The Order ID for the <i>Spotlight</i> transaction.
AdID	The Ad ID for the <i>Spotlight</i> transaction.
Date/Time	The date and time of the <i>Spotlight</i> transaction.

Unsubscription Report Request

The *Unsubscription* report request provides information about unsubscriptions that occurred on during a specified period for a Client or Site scope.

For a description of the elements that can be used, see [ExportReport Request](#).

Additional elements for Unsubscription report request

This table shows the additional elements for this report.

TABLE: Description of the additional elements in a *Unsubscriptions* report request

Parameter Name	Description	Nesting
<ExportScope>	The scope of the report to be exported: <ul style="list-style-type: none">• Client• Site If not specified default scope will be "Client".	Optional. This element must be nested within an <i>ExportReport</i> element.
<ExportFileDetails>	The details of the file in which exported report will be saved in the <i>My Data</i> directory.	Optional. This element must be nested within an <i>ExportReport</i> element.
<ExportFileName>	The name of the file in which exported report will be saved in the <i>My Data</i> directory. If not specified default file name will be same as the name of Export job.	This element must be nested within an <i>ExportFileDetails</i> element.
<ExportFileFormat>	The format of the file in which exported report will be saved in the <i>My Data</i> directory: <ul style="list-style-type: none">• CSV• Tab• Pipe• Double Pipe If not specified default format is CSV.	This element must be nested within an <i>ExportFileDetails</i> element.
<Compression>	The compression of the file in which exported report will be saved in the <i>My Data</i> directory: <ul style="list-style-type: none">• Zip If not specified the default compression is Zip.	This element must be nested within an <i>ExportFileDetails</i> element

Parameters for Unsubscriptions report request

The table below shows the mandatory parameters for this report.

TABLE: Description of the parameters in a *Unsubscriptions* report request

Parameter Name	Description	Nesting
ClientName	The name of the client.	This must be nested within a <i>Name</i> element.
SiteName	The name of the site from which you want the report data.	This must be nested within a <i>Name</i> element.
FromDate	The first date of the interval for which you want the report data. For information about acceptable date formats, see FromDate and EndDate Parameters .	This must be nested within a <i>Name</i> element.
EndDate	The last date of the interval for which you want the report data. For information about acceptable date formats, see FromDate and EndDate Parameters .	This must be nested within a <i>Name</i> element.

Note: The *FromDate* and *EndDate* parameters are inclusive. This means that the date range that you specify includes the data of both the starting and the ending dates that you set.

Example of a Unsubscriptions request

```
<DMConnect>
<ExportReport>
<AcknowledgementsTo>
<EmailAddress>Bob@doubleclick.com</EmailAddress>
<EmailAddress>Dave@doubleclick.com</EmailAddress>
</AcknowledgementsTo>
<ReportType>Unsubscriptions</ReportType>
<ExportName>2001Q1Unsub</ExportName>
<ExportScope> Site</ExportScope>
<Parameters>
<Parameter>
<Name>ClientName</Name>
<Value>BobsSports</Value>
</Parameter>
<Parameter>
<Name>SiteName</Name>
<Value>Volleyball </Value>
</Parameter>
```

```

<Parameter>
<Name>FromDate</Name>
<Value>2001-01-01</Value>
</Parameter>
<Parameter>
<Name>EndDate</Name>
<Value>2001-03-30 </Value>
</Parameter>
</Parameters>
<ExportFileDetails>
    <ExportFileName>file_name</ExportFileName>
    <ExportFileFormat>CSV</ExportFileFormat>
    <Compression>Zip</Compression>
</ExportFileDetails>
</ExportReport>
</DMConnect>

```

Output for Unsubscriptions report request

The fields in this report are explained in the table below.

TABLE: Unsubscriptions report output table

Field Name	Descriptions
EmailAddr	The subscriber's email address.
ClientName	The name of the client, or the client containing the site or list which the subscriber unsubscribed from.
SiteName	The name of the site, or the site containing the list that the subscriber unsubscribed from. This will be empty for the Client Unsub UnsubType
ListName	The list from which the subscriber unsubscribed. This will be empty for the Client or Site Unsub UnsubType.
Source	The method of unsubscription. This can be Email (From Address, Unsubscription Address, Subscription Address), List Generator, Subscriber Manager.
Datetime	The time and date that the unsubscription occurred.

POSTING *DMCONNECT* REQUESTS

DMConnect processes requests through HTTP or HTTPS posts. You can create the HTTP post in any language, as long as the HTTP form parameters are included.

This section discusses the following topics:

- [Request Form Parameters](#)
- [Sending Requests through HTTP Posts.](#)

Request Form Parameters

DMConnect accepts requests through HTTP posts. You must include certain HTTP form parameters for each post.

This section discusses:

- [Form parameters](#)
- [DMConnect 3.5](#)
- [Encode form variables](#)
- [VBScript to handle URLEncoding](#).

Form parameters

Requests made by your application must provide values for the HTTP form parameters listed in the table below.

TABLE: HTTP form parameter descriptions

Form Parameter	Description	Comments
Servename	The name of the <i>DREAMmail</i> server that handles the request. The <i>DREAMmail</i> servename is ugox, where x is the server number; for example ugo1 or ugo10.	The <i>DMConnect</i> server uses this information to forward your request to the appropriate server. <i>Servename</i> must be the name of the server on which your database is located. Check with your <i>DREAMmail</i> representative for this information.
Username	The name of the <i>DREAMmail</i> user account under which the request is processed.	The <i>Username</i> is the same name used to log into <i>DREAMmail</i> . Note: The user must have permission to use <i>DMConnect</i> .
Password	The <i>DREAMmail</i> password for the user specified by the Username form variable.	The <i>Password</i> is the same password used to log into <i>DREAMmail</i> .
AcknowledgementTo	The email address of the requester, to which all acknowledgements are sent. There can be only one address in this field.	Optional variable. The <i>AcknowledgementTo</i> variable must contain a valid email address where an email can be sent once a <i>DMConnect</i> request has been received and properly processed.

TABLE: HTTP form parameter descriptions

Form Parameter	Description	Comments
XML	The XML specification of the content to be accepted in the format.	The XML variable contains the DOCTYPE line as well as the XML-based <i>DMConnect</i> request. The DOCTYPE line must be included first in every <i>DMConnect</i> request.
Version	The version of <i>DMConnect</i> that processes the request. <ul style="list-style-type: none"> 4.0 3.5 3.1 3.0 2.5. 	The <i>Version</i> variable must contain the version of <i>DMConnect</i> that is being used to process the XML code. Note: For any version of DREAMmail above 4.0, specify 4.0 as the version.
RequestType	The type of request that is submitted to DMConnect. One of two values: <ul style="list-style-type: none"> Synchronous: client application waits for the request to complete before receiving a response to the request. Asynchronous – client application receives an immediate response with a JobId, while the request may still be getting processed. 	ImportList, UpdateProfile, and Unsubscribe requests can only be Asynchronous, and RequestType defaults to Asynchronous if no value is specified. Any request starting with “Get” can only be Synchronous, and RequestType defaults to Synchronous if no value is specified. All other requests can be submitted using either method, with the default as Synchronous. Note: The JobId is used to make a subsequent request to get the status of the job and the result.

DMConnect 3.5

For requests that specify a value of 3.5, the XML document must be fully validated against the 3.5 DTD. The XML document must include the following element, which identifies the version of the DTD:

```
<!DOCTYPE DMConnect SYSTEM "DMConnect35.dtd">
```

The root element with the XML document is <DMConnect>.

DMConnect 4.0 and greater

As of the [DREAMmail 4.0](#) release, HTTP posts no longer have to specify the version of the *DMConnect* DTD. All HTTP posts must now specify *DMConnect.dtd*.

Encode form variables

The form variables must be valid for submitting via HTTP post. The form variables must be URL-encoded depending on the programming language and HTTP posting object being used. URL encoding is covered in RFC1866 Chapter 8.2.1 specification. A sample Visual Basic routine to perform this URL encoding is below.

Note: The code has not been optimized for performance.

VBScript to handle URLEncoding

```
'This function is used to encode URL form data per the RFC1866 Chapter 8.2.1 specification
```

```
'Replaces occurrences of ' ' with +, and all non-alphanumeric
'characters with %hh where hh is the hexadecimal ascii code
'of the escaped character.
```

```
'loop through string as an array of bytes
'for each character, convert spaces to + and non-alphanumeric characters
'to hexadecimal representation
'resulting string is at most 3 times the size of the Input string
(source)
```

```
Dim SourceLength As Long
```

```
SourceLength = Len(Source)
```

```
ReDim WorkString(SourceLength) As String
```

```
ReDim ResultArray((SourceLength * 3) + 1) As String
```

```
Dim TempLeft, TempRight, TempString As String
```

```
Dim StrAscii As Integer
```

```
Dim StrHex As String
```

```
Dim Count As Long
```

```
'First setup WorkString as an array of characters of Source string
TempString = Source
For a = 1 To SourceLength
    TempLeft = Left$(TempString, 1)
    WorkString(a) = TempLeft
    If Not a = SourceLength Then
        TempRight = Right$(TempString, (Len(TempString) - 1))
        TempString = TempRight
    End If
Next

'Now step through WorkString and set ResultArray per RFC specs.
Count = 1
For b = 1 To SourceLength
    StrAscii = Asc(WorkString(b)) 'Determine ASCII value of character
    If WorkString(b) = " " Then
        ResultArray(Count) = "+"
        Count = Count + 1
    ElseIf IsNumeric(WorkString(b)) Then
        ResultArray(Count) = WorkString(b)
        Count = Count + 1
    ElseIf (StrAscii >= 65 And StrAscii <= 90) Or (StrAscii >= 97 And
        StrAscii <= 122) Then
        ResultArray(Count) = WorkString(b)
        Count = Count + 1
    Else 'convert string to two digit hex code of string ascii value
        StrHex = Hex$(StrAscii)
        ResultArray(Count) = "%"
        ResultArray(Count + 1) = Left$(StrHex, 1)
        ResultArray(Count + 2) = Right$(StrHex, 1)
        Count = Count + 3
    End If
Next

'Now the long int Count contains the length of the ResultArray in
'characters
'and the ResultArray array contains a URLEncode representation of the
'SourceString

'Return the array as AccessConstants string to the Function

Dim ResultString As String

For c = 1 To Count

    ResultString = ResultString & ResultArray(c)

Next
```

```
URLEncode = ResultString  
  
End Function
```


Sending Requests through HTTP Posts

You can use various languages to send a request via HTTP post. This section provides examples of posts in three formats:

- [HTML form to post to DMConnect](#)
- [PERL script to handle HTTP post of XML request](#)
- [C++ to handle HTTP post of XML request.](#)

HTML form to post to DMConnect

This sample HTML form used to post to *DMConnect* is very basic and is used only for testing.

```
<HTML>
<HEAD>
<TITLE>DMConnect Trigger</TITLE>
</HEAD>

<BODY bgColor=#7b94b3>
<FORM action= http://dmconnect.na.epidm.net/dmconnect/dmconnect.exe
method=post
      name="XML Trigger">
<H3>DMConnect Test Form</H3>
<HR>
<P>
Server Name: <INPUT name=Servername size=50 value=(name of server
used)><BR>
User Name: <INPUT name=Username size=50><BR>
Password: <INPUT name=Password size=40 type=password><BR>
Acknowledgement To: <INPUT name=AcknowledgementTo size=40><BR>
Version: <INPUT name=Version size=40 value="3.5">
<P>
XML Stream:<BR>
<TEXTAREA cols=70 name=XML rows=18>
<!--!DOCTYPE DMConnect SYSTEM "DMConnect35.dtd">
</TEXTAREA>
<P>
<INPUT name=SubButton type=submit value="Send to DMConnect">
</FORM>
</BODY>
</HTML>
```

PERL script to handle HTTP post of XML request

You can use PERL to send the request via an HTTP post.

```
#!/usr/bin/perl

use HTTP::Request::Common;
use LWP::UserAgent;

if ($#ARGV < 3 )
{
    die "Not enough data\n";
}

$username = shift @ARGV;
$password = shift @ARGV;
$acknowledge = shift @ARGV;
$xml_data_file = shift @ARGV;
$post_url =
"http://dmconnect.ddc.dartmail.net/dmconnect/dmconnect.exe";
$DMConnect_server = "rddemosql1";

$result_xml_file = "/var/tmp/result.xml";

open (DATA, "$xml_data_file") ||
    die "Error with $xml_data_file. $_ ";

while (<DATA>)
{
    $xml_code .= $_;
}
close DATA;

$ua= LWP::UserAgent -> new();
$response=$ua->request(POST '$post_url',
    [ Servername => $DMConnect_server,
      Username => $username,
      Password => $password,
      AcknowledgementTo => $acknowledgeto,
      XML => $xml_code,
      Version => '3.5' ]);

open (OUT,"> $result_xml_file") ||
    die "Error with $result_xml_file. $_ \n";

while ( ($key, $value) = each %response)
{
    if ($key =~ /^_content$/)
    {

```

```
printf OUT "$value\n";
    }
}
close OUT;
```

C++ to handle HTTP post of XML request

You can use C++ to post *DMConnect* requests via HTTP.

```
// HttpPost.cpp : Sample Http post code
//

#include <Windows.h>
#include <wininet.h>
#include <string>
#include <fstream.h>
#include <time.h>

//
// UrlEncode
//
// Encodes the passed string to the encoded format defined by
// RFC1866 section 8.2.1.
//
// On return the argument is modified to contain the URL
// encoded string.
//
void UrlEncode(std::string& str)
{
    std::string workStr;

    const int WORK_BUFFER_SIZE = 100 * 1024;
    int workBufferLen;
    char workBuffer[WORK_BUFFER_SIZE+1];

    // Replaces occurrences of ' ' with +, and all non-alphanumeric
    // characters with %hh where hh is the hexadecimal ascii code
    // of the escaped character.
    workStr = "";
    workBufferLen = 0;

    const char *p;
    const char *endP;

    p = str.c_str();
    endP = p + str.length();
```

```
for (; p < endP; p++)
{
    if (workBufferLen > WORK_BUFFER_SIZE - 4)
    {
        workBuffer[workBufferLen] = '\0';
        workStr += workBuffer;
        workBufferLen = 0;
    }

    unsigned char c = *p;
    if (c == ' ')
    {
        workBuffer[workBufferLen] = '+';
        workBufferLen++;
    }
    else if (isalnum(c))
    {
        workBuffer[workBufferLen] = c;
        workBufferLen++;
    }
    else
    {
        char hexBuff[4];
        if (c < 0x0f)
        {
            sprintf(hexBuff, "%0%X", c);
        }
        else
        {
            sprintf(hexBuff, "%X%X", c);
        }
        strncpy(&workBuffer[workBufferLen], hexBuff, 3);
        workBufferLen += 3;
    }
}

workBuffer[workBufferLen] = '\0';
workStr += workBuffer;
workBufferLen = 0;

str = workStr;
}

//
// UrlEncode
//
// Encodes the passed string to the encoded format defined by
```

```
// RFC1866 section 8.2.1.
//
// The source is taken from str, and the encoded string is written
// to dst. This requires that the storage allocated to dst must
// be at least three times the length of the string stored in
// str.
//
void UrlEncode(const char *str, char *dst)
{
    int workBufferLen;
    char *workBuffer = dst;

    // Replaces occurrences of ' ' with +, and all non-alphanumeric
    // characters with %hh where hh is the hexadecimal ascii code
    // of the escaped character.
    workBufferLen = 0;

    const char *p;
    const char *endP;

    p = str;
    endP = p + strlen(p);

    for (; p < endP; p++)
    {
        unsigned char c = *p;
        if (c == ' ')
        {
            workBuffer[workBufferLen] = '+';
            workBufferLen++;
        }
        else if (isalnum(c))
        {
            workBuffer[workBufferLen] = c;
            workBufferLen++;
        }
        else
        {
            char hexBuff[4];
            if (c < 0x0f)
            {
                sprintf(hexBuff, "%0%X", c);
            }
            else
            {
                sprintf(hexBuff, "%X", c);
            }
        }
    }
}
```

```

        }
        strncpy(&workBuffer[workBufferLen], hexBuff, 3);
        workBufferLen += 3;
    }
}

workBuffer[workBufferLen] = '\\0';
return;
}

//
// x2c
//
// Maps a 2-digit hex string to the corresponding integer.
char x2c(char *what)
{
    char digit;

    digit = (what[0] >= 'A' ? ((what[0] & 0xdf) - 'A')+10 : (what[0] -
'0'));
    digit *= 16;
    digit += (what[1] >= 'A' ? ((what[1] & 0xdf) - 'A')+10 : (what[1] -
'0'));
    return(digit);
}

//
// UrlDecode
//
// Decodes the passed string to the encoded format defined by
// RFC1866 section 8.2.1.
//
// On return the argument is modified to contain the unengaged
// string.
//
void UrlDecode(std::string& str)
{
    int i;
    std::string workStr;

    const int WORK_BUFFER_SIZE = 10 * 1024;
    int workBufferLen;
    char workBuffer[WORK_BUFFER_SIZE+1];

    // Replaces occurrences of '+' with ' ', and %HH with the
    // corresponding
    // ascii code.

```

```
workStr = "";
workBufferLen = 0;

for (i=0; i < str.length(); i++)
{
    if (workBufferLen > WORK_BUFFER_SIZE - 4)
    {
        workBuffer[workBufferLen] = '\\0';
        workStr += workBuffer;
        workBufferLen = 0;
    }

    int c = str[i];
    if (c == '+')
    {
        workBuffer[workBufferLen] = ' ';
        workBufferLen++;
    }
    else if (c == '%')
    {
        char k;

        i++;
        k = x2c(&str[i]);
        i++;

        workBuffer[workBufferLen] = k;
        workBufferLen++;
    }
    else
    {
        workBuffer[workBufferLen] = c;
        workBufferLen++;
    }
}

workBuffer[workBufferLen] = '\\0';
workStr += workBuffer;
workBufferLen = 0;

str = workStr;
}

//
// HttpPost
//
// Performs an HTTP post to the specified hostname and action,
```

```
// passing ServerName ('DREAMmailServerName'), UserName
('DREAMmailUserName'),
// Password ('Password'), AcknowledgementsTo ('ackTo')
// and XML ('xmlData') as form parameters.
//
// The remote server's response is returned in the parameter
// 'response'.
//
bool HttpPost(const char *hostname, const char *formAction,
              const char *DREAMmailServerName,
              const char *DREAMmailUserName, const char *DREAMmailPassword,
              const char *ackTo,
              const char *xmlData,
              std::string& response)
{
    std::string formData;
    std::string urlEncodedStr;

    static TCHAR hdrs[] = "Content-Type: application/x-www-form-
urlencoded";
    static TCHAR accept[] = "Accept: */*";

    formData ="Servername=";
    urlEncodedStr = DREAMmailServerName;
    UrlEncode(urlEncodedStr);
    formData += urlEncodedStr;
    formData += "&";

    formData += "Username=";
    urlEncodedStr = DREAMmailUserName;
    UrlEncode(urlEncodedStr);
    formData += urlEncodedStr;
    formData += "&";

    formData += "Password=";
    urlEncodedStr = DREAMmailPassword;
    UrlEncode(urlEncodedStr);
    formData += urlEncodedStr;
    formData += "&";

    formData += "AcknowledgementTo=";
    urlEncodedStr = ackTo;
    UrlEncode(urlEncodedStr);
    formData += urlEncodedStr;
    formData += "&";
}
```



```
////////////////////////////////////
//begin file conversion to LPCTSTR
////////////////////////////////////

std::string xmlStream;

// open file for reading
FILE *fpIn;
fpIn = fopen(xmlData, "rt");
if (fpIn == 0)
{
    return false;
}

int fileBytesRead, totalBytesRead;
const int READ_BUFF_SIZE = 100*1024;
char readBuff[READ_BUFF_SIZE+1];
char urlEncodedBuff[3 * READ_BUFF_SIZE + 1]; // Note this array MUST
be at least 3 times the
    // size readBuff.
std::string workBuffer;

time_t startTime, currTime;
time(&startTime);
cout << "Starting to read from file and URL encode " << xmlData <<
"\n";
cout.flush();

xmlStream = "";
totalBytesRead = 0;
while (!feof(fpIn))
{
    fileBytesRead = fread(readBuff, sizeof(char), READ_BUFF_SIZE, fpIn);
    readBuff[fileBytesRead] = '\0';

    UrlEncode(readBuff, urlEncodedBuff);
    xmlStream += urlEncodedBuff;
}

// close file
fclose(fpIn);

time(&currTime);
cout << "Finished reading from file and URL encoding " << xmlData << "
Elapsed time: " <<
    currTime-startTime << " seconds.\n";
cout.flush();
```

```
////////////////////////////////////
//END file conversion to LPCTSTR
////////////////////////////////////

    formData += "XML=";
formData += xmlStream;
    formData += "&";

    // Connect to the remote web server and perform the post.
    HINTERNET hSession = InternetOpen("SampleHttpPostAgent",
        INTERNET_OPEN_TYPE_PRECONFIG, NULL, NULL, 0);
    if (hSession == 0) return false;

    HINTERNET hConnect = InternetConnect(hSession, hostname,
        INTERNET_DEFAULT_HTTP_PORT, NULL, NULL, INTERNET_SERVICE_HTTP, 0,
        1);
    if (hConnect == 0) return false;

    HINTERNET hRequest = HttpOpenRequest(hConnect, "POST",
        formAction, NULL, NULL, (const char **) accept, 0, 1);
    if (hRequest == 0) return false;

    if (! HttpSendRequest(hRequest, hdrs, strlen(hdrs), (void *)
        formData.c_str(),
        formData.length()))
    {
        return false;
    }

    // Retrieve the response
    const int RECEIVE_BUFF_SIZE=1024;
    char receiveBuff[RECEIVE_BUFF_SIZE+1];
    unsigned long bytesRead;
    response = "";
    while (InternetReadFile(hRequest, receiveBuff, RECEIVE_BUFF_SIZE-1,
        &bytesRead))
    {
        if (bytesRead == 0)
        {
            break;
        }
        else
        {
            receiveBuff[bytesRead] = '\0';
            response += receiveBuff;
        }
    }
}
```

```
        return true;
    }

    int main(int argc, char* argv[])
    {
        std::string response;

        // Usage: httppost host URL filename

        // check args
        if (argc != 4)
        {

            printf("Usage: httppost host URL filename\n");
            exit(-1);
        }

        // declare variables

        LPCTSTR hostName, URL, filePath;

        // set arg values
        hostName = argv[1];
        URL = argv[2];
        filePath = argv[3];

        // perform post
        HttpPost("localhost", URL, hostName,
            "username", "password", "acknowledgeTo@somedomain.com",
            filePath, response);

        /*HttpPost("localhost", "/testpost.asp", "servername",
            "username", "password", "acknowledgeTo@somedomain.com",
            "<xml> request goes here </xml>",
            response);
        */

        printf("Response: %s\n", response.c_str());
        return 0;
    }
```

RESPONSES TO *DMCONNECT* REQUESTS

When you issue a *DMConnect* request, *DMConnect* responds with an XML document, which includes a success or failure code, as well as a descriptive text status line.

Epsilon recommends that you use only the response code as the basis for parsing the result of a *DMConnect* request. Do not use the descriptive text as a test condition because the wording of the text might change in future releases of *DMConnect*.

Note: Some *DMConnect* requests (such as *GetProfileFields* and *GetObjectNames*) have responses that return more data than the response code and descriptive text. For a list and description of these responses, see [Syntax of DMConnect Requests and Responses](#).

This section discusses the following topics:

- [Notification Response Message](#)
- [DMConnect Messages and Response Codes](#)
- [GetJobStatus and GetExportReportStatus Responses](#).

Notification Response Message

After *DMConnect* receives and interprets your request, it responds with an HTTP response. This response informs you of the status of your request. There are two types of notification messages:

- [General notification message](#)
- [Specific notification message](#).

General notification message

A general notification message is sent for any request that is received and processed successfully. The message is sent to email addresses included in:

- the *AcknowledgementsTo* field in the XML request
- HTTP post parameters
- configuration file.

The message contains the following information:

- name of the XML request
- XML request script
- date and time of when request was received
- *DREAMmail* server.

Specific notification message

Three *DMConnect* requests — *ImportList*, *UpdateProfile*, and *Unsubscribe* — fork a child process to complete the request. A second notification email is sent to the *AcknowledgementsTo* recipients with the success or failure response upon completion of this process.

DMConnect Messages and Response Codes

This section discusses:

- [About response messages and codes](#)
- [Format of response codes](#)
- [Response messages for asynchronous requests](#)
- [Contacting the DREAMmail support desk](#)
- [Types of user-originated errors](#)
- [Types of system and configuration errors](#)
- [Types of DMConnect response messages](#)
- [Success messages](#)
- [Error messages in HTTP post](#)
- [Error messages in the XML file](#)
- [Error messages generated by DREAMmail server](#)
- [Error messages for trigger messages](#)
- [Error messages for messages](#)
- [Error messages for RTM messages](#)
- [Error messages for reports](#)
- [DMConnect service error](#)
- [URL Update error messages](#)
- [General error messages](#)
- [Validation errors](#)
- [Security errors](#)
- [Deployment errors](#)
- [Import errors](#)
- [Direct-deployment errors.](#)

About response messages and codes

DMConnect organizes error messages by response code. A positive number under 1000 indicates that the request was successfully executed. A response code over 1000 and all negative response codes indicate that the request failed.

Note: Epsilon recommends that you use only the response code as the basis for parsing the result of a *DMConnect* request. Do not use the descriptive text as a test condition because the wording of the text might change in future releases of *DMConnect*

Response messages can be broken down into two categories:

- user-originated errors (see [Types of user-originated errors](#))
- system and configuration messages and errors (see [Types of system and configuration errors](#))

Format of response codes

The response messages are returned in XML format. Some of these response messages include a code and descriptive text. Other responses include additional information. For list of these responses, see [Responses Returned by DMConnect Requests](#).

Response messages for asynchronous requests

If you successfully submit an asynchronous request, you get the following results:

- If you post your request as version 3.0 or 3.1, response code 1 is returned. No jobname information is provided.
- If you post your request as version 3.5, response code 2 is returned along with the job name. *DMConnect* then informs you by email when the job has been successfully completed. If you want to know the status of the job, you can use the <GetJobStatus> request to find out the job's status. For more information, see [GetJobStatus Request](#) and [GetJobStatus and GetExportReportStatus Responses](#).

Contacting the DREAMmail support desk

Some error codes tell you to contact *Client Relations*. Client Relations can be reached by email 24 hours per day at dreammail@epsiloninteractive.net. You can also go to the website <http://lms.learnshare.com>.

Contact information by telephone for Client Relations is as follows:

North America: Phone: (416) 847-3223; Monday-Friday, 8am - 8pm AEST/ADST

Europe, Middle East and Africa (EMEA): Phone: +353-1-246-044; Monday-Friday, 8:00 a.m. - to 6:00 p.m. GMT

Asia and Pacific locations (APAC): Phone: +61-2-8247-3705; Monday-Friday, 8am-7/8pm AEST/ADS.

Types of user-originated errors

User-originated errors occur when there is an error in writing the scripts. There are two types of user-originated errors:

- XML error
- HTTP post errors.

XML error: An XML error can occur if you make a mistake in the XML document that you send as a *DMConnect* request. These errors include:

- incorrect XML syntax
- missing or incorrect element/node
- name of an element is invalid
- value specified for the parameter is invalid.

HTTP post errors: HTTP post errors exist in the HTTP post parameter received by *DMConnect*. They include:

- invalid *Servername*
- invalid *Username* and *password*
- invalid *AcknowledgementTo* element.

Types of system and configuration errors

Requests can also fail due to errors with configuration, hardware, or software. There are two types of system and configuration errors:

- General communication error
- Internal error.

General communication error: General communication errors result from incorrect configuration or unavailable server. They include:

- communication error with the *DREAMmail* server
- communication error with the report server
- invalid configuration of your database.

Internal error: Internal errors are caused by internal system or software failure. They include:

- failure to load information from database
- unknown software failure (*DMConnect* and DLL components).

If you have an internal error, contact *Client Relations* as soon as possible.

Types of DMConnect response messages

The system and configuration messages returned by *DMConnect* can be broken down into the following categories:

- [Success messages](#)
- [Error messages in HTTP post](#)
- [Error messages in the XML file](#)
- [Error messages generated by DREAMmail server](#)
- [Error messages for trigger messages](#)
- [Error messages for messages](#)
- [Error messages for reports](#)
- [General error messages](#)
- [Validation errors](#)
- [Security errors](#)
- [Deployment errors](#)
- [Import errors.](#)

Success messages

Messages sent when a request is successfully executed are denoted by response codes 1 through 1000. (Response codes over 1000 and all negative response codes indicate a failed request.)

TABLE: Successful message response codes

Code	Text
1	Your request was successfully received and processed.
2	Your request was successfully received. You receive a notification when once the process has been completed.
5	Your GetJobStatus request was successfully processed.
100	Your request for all export names was successfully received and processed.
101	Your request for report status was successfully received and processed.
102	Your request for report retrieval was successfully received and processed.

Error messages in HTTP post

Error messages generated by the HTTP post usually result from missing fields or incorrect information. The list of errors that can be generated by the HTTP post is shown in the table below.

TABLE: HTTP post-related response codes

Code	Text
-1	No Servername field provided in the http post.
-2	No Username field provided in the http post.
-3	No Password field provided in the http post.
-4	No AcknowledgementTo field provided in the http post.
-5	No XML field provided in the http post.
-6	No ReportServername field provided in the http post.
-7	The version supplied in the http post is not valid. Valid versions are 2.5, 3.0, 3.1 and 3.5.
-8	The request type supplied in the http post is not valid or the request does not support the type of request you supplied.
-100	The Servername provided in the http post is unknown.
-101	Login request using the specified Username/Password failed.
-102	The Username does not have DMConnect privileges.

Error messages in the XML file

Error messages generated by the XML file usually result from *a syntax error or missing or invalid content in the XML elements*. The list of errors that can be generated by the XML file is shown in the table below.

TABLE: XML-related response codes

Code	Text
-200	The XML script provided is not valid.
-201	The SiteName specified in the request does not exist or your Username has not been granted access.
-202	The CampaignName specified in the request does not exist or the campaign type does not match the type of mailing or the campaign is finished.
-203	The ListName specified in the request does not exist, is inactive or is read only.

TABLE: XML-related response codes

Code	Text
-204	The MailingName specified in the request already exists.
-205	The Username specified has not been granted permission to create mailings.
-206	An invalid MailingType was specified in the request.
-207	No text, AOL or HTML MessageContent was specified in the request.
-208	The mailing failed to start.
-209	Your Username does not have permission to perform the action requested.
-210	The ClientName specified in the request does not exist or your Username has not been granted access.
-211	The TestMailing failed to start.
-212	The SendMailingForApproval failed to start.
-213	The MailingType is invalid.
-214	The CreateMailing failed: Failure to commit the mailing.
-215	The ListName specified in the TestMailing request is not a test list.
-216	The field name specified in the ImportList/UpdateProfile request is not valid.
-217	The profile field specified is invalid.
-218	The URL tracked is not in the mailing/content body.
-219	Using system fields in profile fields is not allowed.
-220	An invalid ReportType was specified in the request.
-221	The ExportName specified in the request already exists.
-223	The ExportName specified in the request does not exist or is invalid.
-224	An invalid Parameter Name specified in the request.
-225	An invalid Date Value specified in the parameter.
-226	Date range exceeds maximum allowable range.
-227	EndDate parameter is an earlier date than FromDate.
-228	A required Parameter is missing.

TABLE: XML-related response codes

Code	Text
-229	An invalid Parameter Value specified in the request.
-230	The campaign type and mailing type do not match.
-231	No test list was specified in the TestMailing request.
-232	The campaign is not an approval or a direct mail campaign.
-233	Invalid DeliveryType or invalid DeliveryDestination was specified.
-235	Failed to load subscription address at message level.
- 237	Invalid Track Unsub value at message level.
- 238	TrackUnsub element is not a valid part of CreateRTMMailing/ UpdateRTMMailing request.
- 239	Failed to load unsubscription address at message level.
-240	The ContentName was not specified in a CreateContent or UpdateContent request.
-241	The ContentType must be either StandardContent or ScriptContent.
-242	The ContentText field is blank.
-243	The Job name is blank or the database cannot find the status of the job. Please check the job name.
-245	The template specified could not be loaded.
-246	The send rate supplied in the XML post is not valid
-247	The CookieOptionFlag supplied in the XML post is not valid.
-248	The CookieOptionFlag cannot be set if Html open rate is false.
-262	Failed to load the unsubscription address.
- 263	Invalid Alias name, failed to load the From Address.
- 265	Invalid TrackUnsubFromReplies value
- 268	Invalid Alias name, failed to load the ReplyToAddress.
-270	The current status of the mailing does not support this operation.
-271	The mailing type and the campaign type does not match. Possible reason: trying to create a approval mailing in a standard campaign.
-272	Test mailing does not support trigger messages.

TABLE: XML-related response codes

Code	Text
-273	The site specified does not have multi-language support.
-274	The language specified is not supported by DREAMmail server.
- 276	The Alias name is not specified in ReplyToAddress.
- 277	The status of the notification trigger mailing is Armed. Cannot be updated".
-372	Invalid value for element Format.
- 374	Invalid Arm Message value.
- 377	Failed to find unsubscription address at campaign or message level.
- 421	An invalid export scope was specified in the request.
-800	Expire date is missing or invalid
-810	The job name specified is not unique
- 1000	A general communications error occurred. contact Client Relations.
-1011	The CampaignName specified in the SuppressCampaign request does not exist or the Campaign is Inactive.
-1012	An invalid MailingName was specified in the Suppression request.
-1013	A Mailing with Invalid Status or with Invalid Type was specified in the Suppression request.
-1018	The value specified for the IF tag is invalid.
-1019	Operation specified for the Condition tag is invalid.
- 1024	Mailing can not be updated with suppression List.
- 1028	URL's that exceed xxx characters cannot be tracked.
-1033	Number of conditional contents in the request is more than maximum conditional contents in configuration file.
- 1034	Content name is not specified in xml request.
- 1035	Invalid content name specified in xml request.

Error messages generated by DREAMmail server

DREAMmail-generated error messages usually result from an incorrect configuration of the server, the server not being available, or an invalid configuration of your

database. The list of errors that can be generated by the *DREAMmail* server is shown in the table below.

TABLE: *DREAMmail* server-related response codes

Code	Text
-250	Internal error, contact Client Relations (failed to create content).
-251	Internal error, contact Client Relations (failed to set content name).
-252	Internal error, contact Client Relations (failed to commit content).
-255	Internal error, contact Client Relations (failed to load the content).
-256	Internal error, contact Client Relations (failed to load the default from address).
-257	Internal error, contact Client Relations (failed to load the subscription address).
-258	Internal error, contact Client Relations (failed to load the unsubscription address).
-259	Internal error, contact Client Relations (failed to create the campaign).
-260	Internal error, contact Client Relations (failed to load the reply to address).
-261	Internal error, contact Client Relations (failed to set send rate).
-300	Internal error, contact Client Relations (the list cannot be created).
-301	Internal error, contact Client Relations (the list cannot be loaded).
-302	Internal error, contact Client Relations (the batch importing object cannot be created).
-303	Internal error, contact Client Relations (the SetField method of batch importing object returns false).
-304	Internal error, contact Client Relations (the batch importing object cannot be committed).
-305	Internal error, contact Client Relations (server suppression rules load fail).
-306	Internal error, contact Client Relations (server suppression rules cannot be created/deleted/modified).
-307	Internal error, contact Client Relations (client suppression rules load fail).
-308	Internal error, contact Client Relations (client suppression rules cannot be created/deleted/modified).

TABLE: *DREAMmail* server-related response codes

Code	Text
-309	Internal error, contact Client Relations (the subscription object cannot be created).
-310	Internal error, contact Client Relations (the client batch unsubscription object cannot be committed).
-340	Internal error, contact Client Relations (failed to load Site Object).
-350	Internal error, contact Client Relations (failed to load Client Object).
-370	Internal error, contact Client Relations (failed to send notification message).
-371	Internal error, contact Client Relations (bad function parameter).
-380	Internal error, contact Client Relations (failed to load user).
-390	Internal error, contact Client Relations (failed to load exports directory).
-391	Internal error, contact Client Relations (failed to load export).
-392	Internal error, contact Client Relations (failed to create export).
-500	Internal error, contact Client Relations (invalid SQL statement).
-501	Internal error, contact Client Relations (database session error).
-600	Internal error, contact Client Relations (failed to generate response to your request).
-760	Internal error, contact Client Relations (failed to load Alias directory).
-761	Internal error, contact Client Relations (failed to save campaign information).
-780	Internal error. contact Client Relations. (Failed to initialize request struct)
-781	Internal error. contact Client Relations. (Failed to get response from Report Server)
-782	Internal error. contact Client Relations. (Failed to process request)
-783	Internal error. contact Client Relations. (Failed to allocate memory)
2205	An internal error that is returned if <i>DMConnect</i> cannot establish a connection to a <i>DREAMmail</i> server. This condition is likely to occur during system maintenance, when <i>DREAMmail</i> servers are brought down for maintenance and <i>DMConnect</i> CGI programs are still running.

Error messages for trigger messages

Error messages generated by trigger messages usually result from trying to send an unarmed, or a non trigger message. The list of errors that can be generated by trigger messages is shown in the table below.

TABLE: Trigger message-related response codes

Code	Text
-401	Trigger message is unarmed: arm message in <i>DREAMmail</i> interface.
-402	This is not a trigger message: triggering action failed.
-403	The number of event variables for all email addresses do not match.
-701	The Referral From Address specified is invalid.
-702	The TriggerFromAddr tag is empty.
-703	Cannot track unsubscribes from replies when using ReferralAddress. Ensure "Track unsubscribes from replies" option is not selected in DMUI.

Error messages for messages

Error messages generated by messages usually result from trying to send an action with the wrong mailing list specified in the request. The list of errors that can be generated by messages is shown in the table below.

TABLE: Mailing-related response codes

Code	Text
-410	UpdateMailing action failed: this specific message cannot be modified.
-420	StartMailing action cannot be processed: this message is already finished.
-450	Error retrieving Mailing Summary Information.

Error messages for RTM messages

The following response codes can be returned by *DMConnect* when you create and trigger *RTM* messages.

TABLE: RTM-related response codes

Code	Text
-279	URLs that exceed 255 characters cannot be tracked.
-280	You cannot edit a trigger message whose status is closed.
-281	You cannot arm/open a message whose status is closed.

TABLE: RTM-related response codes

Code	Text
-282	You cannot arm a message which does not conform with FTC status.
-283	You are trying to set an invalid status. Valid status are Open, Close or Arm.
-284	You cannot arm a message whose status is already Armed.
-285	You cannot use a RTM update functionality on a Non-RTM message.
-286	The current RTM mailing status is not valid for this operation.
-287	Invalid value for element AgreeToFTCSpec.
-288	Invalid value for element IgnoreSubUnsub.
-289	Invalid value for element ArmMessage.
-290	Invalid value for element HtmlOpenRateFlag.
-291	Invalid value for element AutosenseFlag.
-292	Invalid value for element RetryAutosenseFlag.
-293	Invalid value for element IncludeCampaignFooterFlag.
-294	The RTM message is created but it is failed to be armed.
-295	Failed to arm RTM mailing in UpdateRTMMailing.
-296	Failed to open RTM mailing in UpdateRTMMailing.
-297	Invalid value for element IgnoreInvalidEmail.
-298	Invalid value for element MessageType - must be from the defined list.
-404	DMConnect does not support triggering of the Real-Time Messages.

Error messages for reports

Error messages generated by the reports usually result from an incorrect configuration of the server, the server not being available, or an invalid configuration of your database. The list of errors that can be generated by reports is shown in the table below.

TABLE: Report-related response codes

Code	Text
-460	Export is still in progress. Please try again later.

TABLE: Report-related response codes

Code	Text
-710	Internal error, contact Client Relations (failed to load client from Report Server database).
-711	Internal error, contact Client Relations (failed to load site from Report Server database).
-712	Internal error, contact Client Relations (failed to load campaign from Report Server database).
-713	Internal error, contact Client Relations (failed to load list from Report Server database).
-714	Internal error, contact Client Relations (failed to load mailing from Report Server database).
-715	Internal error, contact Client Relations (failed to retrieve report from Report Server).
-716	Internal error, contact Client Relations (report server DB locked, bad SQL statement, or incorrect RS database configuration).
-750	Internal error, contact Client Relations (failed to connect to Report Server).
-751	Internal error, contact Client Relations. (Report Server Protocol mismatch. Please check DMConnect configuration.)
-752	Internal error, contact Client Relations. (Report Server version not supported. Please check the DMConnect configuration.)

DMConnect service error

The *DMConnect* service error occurs when a user attempts to use *DMConnect*, without having the necessary authorization.

TABLE: *DMConnect* service error message response codes

Code	Text
-900	Failed to register DMConnect service.

URL Update error messages

URL Update error messages occur when a request is submitted successfully but fails to update.

Code	Text
2501	The Update Tracked URL Request was successfully processed. The Following URL(s) have exceeded the allowable limit to update URLs (5 updates in 30 days). URL(s) cannot be updated. URL 1 :http://www.yahoo.com/bin22
704	The Mailing is in either Open, Blocked, Composing, or ToBeSent status.
725	This is a Trigger mail, It is either a pre 4.9.6 mail or it has to be sent atleast once to Update Tracked URLs or the message does not contain Tracked URLs.
729	An Invalid Mailing was specified.
736	The URL Update Limit is currently set to 0. This will not allow you to update URLs. Please contact support.
737	Update URLs functionality does not allow Updated URLs to include new Personalization tags in any part of the URL.

General error messages

General errors occur due to problems within *DREAMmail*.

TABLE: General error message response codes

Code	Text
-1000	A general communications error occurred. Contact Client Relations.
-1001	A general configuration error occurred. Contact Client Relations.
-1002	A log file creation error occurred, contact Client Relations.
-1003	An XML file creation error occurred. Contact Client Relations.
-1004	An invalid name identifier was specified in the request, the identifier should only contain --0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz-_.
-1005	Internal error. Contact Client Relations.
-1006	An unknown error occurred. Contact Client Relations.
-1007	Internal error. Contact Client Relations (unknown DLL error).
-1008	A general communications error occurred. Contact Client Relations.

TABLE: General error message response codes

Code	Text
-1009	A general database communications error occurred. Contact Client Relations.
-1010	Internal error. Contact Client Relations. (failed to load dll).
-2000	Internal error. Please contact Client Relations.

Validation errors

The following errors result from the validation of data that users enter into *DREAMmail*.

TABLE: Validation error response codes

Code	Meaning
1001	Attempting to create an object with an identifier that is the same as that of an existing object (INPUT_DUPLICATE).
1002	Cannot find the requested object, or operation requested on an object that does not exist (INPUT_NOTFOUND).
1003	Out-of-bounds error: an attribute, field, or a data item is out of bounds (INPUT_OOB).
1004	Wrong type — that is, a value supplied for an attribute, field, or a data item has incorrect or incompatible type (INPUT_WRONG_TYPE).
1005	Format of a data item supplied is incorrect. There is a subtle difference between it and INPUT_WRONG_TYPE, in that the format error is returned when a name contains a forbidden character, or a special character that is not escaped (INPUT_WRONG_FORMAT).
1006	A mandatory argument, attribute, or data is missing (INPUT_MISSING).
1007	Attribute or data item supplied is not recognized (INPUT_UNKNOWN).
1008	An operation or action is not supported because it is not yet ready or is obsolete (INPUT_NOT_SUPPORTED).

Security errors

The following response codes consist of security-related errors.

TABLE: Security error response codes

Code	Meaning
1101	Authentication error (login failed, etc.) (SEC_AUTHENTICATION).
1102	Password (or security credentials in general) expired (SEC_EXPIRED).
1103	Authorization error — that is the user does not have privileges to execute a certain operation, or to access an object, data, etc. (SEC_AUTHORIZATION).
1104	Wrong response for generate password (SEC_WRONGRESPONSE).
1105	User account is locked (SEC_ACCOUNTLOCKED).

Deployment errors

The following error messages result when message deployment fails. These syntax and run-time error codes are returned in the `/DMResponse/ResultData/GetMailingStatusRs/Code` element in response to `GetMailingStatus` request.

TABLE: Deployment error response codes

Code	Meaning
1201	Error occurs when user tries to send a message (or execute any other operation on the mailing object) while in a state in which <i>DREAMmail</i> does not allow deployment or any other action (MDEPLOY_WRONG_STATE).
Syntax	
1202	Syntax error with an expression; description contains more information of the expression (MDEPLOY_EXPR).
1203	Syntax error in mailing's <i>If</i> tag (MDEPLOY_IF).
1204	Syntax error in mailing's <i>SetVar</i> tag (MDEPLOY_SETVAR).
1205	Syntax error in mailing's <i>Switch</i> tag (MDEPLOY_SWITCH).
1206	Syntax error in mailing's <i>IncrementCounter</i> tag (MDEPLOY_INC_COUNTER).
1207	Syntax error in mailing's <i>DecrementCounter</i> tag (MDEPLOY_DEC_COUNTER).
1208	Syntax error in mailing's <i>RandomInteger</i> tag (MDEPLOY_RANDOM).
1209	Syntax error in mailing's <i>DeclareGlobalVar</i> tag (MDEPLOY_GLOBAL).

TABLE: Deployment error response codes

Code	Meaning
1210	Syntax error in mailing's <i>Skip</i> tag (MDEPLOY_SKIP).
1211	Syntax error in mailing's <i>Print</i> tag (MDEPLOY_PRINT).
Run-time errors	
1215	Message running <i>aLohaRecipient</i> tag (MDEPLOY_RECIPIENT_RUN).
1216	Error running <i>SetVar</i> or <i>aLohaSetVar</i> tag (MDEPLOY_SETVAR_RUN).
1217	Error running <i>If</i> or <i>aLohaIf</i> tag (MDEPLOY_IF_RUN).
1218	Error running <i>Switch</i> or <i>aLohaSwitch</i> tag (MDEPLOY_SWITCH_RUN).
1219	Error running <i>IncrementCounter</i> or <i>aLohaIncrementCounter</i> tag (MDEPLOY_INC_COUNTER_RUN).
1220	Error running <i>DecrementCounter</i> or <i>aLohaDecrementCounter</i> tag (MDEPLOY_DEC_COUNTER_RUN).
1221	Error running <i>DeclareGlobalVar</i> or <i>aLohaDeclareGlobalVar</i> tag (MDEPLOY_GLOBAL_RUN).
1222	error running <i>RandomInteger</i> or <i>alohaRandomInteger</i> tag (MDEPLOY_RANDOM_RUN).
1223	Recursive reference to content is found (MDEPLOY_RECURSIVE_REF).

Import errors

The following response codes result from errors that occur during the import process. These codes are returned in the `/DMResponse/Code` element as a response to the *TriggerFtpImport* request (that is, initiating a one-time import), or in the `/DMResponse/ResultData/GetFtpImportStatusRs/Code` element in response to *GetFtpImportStatus* request.

TABLE: Import error response codes

Code	Meaning
1300	File not found - No file of the indicated name is present in the FTP folder.
1301	Error trying to initiate an FTP import or a related operation for a site that is not enabled for FTP imports. (IMPORT_FTP_NOT_ENABLED).
1302	Source field(s) specified is not valid (that is, specified source field cannot be found in the source data heading). (IMPORT_SOURCE_FIELD).

TABLE: Import error response codes

Code	Meaning
1303	Destination (target) field(s) specified is not valid (either a field under the given name cannot be found or it has invalid type) (IMPORT_TARGET_FIELD).
1304	General error trying to decompress the file (IMPORT_ZIP).
1305	Compressed (zipped) file contains multiple files to be imported (IMPORT_ZIP_MULTIFILE).
1306	Zipped file is corrupt (IMPORT_ZIP_CORRUPT).
1307	File to be imported does not have valid content or format (IMPORT_INPUT_FILE).
1308	Maximum number of errors reached (that is, keep trying until the error count reaches its threshold.) (IMPORT_MAX_ERROR).
1309	Import job failed during the data hygiene phase.

Direct-deployment errors

The following response codes result from errors that occur during the direct-deployment process, which is part of the integration between DREAMmail and Epsilon's *Ensemble* campaign management product.

TABLE: Direct-deployment error response codes

Code	Meaning
-910	The new mailing name provided in the XML is not valid.
-911	Mailing type of source mailing does not support duplication.
-912	The date format is not valid.
-913	The campaign status is not valid.
-914	The mailing status is not valid.
-950	Internal error. contact Client Relations. (Failed to duplicate mailing.)
-951	Internal error. contact Client Relations. (Failed to get the content fields for direct deployment mailing.)
-952	Internal error. contact Client Relations. (Failed to load campaign summary.)
-953	Internal error. contact Client Relations. (Failed to load Direct deployment mailing summary.)

TABLE: Direct-deployment error response codes

Code	Meaning
-954	Internal error. contact Client Relations. (Failed to process request.)
-955	Internal error. contact Client Relations. (Failed to start direct deployment.)
-956	Internal error. contact Client Relations. (Failed to get FTP config.)
-957	Internal error. contact Client Relations. (Failed to load direct deployment status.)
-958	Internal error. contact Client Relations. (Failed to get processed files.)
-959	Internal error. contact Client Relations. (Failed to get direct deployment statistics.)
-960	Internal error. contact Client Relations. (Failed to prepare direct deployment mailing.)

GetJobStatus and GetExportReportStatus Responses

DMConnect allows you to check the status of jobs and reports that you are exporting. If you want to check on the status of a job that you have submitted, you use the `<GetJobStatus>` request. If you want to check on the status of a report for which you have submitted an `<ExportReport>` request, you use the `<GetExportReportStatus>` request.

This section discusses:

- [<GetJobStatus> request response codes](#)
- [<GetExportReportStatus> request response codes](#).

`<GetJobStatus>` request response codes

If your `<GetJobStatus>` request is successfully parsed, the first line of the response is the *DMConnect* response code, that tells you that your `<GetJobStatus>` request was successfully processed. The notification message provides a second code with the status of the job requested.

TABLE: `<GetJobStatus>` request response codes

Code	Text
-3	User error
-2	XML error
-1	Network error
0	Waiting
1	Processing
2	Completed

`<GetExportReportStatus>` request response codes

If your `<GetExportReportStatus>` request is successfully parsed, the first line of the response message is the *DMConnect* response message, that tells you that your `<GetExportReportStatus>` request was successfully processed. The notification message provides a second code with the status of the Exported Report that was requested.

TABLE: `<GetExportReportStatus>` request response codes

Code	Text
0	Scheduled

TABLE: <GetExportReportStatus> request response codes

Code	Text
1	Started
3	Finished
4	Export has been rescheduled. This is a normal state for scheduled jobs. You cannot use <i>DMConnect</i> to create scheduled jobs. However, you can use <i>DMConnect</i> to retrieve the status of scheduled jobs.
5	Export failed on the report server. Contact client relations.

USING SPOTLIGHT

Spotlight is a built-in feature of DREAMmail that helps you analyze the return on investment (ROI) from an email marketing effort. You can use *Spotlight* to measure ecommerce activities, including the amount of revenue that is generated by a website, or to track the number of subscribers who performed specific activities such as filling out a registration form or requesting a brochure.

This section discusses the following topics:

- [About Spotlight](#)
- [Description of Spotlight Tags](#)
- [Creating Spotlight Tags](#)
- [Generating Random Numbers in Spotlight Tags](#).

About *Spotlight*

This section contains an overview of *Spotlight* and explains what *Spotlight* is and how it functions in a website to help you track activity.

This section discusses the following topics:

- [What is *Spotlight*?](#)
- [About *Spotlight* implementation](#)
- [How *Spotlight* tracks activity](#)
- [How *Spotlight* eliminates duplicate transactions](#)
- [About secure websites](#)
- [Limitations of *Spotlight*](#)
- [Using *Spotlight* to track campaign performance](#).

What is *Spotlight*?

Spotlight is a post-click analysis tool that enables you to measure a subscriber's response to specific sections of a website by tracking that subscriber's activity. You can view information about this activity in the *Delivery and Response* report in *DREAMmail*. For more information about this report, see the *DREAMmail* help system.

About subscriber activity: An *activity* is any action, such as a purchase or a request for information, performed by a subscriber during his or her visit to a given section of a website. *Spotlight* is not a site analysis tool and does not track all activity on a website, such as how much time subscribers spend on a given web page. *Spotlight* tracks activity only on designated areas of your website.

About post-click activity: The type of activity that *Spotlight* tracks is called *post-click activity*. Post-click activity includes any action that a subscriber performs on your website after clicking a link in a message. This includes:

- activity for subscribers who access your website immediately after clicking a link in a message
- activity for subscribers who click a link in a message and then access your website some time in the future. There is no definite time limit, it depends on when the subscriber clicks the link in the message.

About *Spotlight* implementation

In order for *Spotlight* to track activities, you must do the following:

- insert a *Spotlight* tag into the sections, or pages, of your website where you want to track activity
- insert links to these pages in a message that you compose in *DREAMmail*

- enable the URL tracking feature in *DREAMmail* when you compose a message and track the links that correspond to the sections of your website where you insert *Spotlight* tags
- if your website resides on a secure server, contact your Epsilon representative about purchasing a digital certificate on your behalf (see [About secure websites](#)).

For more information, see [Creating Spotlight Tags](#).

How *Spotlight* tracks activity

Spotlight tracks activities by using an HTML tag called a *Spotlight* tag. You insert *Spotlight* tags into the pages of your website where you want to track activity. For example, you can insert a *Spotlight* tag on your checkout page that tracks the total amount of money that subscribers spend on your website. For more information, see [Description of Spotlight Tags](#).

When a subscriber opens a message or clicks a tracked link in a message (depending upon how you have configured *DREAMmail*), the subscriber receives a *cookie*. Cookies enable *Spotlight* to match a subscriber who has opened a message or clicked a link in a message to a person visiting your website. This cookie is used with the *Spotlight* tags on your web pages to identify the subscriber and track his or her activity on your website.

Spotlight tracks a subscriber's activity based on the most recent tracked URL on which a subscriber clicks in a message. For example, a subscriber receives a message from your *Winter_Sale* campaign, clicks a link in that message, and receives a cookie. Two months later, this same subscriber receives another message from your new campaign, *Summer_is_Here*, clicks a link in this message, and receives another cookie. When the subscriber visits your website, she is identified by the cookie that she received from the *Summer_is_Here* campaign, rather than the cookie she received from the *Winter_Sale* campaign, because the most recent URL that the subscriber clicked on was in a message from the *Summer_is_Here* campaign.

How *Spotlight* eliminates duplicate transactions

When processing *Spotlight* transactions, *DREAMmail* identifies and eliminates duplicate transactions through the following programmatic logic.

```
Criteria for duplicates is empty
If an Id is specified then
  Criteria for duplicates include: same Message name, Id, Category,
  Subcategory, TagType
If an AdId was specified then
  Criteria for duplicates includes: Same AdId
Else
  Criteria for duplicates includes: same AdId is Null
End If
```

```
EndIf
Examine the Spotlight Log for any transaction that matches all criteria.
If a match is not found then
    Log the transaction to the Database
Else
    Drop the duplicate transaction
End If
```

About secure websites

If your website resides on a secure server, you can specify that the *Spotlight* tags in your web pages request an image from one of Epsilon's secure servers. For more information about the image, see [Why do Spotlight tags use the tag?](#).

Additionally, Epsilon must purchase a secure digital certificate on your behalf and install this certificate on the server. This certificate prevents subscribers from receiving security warnings when they access your *Spotlight*-enabled website. To request a secure digital certificate, ask your Epsilon representative for a Digital Certificate form.

Limitations of Spotlight

Because *Spotlight* uses cookies to record data, there are some limitations in the ability to track information about subscriber activity:

- Information about a subscriber's activity on a website is recorded only if the subscriber clicks a tracked link in a message sent through *DREAMmail*. A subscriber does not have to visit a website the same day that he or she clicked a link in order to be tracked. For more information about creating messages and URL tracking, see the *DREAMmail* help system.
- If a subscriber has disabled cookies in his or her web browser and visits a *Spotlight*-enabled website, the subscriber's activity cannot be tracked through *Spotlight*.
- If a subscriber has opted out of receiving messages from a website by clicking the site's optout URL, the subscriber's activity cannot be tracked through *Spotlight*.
- *DREAMmail*'s implementation of *Platform for Privacy Preferences (P3P)* enables you to disable the setting of cookies when you create a web domain at the client level in *DREAMmail*. If the setting of cookies is disabled, subscribers' activity cannot be tracked through *Spotlight*.

For information about P3P and creating web domains, see the *DREAMmail* help system.

Using *Spotlight* to track campaign performance

You can use *Spotlight* to track the performance of an email campaign. The following example explains a campaign where a web publisher wants to track the number of subscribers who completed a form requesting a free compact disc. The web publisher sends a message promoting this offer and includes in the message the URL `http://www.music-fun.com` which is the link to the web page that contains the form.

The following diagram shows how *Spotlight* tracks a subscriber's post-click activities

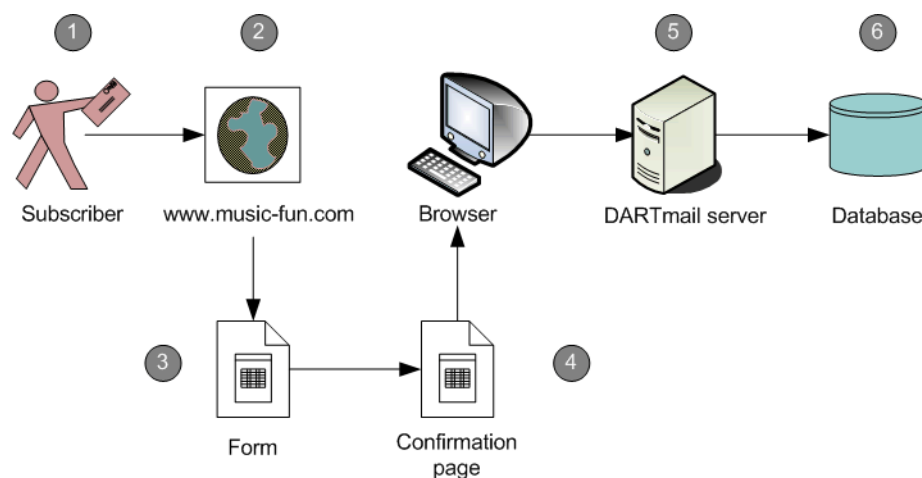


FIGURE: How *Spotlight* works

Stage	Description
1	A subscriber receives a message from an email campaign for a free CD.
2	The subscriber clicks a link in the message and is redirected to <code>http://www.music-fun.com/form.cgi</code> , the web page that contains the form for the free compact disc.
3	The subscriber completes and submits the form.
4	The confirmation web page containing the <i>Spotlight</i> tag is displayed to the subscriber.
5	The browser requests the information defined in the <i>Spotlight</i> tag.
6	The database records the request as an action taken by the subscriber in this message.

Description of *Spotlight* Tags

Spotlight tags enable you to specify the type of activity that you want to track on your website. *Spotlight* tags work in conjunction with cookies to track a particular subscriber's activities during a visit to your website. For more information about cookies, see [How Spotlight tracks activity](#).

This section discusses the following topics:

- [About Spotlight tags](#)
- [Why do Spotlight tags use the tag?](#)
- [Why are key-values used in Spotlight tags?](#)
- [Types of Spotlight tags](#).

About *Spotlight* tags

A *Spotlight* tag is an HTML tag that is used to track information about a subscriber's activity on a website. *Spotlight* tags are comprised of the following:

- the HTML tag `` and the attribute `src` (for example, ``)
- a series of key-values
- a random number that prevents caching by a subscriber's web browser. For more information, see [Generating Random Numbers in Spotlight Tags](#).

Why do *Spotlight* tags use the `` tag?

The `` HTML tag is used in *Spotlight* tags so that when a subscriber accesses a web page, an image request is made to the Epsilon servers. This image request is what enables the servers to identify a subscriber based on the cookie that the subscriber received when he or she opened the message or clicked a link in a message (depending upon how you have configured *DREAMmail* to send cookies).

The value that you specify for the `` tag is the location of a GIF file on the Epsilon servers. This GIF file is a 1x1 pixel transparent image. If the 1x1 pixel GIF cannot load for any reason, the rest of the web page displays correctly without waiting for the tag to load. Since this *Spotlight* image is small and transparent, this image cannot be seen and does not affect loading times, even on slow modem connections.

Why are key-values used in *Spotlight* tags?

A **key-value** is an attribute in a *Spotlight* tag that transfers information about a subscriber's activity, such as the type of activity that the subscriber performed, to the Epsilon servers. Key-values are comprised of two parts: the key and its value, separated by an equals (=) sign. There are no spaces between the key and the equals

sign or between the equals sign and the value. The format of a key-value is *key=value*. For example, in the key-value `type=1`, `type` is the **key**, and `1` is the **value**.

Note: To include unsafe characters in a key-value, you must encode them using escaped ASCII format. For example, to include a space, replace the space with `%20` as in `category=This%20and%20That`. Accented or multi-byte characters are not supported. For more information, see [Appendix - ASCII Escape Codes](#).

The value in a key-value can either be hard-coded or dynamic.

Hard-coded values: You can use hard-coded values to transfer information that is not dependent on the input of a subscriber. For example, you can use the *Type* key-value to specify the type of *Spotlight* tag that you want to create. In order to create a specific type of tag, you must hard code the value that corresponds to the type that you want to create (for more information, see [Types of Spotlight tags](#)).

Dynamic values: You can use dynamically generated values to transfer information that is dependent on an external action or a back-end event, such as the activity of a subscriber or a script that you write to generate a unique session ID. For example, you can use the *Amount* key-value to transfer the number of items purchased during a visit to your website (for more information, see [Description of key-values](#)).

Types of Spotlight tags

There are two types of *Spotlight* tags: counter tags and sales tags.

You use the *Type* key-value to specify whether a tag is a counter tag or a sales tag. The type of tag that you use on your website depends on what activity you want to track. For more information about deciding what activities to track, see [Deciding what activity to track](#).

Counter tags: Counter tags track the number of times subscribers visit a given web page. To create counter tags, you use the *Type* key-value and specify a value of *1*. For more information, see [Description of key-values](#).

Sales tags: Sales tags track the number of purchases made on a given web page. A **transaction** is a purchase made by a subscriber, which can include one or more items. You can then use this information to determine unique purchasers, total items purchased, and so on.

To create sales tags, you use the *Type* key-value and specify a value of *0*. For more information, see [Description of key-values](#).

When you create sales tags to track purchases, you must decide how you want to count a multiple-item purchase. A multiple item purchase can be counted as either:

- a single transaction, where each item is counted in the same category or subcategory
- multiple transactions, where a separate transaction is recorded for each unique category or subcategory combination. To track multiple transactions, you must insert multiple *Spotlight* tags on your web page—one tag for each unique combination of category and subcategory. For more information, contact your Epsilon representative.

Creating *Spotlight* Tags

You create *Spotlight* tags to track subscriber activity on different pages in your website. Before you create *Spotlight* tags, decide in which web pages you want to place the tags and what subscriber activity that you want to track. Identifying the pages of your website where you want to place the tags can help you decide what activity to track. Based on these decisions, you use different combinations of key-values to create a tag that tracks activities.

This section discusses the following topics:

- [Deciding on the placement of *Spotlight* tags](#)
- [Deciding what activity to track](#)
- [Syntax of *Spotlight* tags](#)
- [Description of key-values](#)
- [Required key-values for sales tags and counter tags](#)
- [Procedure for creating *Spotlight* tags](#)
- [Testing *Spotlight* tags](#)
- [Additional troubleshooting methods.](#)

Deciding on the placement of *Spotlight* tags

Epsilon recommends that you place *Spotlight* tags immediately after the `<body>` HTML tag. This position enables a request to be sent to the Epsilon servers even if a subscriber clicks *Stop* in his or her browser, closes the browser, or interrupts the loading of the web page.

Most Epsilon customers insert *Spotlight* tags at the top of confirmation pages. Confirmation pages are displayed when a subscriber submits information using a web form; for example, when a subscriber places an order or submits a survey. Placing *Spotlight* tags on confirmation pages enables the values in key-values to be determined by the action of the subscriber and to be inserted dynamically. For example, if you want to track the number of items that a subscriber purchased during a visit to your website this dynamic number can be passed to the tag via the confirmation page. For more information, see [Why are key-values used in *Spotlight* tags?](#)

Deciding what activity to track

After you have determined in which pages you want to place *Spotlight* tags, identify what activities you want to track. Deciding on the purpose of your *Spotlight* tags helps you to create more effective tags.

You can create different types of *Spotlight* tags depending on what activity you want to track. You use different combinations of key-values to track different types of activity.

You can track two types of activity: ecommerce or non-ecommerce activity.

Tracking ecommerce activity: You use a sales tag to track ecommerce activities, such as a subscriber purchasing an item on a website (see [Sales tags](#) for more information). When you track ecommerce activity, you can obtain the following information:

- subscribers who purchase items on a website
- the number of items purchased and the amount of the transactions.

You can view this ecommerce information in the *Delivery and Response* report in *DREAMmail*. For information about reports in *DREAMmail*, see the *DREAMmail* help system.

You can view report information for a single tracked URL in a message, all of the tracked URLs in a message, or all of the messages in a campaign. This information can help you to analyze the success of a message or campaign, and the most active customers in your mailing lists.

Tracking non-ecommerce activity: You use a counter tag to track non-ecommerce activities. When you track non-ecommerce activity, you can obtain the following information:

- subscribers who click to view a form
- subscribers who enter information into online forms, such as surveys or registration forms.

You can view non-ecommerce information in the *Delivery and Response* and *Most Active Customer* reports in *DREAMmail*.

When you track non-ecommerce activity, you must write a script that generates a random number to populate the value of the *ID* key-value. This random number prevents a subscriber from being tracked twice. For more information, see [Generating Random Numbers in Spotlight Tags](#).

Syntax of Spotlight tags

The following example displays the syntax of a *Spotlight* tag and all of the possible key-values that you can use as attributes. When inserting key-values into *Spotlight* tags, each key-value must be preceded by an ampersand separator(&).

```
<IMG SRC="http[s]://subdomain.domain.com/cgi-binx/  
buyrate?Amount=[amount_of_transaction]&ID=[ID]&type=[type_of_activity]&  
category=[category_name]&subcategory=[subcategory_name]&quantity=[quant
```

```
ity]&AdID=[adID]" width="1" height="1">
```

Note: To access an image on a Epsilon secure server, change the protocol in the URL of the *Spotlight* tag from `http` to `https`.

Description of key-values

The following table describes the key-values that you can use to create *Spotlight* tags. For information about which key-value to use in each type of tag, see [Required key-values for sales tags and counter tags](#).

TABLE: Description of key-values

Keys	Description
Amount	<p><i>Amount</i> specifies the monetary amount of a purchase. The value for <i>Amount</i> must be in numeric format, with a maximum length of 18 characters (including two decimal places).</p> <p>The value for <i>Amount</i> is usually populated dynamically; for example, when a subscriber confirms the total amount of his or her purchase, that amount is inserted into the <i>Spotlight</i> tag.</p> <p>You must use this attribute when creating sales tags, but cannot use it when creating counter tags. This key-value does not collect data when used in a counter tag. For more information, see Types of Spotlight tags.</p> <p>Important: Do not pass a dollar sign (\$) or other currency symbol with the value in the <i>Amount</i> key-value. If you do pass a symbol with the value, DREAMmail does not record the value that you pass.</p>
ID	<p><i>ID</i> specifies a unique order or ID number. The value for <i>ID</i> can be in variable character format (a combination of letters, numbers, and special characters), with a maximum length of 128 characters. This value must be a unique number and can be populated by an external system or by writing a script that generates a random number. For more information, see Generating Random Numbers in Spotlight Tags.</p> <p>For example, if you use this key-value in a sales tag, the value can be populated by your inventory system. If you use this key-value with a counter tag, the value can be populated by generating a random number to identify a unique subscriber who fills out a survey.</p> <p>Important: Epsilon strongly recommends that you use this key-value in your tags. The unique number specified in this key-value prevents caching and therefore prevents the same activity from being tracked twice.</p>
type	<p>Use the <i>Type</i> key-value to specify the type of <i>Spotlight</i> tag that you want to create:</p> <ul style="list-style-type: none"> To create a counter tag, specify a value of <code>1</code>. To create a sales tag, specify a value of <code>0</code>. <p>The default for this key-value is a sales tag. If you do not specify the <i>Type</i> key-value, the default tag is set. For more information, see Types of Spotlight tags.</p>

TABLE: Description of key-values

Keys	Description
category	<p><i>Category</i> specifies the category to which the action defined in the <i>Spotlight</i> tag is associated. The value for <i>Category</i> must be in text format, with a maximum length of 255 characters. You can include spaces by escaping them with a + or %20; for example, <code>category=This%20and%20That</code>.</p> <p>A category can be a group created by you or your company to identify different categories of inventory in DREAMmail reports. For example the <i>Category</i> key-value in a sales tag can specify that a subscriber purchased an item from the <i>Shoes</i> category.</p> <p>The value for the <i>Category</i> key-value can be populated dynamically by an external inventory system or hard-coded by you on a page of your website that contains only one type of item.</p> <p>You can use the <i>Category</i> key-value for both sales and counter tags. If you do not specify a <i>Category</i> key-value, the transaction is displayed in DREAMmail reports as <i>Unclassified</i>.</p>
subcategory	<p><i>Subcategory</i> specifies a subcategory within the category to which the action defined in the <i>Spotlight</i> tag is associated. The value for <i>Subcategory</i> must be in text format, with a maximum length of 255 characters. You can include spaces by escaping them with a + or %20; for example, <code>subcategory=This%20and%20That</code>.</p> <p>A subcategory can be a group created by you or your company to identify in DREAMmail reports different subcategories of inventory within larger categories. For example, the <i>Subcategory</i> key-value in a sales tag can specify that a subscriber purchased an item from the <i>Loafer</i> subcategory, which is part of the <i>Shoe</i> category.</p> <p>The value for the <i>Subcategory</i> key-value can be populated dynamically by an external inventory system or hard-coded by you on a page of your website that contains only one type of item.</p> <p>You can use the <i>Subcategory</i> key-value for both sales and counter tags. If you do not specify a <i>Subcategory</i> key-value, the transaction is displayed in DREAMmail reports as <i>Unclassified</i>.</p> <p>Important: Subcategory names must be unique within a category but can be repeated in other categories</p>
quantity	<p><i>Quantity</i> specifies the number of items purchased in a sales transaction or the shipping weight of an item purchased. The value for <i>Quantity</i> must be in numeric format, with a maximum length of 18 characters (including two decimal places).</p> <p>The value for <i>Quantity</i> is usually populated dynamically; for example, when a subscriber confirms the total number of items in his or her purchase, that number is passed to the <i>Spotlight</i> tag.</p> <p>You can use this key-value only when creating sales tags. This key-value does not collect data when used in a counter tag. For more information, see Types of Spotlight tags.</p>

TABLE: Description of key-values

Keys	Description
AdID	<p><i>Ad ID</i> is an extra key-value that you can use to specify any additional information about the subscriber action that you track. For example, you can use <i>Ad ID</i> to specify a promotion code.</p> <p>The value for <i>AdID</i> can be in variable character format (a combination of letters, numbers, and special characters), with a maximum length of 128 characters.</p>

Required key-values for sales tags and counter tags

To create sales tags and counter tags, you must use certain key-values. Some of the key-values that you use are required and others are optional. The following table displays the key-values that you use to create each type of tag and indicates which tags are required and which tags are optional.

TABLE: Required key-values for sales tags and counter tags

Tag	Key-values used to create tag	Required or optional key-value
Sales	Type=0	Optional. Since the default for the <i>Type</i> key-value is a sales tag, you do not have to specify a <i>Type</i> when creating a sales tag. For more information, see Types of Spotlight tags .
	Amount=[amount]	Required
	ID=[ID]	Optional
	AdID=[AdID]	Optional
	Quantity=[number_of_items]	Optional
	Category=[category]	Optional
	Subcategory=[subcategory]	Optional

TABLE: Required key-values for sales tags and counter tags

Tag	Key-values used to create tag	Required or optional key-value
Counter	Type=1	Required. Since the default for the <i>Type</i> key-value is a sales tag, you must specify this key-value with a value of 1 to create a counter tag. For more information, see Types of Spotlight tags .
	ID=[ID]	Optional. You have to write a script that generates random number in a counter tag. For more information, see Generating Random Numbers in Spotlight Tags .
	AdID=[AdID]	Optional
	Category=[category]	Optional
	Subcategory=[subcategory]	Optional
	Amount=[amount]	You cannot use these key-values in counter tags.
	Quantity=[number_of_items]	

Procedure for creating *Spotlight* tags

The following procedure explains how to create *Spotlight* tags and add them to a website. To create *Spotlight* tags:

Step	Procedure
1	Identify the locations on your website where you want to insert <i>Spotlight</i> tags. Typically, you place <i>Spotlight</i> tags on the confirmation pages that are displayed when a subscriber submits a form, such as a survey or an online order.
2	Identify the subscriber activity that you want to track. Use this information to determine the following: <ul style="list-style-type: none"> the number of sales tags that you want to create the number of counter tags that you want to create the key-values that you need to use in each tag in order to track the correct activity.

Step	Procedure
3	<p>Begin building the <i>Spotlight</i> tag using the following basic tag.</p> <pre></pre> <p>The tag records the number of actions performed by each subscriber.</p>
4	<p>Add the <code>type</code> key-value to specify the type of tag that you want to create (a sales tag or a counter tag).</p> <p>This step is optional. If you do not specify this key-value, the type defaults to a sales tag. Specify a value of <code>0</code> to create a sales tag or a value of <code>1</code> to create a counter tag.</p> <pre></pre> <p>The tag now records the number of actions performed by each subscriber and whether the action is a purchase (for sales tags) or some other event, such as filling out a survey (for counter tags).</p>
5	<p>Add the <code>ID</code> key-value to track the unique order number or ID number of each subscriber action.</p> <p>Although this step is optional, Epsilon strongly recommends that you use this key-value in your <i>Spotlight</i> tags. This prevents the same activity from being tracked twice. Do not leave any spaces at the start of the key-values.</p> <pre></pre> <p>The tag now records the number of actions performed by each subscriber, whether the action is a purchase or some other event, and the unique ID number associated with this action, such as an ID number from an inventory system.</p>
6	<p>Add the <code>category</code> key-value to track the category associated with a subscriber action.</p> <p>This step is optional. If you do not include this key-value, subscriber action is displayed in reports as <i>unclassified</i>.</p> <pre></pre> <p>The tag now records the number of actions performed by each subscriber, whether the action is a purchase or some other event, the unique ID number associated with this action, and the category to which the subscriber action is associated; for example in a sales tag, that a subscriber purchased an item from the <i>Shoe</i> category.</p>

Step	Procedure
7	<p>Add the <code>subcategory</code> key-value to track the subcategory that is associated with a subscriber action.</p> <p>This step is optional. If you do not include this key-value, subscriber action is displayed in reports as <i>unclassified</i>.</p> <pre></pre> <p>The tag now records the number of actions performed by each subscriber, whether the action is a purchase or some other event, the unique ID number associated with this action, the category to which the subscriber action is associated, and the subcategory to which the subscriber action is associated; for example, that a subscriber purchased an item from the <i>Loafer</i> subcategory within the <i>Shoe</i> category.</p>
8	<p>If you are creating a sales tag, you must add the <code>amount</code> key-value to track the monetary amount of each subscriber purchase (required for sales tags).</p> <p>You must use this key-value when creating sales tags. You cannot use this key-value with counter tags.</p> <pre></pre> <p>The tag now records the number of actions performed by each subscriber, whether the action is a purchase or some other event, the unique ID number associated with this action, the category to which the subscriber action is associated, the subcategory to which the subscriber action is associated, and the value of each purchase.</p>
9	<p>If you are creating a sales tag, add the <code>quantity</code> key-value to track the number of items that a subscriber purchased.</p> <p>You can also use this key-value to track the shipping weight of an item. This key-value supports a decimal value up to 0.999 for this purpose. You cannot use this key-value in counter tags.</p> <pre></pre> <p>The tag now records the number of actions performed by each subscriber, whether the action is a purchase or some other event, the unique ID number associated with this action, the category to which the subscriber action is associated, the subcategory to which the subscriber action is associated, and the value of each purchase and the number of items purchased.</p>

Step	Procedure
10	<p>Add the <code>AdID</code> key-value to track additional information about a subscriber activity.</p> <p>This step is optional. You can use this key-value for any information that you want to <i>DREAMmail</i> to track; for example, a promotion code.</p> <pre></pre> <p>The tag now records the number of actions performed by each subscriber, whether the action is a purchase or some other event, the unique ID number associated with this action, the category to which the subscriber action is associated, the subcategory to which the subscriber action is associated, for sales tags, the value of each purchase and the number of items purchased, and additional information that you want to track.</p>
11	<p>Insert the <i>Spotlight</i> tag in the web page on which you want to track the subscriber action that is defined in this tag.</p> <p>For more information, see Deciding on the placement of Spotlight tags.</p>

Testing Spotlight tags

Before you send a message that contains a link to a *Spotlight*-enabled web page, test your *Spotlight* tags. To test *Spotlight* tags:

Step	Action
1	In <i>DREAMmail</i>, create a message and include in this message a link to at least one of the web pages that you have tagged.
2	Apply the <i>URL Tracking</i> setting to this link.
3	Send this message to yourself or to some other email addresses that you use for testing purposes.
4	<p>Open the test message and click the tracked link.</p> <p>When you click the tracked link, a cookie is sent to your computer. (Alternatively, you may have configured <i>DREAMmail</i> to send the cookie when you open the message.)</p>
5	<p>From the same computer, visit the website referenced in the tracked URL and access the web page that contains the <i>Spotlight</i> tag.</p> <p>To access the tagged web page, you may need to perform the activity that you are trying to track. For example, if you have a sales tag on the confirmation page of your website, you must complete the shopping process and submit an order so that the confirmation page, which contains the <i>Spotlight</i> tag, is displayed.</p>
6	Check the <i>DREAMmail Active Tracking</i> report to see if your activity was recorded.

Additional troubleshooting methods

Once you have added the tag to the web page, and followed the testing procedure in [Testing Spotlight tags](#), you can visually ensure that the tag is functioning correctly. You can check the following:

- Ensure that the 1x1 pixel GIF is included in the list of calls for the web page. You can perform this check using the *Page Info* feature of Netscape Navigator. For more information about how to perform this check, contact your Epsilon representative.
- Ensure that the correct type of tag has been implemented on the web page. Sales and counter tags perform different functions and track activities differently. You must ensure that, for example, a counter tag has not been added to a web page where you expect to measure sales activity.
- Ensure that the correct key-values are being used in the *Spotlight* tag. For example, if the *Spotlight* tag is located in a web page served from a secure server (https), but the key-value in the *Spotlight* tag specifies a non-secure server (http), data on a subscribers activity may not be captured, and some browsers display a security warning.

Generating Random Numbers in *Spotlight* Tags

To prevent browser caching, when you create a *Spotlight* tag, you **must** write a script that populates the *ID* key-value with a unique number. Unique numbers are required in all *Spotlight* tags to prevent browser caching, which can cause *Spotlight* to count the same subscriber twice.

For sales tags, this unique number can be passed to the sales tag from an external source, such as an inventory or shopping cart system. Most shopping cart systems generate a unique order number at the time of a purchase. This ensures that the same purchase is not tracked twice if a subscriber refreshes the web page. For counter tags, however, you must write a script that generates a unique number for you.

You create unique numbers by writing a script that generates random numbers. Random numbers must be alphanumeric, long enough, and random enough so the same number has almost no chance of appearing twice in a day. You can write a script that generates random numbers using JavaScript, ASP, JSP, PHP, or Cold Fusion.

Note: Do not implement these sample tags. Contact your Epsilon representative to create real tags.

This section provides five examples:

- [JavaScript](#)
- [ASP](#)
- [JSP](#)
- [PHP](#)
- [Cold Fusion \(CF\)](#).

JavaScript

This is sample JavaScript code used to generate random numbers in *Spotlight*. Use this code only for testing.

```
<!-- Start of Epsilon Tracking Code: Please do not remove THIS IS NOT A  
REAL TAG DO NOT USE -->  
<SCRIPT language="JavaScript"> var axel = Math.random()+""; var a = axel  
* 10000000000000; document.write('<IMG SRC="http://redir.Epsilon.net/  
cgi-bin1/buyrate?type=0&category=mensshoes&subcategory=wingtips&  
quantity=100.00&adid=Ad58&id='&a&' " WIDTH=1 HEIGHT=1 BORDER=0>');  
</SCRIPT>  
<!-- End of Epsilon Tracking Code: Please do not remove THIS IS NOT A  
REAL TAG DO NOT USE -->
```

ASP This is sample ASP code used to generate random numbers in *Spotlight*. Use this code only for testing.

```
<!-- Start of Advertiser Tracking Code: Please do not remove THIS IS NOT
A REAL TAG DO NOT USE -->
<% Randomize() %>
<IMG SRC="http://redir.Epsilon.net/cgi-bin1/
buyrate?type=1&category=mensshoes&subcategory=wingtips&quantity=100.00&
adid=Ad58&id=<% Response.Write(int(rnd()*3141581723+10)) %>" WIDTH=1
HEIGHT=1 BORDER=0>
<!-- End of Epsilon Tracking Code: Please do not remove THIS IS NOT A
REAL TAG DO NOT USE -->
```

JSP This is sample JSP code used to generate random numbers in *Spotlight*. Use this code only for testing.

```
<!-- Start of Advertiser Tracking Code: Please do not remove THIS IS NOT
A REAL TAG DO NOT USE -->
<% long random_number = (long)(Math.random() * 1000000000); %>
<IMG SRC="http://redir.Epsilon.net/cgi-bin1/buyrate?type=1
&category=mensshoes&subcategory=wingtips&quantity=100.00
&adid=Ad58&id=<%= random_number %>" WIDTH=1 HEIGHT=1 BORDER=0>
<!-- End of Epsilon Tracking Code: Please do not remove THIS IS NOT A
REAL TAG DO NOT USE -->
```

PHP This is sample PHP code used to generate random numbers in *Spotlight*. Use this code only for testing.

```
<!-- Start of Epsilon Tracking Code: Please do not remove THIS IS NOT A
REAL TAG DO NOT USE -->
<? $t= time()> <IMG SRC="http://redir.Epsilon.net/cgi-bin1/
buyrate?type=1&category=mensshoes&subcategory=wingtips&quantity=100.00
&adid=Ad58&id=<? echo $t>?" WIDTH=1 HEIGHT=1 BORDER=0>
<!-- End of Epsilon Tracking Code: Please do not remove THIS IS NOT A
REAL TAG DO NOT USE -->
```

Cold Fusion (CF) This is sample Cold Fusion code used to generate random numbers in *Spotlight*. Use this code only for testing.

```
<!-- Start of Epsilon Tracking Code: Please do not remove THIS IS NOT A
REAL TAG DO NOT USE -->
<CFSET randnum = RandRange(1000000, 9999999)> <CFOUTPUT> <IMG
SRC="http://redir.Epsilon.net/cgi-bin1/buyrate?type=1
&category=mensshoes&subcategory=wingtips&quantity=100.00
&adid=Ad58&id=#randnum#" WIDTH=1 HEIGHT=1 BORDER=0>
</CFOUTPUT>
<!-- End of Epsilon Tracking Code: Please do not remove THIS IS NOT A
REAL TAG DO NOT USE -->
```

APPENDIX - ASCII ESCAPE CODES

When you include unsafe, accented, non-English, double-byte, and multi-byte characters in the content of a *DREAMmail* message, you must escape special characters and those characters that have an ASCII value of greater than 127, so that they are not misinterpreted in HTML formats or by the XML parser. The following table shows the character and ASCII code for special and accented characters above the 127 basic US-ASCII characters, as well as the corresponding code that can be used in XML. Use the codes in this table to replace these characters in your content.

TABLE: Escaped ASCII codes for extended-ASCII characters

Symbol	ASCII Number	Escape Code	Symbol	ASCII Number	Escape Code
--	128	%80	--	146	%92
--	129	%81	--	147	%93
--	130	%82	--	148	%94
--	131	%83	--	149	%95
--	132	%84	--	150	%96
--	133	%85	--	151	%97
--	134	%86	--	152	%98
--	135	%87	--	153	%99
--	136	%88	--	154	%9a
--	137	%89	--	155	%9b
--	138	%8a	--	156	%9c
--	139	%8b	--	157	%9d
--	140	%8c	--	158	%9e
--	141	%8d	--	159	%9f
--	142	%8e		160	%a0
--	143	%8f	ı	161	%a1
--	144	%90	ç	162	%a2
--	145	%91	£	163	%a3
¤	164	%a4	¼	190	%be

TABLE: Escaped ASCII codes for extended-ASCII characters

Symbol	ASCII Number	Escape Code	Symbol	ASCII Number	Escape Code
¥	165	%a5	ı	191	%bf
ı	166	%a6	À	192	%c0
§	167	%a7	Á	193	%c1
¨	168	%a8	Â	194	%c2
©	169	%a9	Ä	196	%c4
ª	170	%aa	Å	195	%c3
«	171	%ab	Ä	197	%c5
¬	172	%ac	Æ	198	%c6
	173	%ad	Ç	199	%c7
®	174	%ae	Ð	208	%d0
—	175	%af	È	200	%c8
º	176	%b0	É	201	%c9
±	177	%b1	Ê	202	%ca
²	178	%b2	Ë	203	%cb
³	179	%b3	Ì	204	%cc
´	180	%b4	Í	205	%cd
µ	181	%b5	Î	206	%ce
¶	182	%b6	Ï	207	%cf
·	183	%b7	Ñ	209	%d1
¸	184	%b8	Ò	210	%d2
¹	185	%b9	Ó	211	%d3
º	186	%ba	Ô	212	%d4
»	187	%bb	Õ	213	%d5
¼	188	%bc	Ö	214	%d6
½	189	%bd	×	215	%d7
Ø	216	%d8	ó	243	%f3

TABLE: Escaped ASCII codes for extended-ASCII characters

Symbol	ASCII Number	Escape Code	Symbol	ASCII Number	Escape Code
Ù	217	%d9	ô	244	%f4
Ú	218	%da	õ	245	%f5
Û	219	%db	ö	246	%f6
Ü	220	%dc	÷	247	%f7
Ý	221	%dd	ø	248	%f8
Þ	222	%de	ù	249	%f9
ß	223	%df	ú	250	%fa
à	224	%e0	û	251	%fb
á	225	%e1	ü	252	%fc
â	226	%e2	ý	253	%fd
ã	227	%e3	þ	254	%fe
ä	228	%e4	ÿ	255	%ff
å	229	%e5			
æ	230	%e6			
ç	231	%e7			
è	232	%e8			
é	233	%e9			
ê	234	%ea			
ë	235	%eb			
ì	236	%ec			
í	237	%ed			
î	238	%ec			
ï	239	%ef			
ñ	241	%f1			
ò	242	%f2			

INDEX

A

accented characters in message content 4, 279
 activities
 definition of 260
 tracking in *Spotlight* 268
 AddProfileFields request 28
 AOL format, escaping special characters in 6, 279
 ASCII escaped format 40, 52, 53, 160, 169, 170
 ASCII, extended 4, 279
 Asian languages in message content 4

C

campaigns
 creating in *DMConnect* 34
 CDATA sections, including message content in 5
 ChangeRTMMailingStatus request 31
 characters, special 4
 ClearList request 32
 client relations, contacting 239
 cookies 261
 counter tags, required key-values 271
 CreateCampaign request 34
 CreateConditionalContent request 37
 CreateDuplicateMailing request 46
 CreateList request 48
 CreateMailing request 50
 CreateMailingFromTemplate request 61
 CreateRTMMailing request 67
 customer support, see *client relations*

D

direct-deployment errors 255
 disabling cookies 262
DMConnect
 accepted parameters 26
 description of vii
 languages supported 7
 maximum file size 11
 notification response messages 237
 parameters 26, 27
 responses 22, 236
 set up process 14
 types of requests 19
DMConnect requests
 AddProfileFields 28
 ChangeRTMMailingStatus 31

ClearList 32
 CreateCampaign 34
 CreateConditionalContent 37
 CreateDuplicateMailing 46
 CreateList 48
 CreateMailing 50
 CreateMailingFromTemplate 61
 CreateRTMMailing 67
 descriptions of 23
 ExportReport 68
 GetAllExportNames 70
 GetExportedReport 72
 GetExportReportStatus 74
 GetFtpImportStatus 76
 GetHygieneRuleNames 80
 GetJobStatus 82
 GetListNames 84
 GetMailingContent 89
 GetMailingStatus 92
 GetMessageSize 98
 GetObjectNames 84, 102
 GetProfileFields 109
 GetTemplateVariables 115
 GetTestMailingStatus 118
 ImportList 127
 ManageLists 130
 SendMailingForApproval 132
 StartMailing 134
 TriggerMailingToEmailAddresses 143
 TriggerMailingToList 149
 Unsubscribe 154
 UpdateConditionalContent 157
 UpdateContent 163
 UpdateMailing 167
 UpdateProfile 179
 UpdateRTMMailing 182
DMConnect responses
 descriptions of 23
 GetFtpImportStatusRs 78
 GetHygieneRuleNamesRs 81
 GetListNamesRs 86
 GetMailingContentRs 90
 GetMailingStatusRs 94
 GetMessageSizeRs 100
 GetObjectNamesRs 105
 GetProfileFieldsRs 111
 GetTemplateVariablesRs 117
 GetTestMailingStatusRs 120
 double-byte characters 4
 dynamic personalization
 using *DMConnect* 163

E

- EndDate parameter 27
- Ensemble*
 - error codes 255
- error codes, see *notification response messages*
- errors
 - DMConnect* deployment 253
 - DMConnect* importing 254
 - DMConnect* security 253
 - DMConnect* service 250
 - DMConnect* validation 252
 - DREAMmail* server 245
 - general 251
 - HTTP post 242
 - import 254
 - mailing, see *messages*
 - messages 248, 252, 253
 - Real-Time Messaging 248
 - report 249
 - response notification messages 239
 - trigger message 248
 - user-response 240
 - XML file 242
- escaping characters 4, 279
 - in HTML and AOL format messages 6
 - in XML requests 4
 - see also *CDATA sections*
- European languages in message content 4
- exporting data using *DMConnect* 68
- ExportList report 202
- ExportReport* request types
 - about 68
 - Bouncebacks* 190
 - ClickThroughs* 193
 - DeliveryResponse* 187
 - EmailAddressChanges* 200
 - ExportList* 202
 - ListActivity* 196
 - ProfileChanges* 205
 - RTMClicks* 211
 - RTMDelivered* 208
 - RTMHTMLOpens* 213
 - RTMNotDelivered* 209
 - RTMSpotlight* 214, 216
 - Unsubscription* 216
- extended ASCII characters, escaping 4, 279

F

- FromDate parameter 27
- FTP
 - get status of import 76

G

- general communication errors 240
- GetAllExportNames request 70
- GetExportedReport request 72
- GetExportReportStatus request 74
- GetFtpImportStatus request 76
- GetFtpImportStatusRs response 78
- GetHygieneRuleNames request 80, 81
- GetHygieneRuleNamesRs response 81
- GetJobStatus request 82
- GetListNames request 84
- GetListNamesRs response 86
- GetMailingContent request 89
- GetMailingContentRs response 90
- GetMailingStatus request 92
- GetMailingStatusRs response 94
- GetMessageSize request 98
- GetMessageSizeRs response 100
- GetObjectNames request 102
- GetObjectNamesRs response 105
- GetProfileFields request 109
- GetProfileFieldsRs response 111
- GetTemplateVariables request 115
- GetTemplateVariablesRs response 117
- GetTestMailingStatus request 118
- GetTestMailingStatusRs response 120
- guidelines for XML documents 3

H

- help desk, see *client relations*
- HTML format
 - escaping characters in message content 6
 - escaping special characters in 279
 - reserved characters 6
- HTTP post, *DMConnect*
 - C++ 227
 - HTML 225
 - PERL 226
 - URLEncoding form variables 222
 - user-originated errors 240

I

- ImportList request 127
- integrating *DREAMmail* with
 - SiteCatalyst* 15
- internal errors in *DREAMmail* 240

K

- key-values
 - counter tag requirements 271

descriptions of 269
sales tag requirements 271

L

language support
in message content 7

M

mailing list requests, *DMConnect*
 CreateList 48
 ImportList 127
 Unsubscribe 154
 UpdateProfile 179
mailing lists
 creating in *DMConnect* 48
 importing subscriber data using *DMConnect* 127
 unsubscribing addresses in *DMConnect* 154
 updating subscriber data in *DMConnect* 179
ManageLists request 130
message content
 accented characters 4, 279
 Asian languages in 4
 escaping extended ASCII characters 4, 279
 European languages in 4
 languages supported 7
 placement of content in *DMConnect* requests 12
 types of characters in 4
 updating using *DMConnect* 163
message requests, *DMConnect*
 ChangeRTMMailingStatus 31
 CreateMailing 50
 CreateMailingFromtemplate 61
 CreateRTMMailing 67
 UpdateMailing 167
 UpdateRTMMailing 182
messages
 creating in *DMConnect* 50
 sending in *DMConnect* 132, 134
 updating in *DMConnect* 167
multi-byte characters 4
multi-part mail 55, 172

N

non-English characters 4
notification response messages
 239
 deployment errors 253
 direct-deployment 255
 DMConnect service errors 250
 DREAMmail server errors 245

Ensemble 255
errors in HTTP post 242
format of code 239
general errors 251
general notification 237
importing errors 254
mailing errors 248
Real-Time Messaging 248
report errors 249
response messages and codes 239
security errors 253
specific notification 237
success messages 241
trigger message errors 248
types 241
user-originated errors 240
validation errors 252
XML file errors 242
notification response security errors 253

P

P3P 262
parameters
 names of in *DMConnect* 26
Platform for Privacy Preferences, see *P3P*
post-click activities 260

R

Real-Time Messaging
 errors 248
 updating mailing status in *DMConnect* 31
report requests, *DMConnect*
 ExportReport 68
 GetAllExportNames 70
 GetExportedReport 72
 GetExportReportStatus 74
 GetJobStatus 82
reports
 Spotlight 260
 status of export job 74, 82
 Bouncebacks 190
 ClickThroughs 193
 DeliveryResponse 187
 EmailAddressChanges 200
 ExportList 202
 ListActivity 196
 ProfileChanges 205, 208
 RTMClicks 211
 RTMHTMLOpens 213
 RTMNotDelivered 209
 RTMSpotlight 214

Unsubscription 216
request syntax, *DMConnect* 16
reserved characters
 in HTML 6
 in XML 5
response codes, *DMConnect* 239
response syntax, *DMConnect* 16
ROI 1, 259
RTM, see *Real-Time Messaging*

S

safe special characters 4
sales tags, required key-values 271
secure websites, *Spotlight* tags in 276
security errors 253
send multi-part mail 55, 172
sending mailing requests, *DMConnect*
 SendMailingForApproval 132
 StartMailing 134
 TriggerMailingToEmailAddresses 143
 TriggerMailingToList 149
sending messages in *DMConnect* 132, 134
SendMailingForApproval request 132
SiteCatalyst, integration with *DREAMmail* 15
special characters
 escaping 279
 reserved 4
 safe and unsafe 4
 unsafe 4
Spotlight
 counter 265
 limitations of 262
 sale 265
 types of tags 265
StartMailing request 134
system and configuration errors
 general communication 240
 internal 240

T

tags, *Spotlight* 268
technical support, see *client relations*
templates, creating messages in *DMConnect* 61
transactions, definition of 265
triggered messages, sending in *DMConnect* 143, 149

U

unsafe special characters 4
UpdateConditionalContent request 157
UpdateContent request 163

UpdateMailing request 167
UpdateRTMMailing request 182
URLEncoding form variables, HTTP post 222
URLs
 entering in message body 12
 tracking 261
user-originated errors, *DMConnect*
 HTTP post 240
 XML 240

V

validation errors 252
validation tool, XML 11
version of *DREAMmail* in request 221

X

XML
 16
 CDATA sections 5
 elements 3
 escaping special characters in 4, 279
 guidelines 3
 reserved characters 5
 syntax of requests and responses 16
 tags in 3
 tips for development 11
 user-originated errors 240
 validation tool 11