Lyris ListManager 9.0 API Guide

Lyris Technologies, Inc.

Introduction to the Application Programming Interface (API)

ListManager allows administrators to access and manipulate the database tables using SOAP (Simple Object Access Protocol). SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages. SOAP is a key element of Microsoft's .NET architecture for future Internet application development. SOAP provides a higher-level interface to database information so that no knowledge of database query protocols is required.

ListManager's API can be invoked from any language with SOAP support, including:

.NET
Perl
Java
Php
Tcl
Ruby

Examples of ListManager's functions in different languages are available in your ListManager's soap directory:

http://yourlistmanagerurl/soap

Or, you may navigate to Utilities: Other: Programming API.

To use the ListManager API effectively, it is important to understand the information stored in each table column. Many functions require modification of record fields in order to work properly. It is also important to understand the possible side effects of modifying these fields. For example, if you set a member’s 'MemberType' attribute to 'unsub', then that member’s record may be deleted if the list is set up to purge unsubscribed members.

Migrating from LMAPI 1.4.2 to 1.5

General / All Clients

1. If performing a version check, it is now **1.5 Beta** instead of **1.4.2 Release**.

2. The default WSDL location is now http://yourlistmanagerurl:82/?wsdl where "servername" is the location of the ListManager back end, and 82 is the default port. **Note**: The default port can be changed.

3. The function UpdateMemberDemographics has changed:

- Previous argument list: SimpleMemberStruct, DemographicsArray
- New argument list: SimpleMemberStruct, KeyValueType

•**Note**: DemographicsArray and KeyValueType are equivalent, so code should convert by simply altering the type for any variables of this type.

.NET 1.1

No changes required other than general notes above.

.NET 2.0

Any integer or boolean member of a struct class must have the accompanying **Specified** value set to **true** in order for it to be transmitted to the server. Please refer to example client code.

Java

You may need to change a line like:

public static LmapiSoap_PortType lm;

to:

public static LmapiSoap lm;

PHP

No changes required other than general notes above.

Perl

Perl is not yet supported with API v1.5. It may be possible to make it work, but code written to the 1.4.2 API will not work without alteration.

TCL

TCL is not yet supported with API v1.5. It may be possible to make it work, but code written to the 1.4.2 API will not work without alteration.

Disabling the API

By default, the SOAP API is enabled.

To disable the 1.4.2 API:
1. Navigate to your ListManager ~\tclweb\bin directory and open tclhttpd.rc with a plain-text editor such as vi or Notepad.
2. Locate the Config disable_api parameter, which is set to 0 by default. Change the value to 1.

Example:

```
#########################
# Enable/disable API
```

```
Config disable_api 1
```

To disable the 1.5 API:
Start ListManager with the keyword **nolmapi**:

lm start nolmapi

Resources

The following links may be helpful to you in programming for the ListManager API.

Lyris Resources

Lyris ListManager 9.0 Reference Manual
Lyris ListManager 9.0 Programmer's Guide

SOAP Resources

Overview of SOAP and Web Services (Microsoft)
Microsoft Web Services Developer Center
Java Web Services - Axis
Perl SOAP::Lite Home
Pocket SOAP
W3C SOAP v 1.2 Specification
SOAP Tutorial (W3Schools)

Programming Quick Start

To use ListManager's API, follow these steps:

1. Download and Install the SOAP Client for Your Language

2. Point SOAP Toolkit to WSDL URL

To use the API, you must point your SOAP toolkit at your ListManager installation's WSDL URL, located at:

http://yourlistmanagerurl:82/?wsdl

3. Call the API Version

Always call ApiVersion before running any other functions to ensure your client and the ListManager server you are attempting to communicate with are using the same version.

4. Set the Login

The security for the ListManager API is the same as that for the ListManager web interface; the same user name/password must be set in the script.

More Information

The following pages give more detailed and specific instructions for various languages:

PHP
Perl
Java
C#.NET

Examples of ListManager's functions are available in your ListManager's soap directory:

http://yourlistmanagerurl/soap

PHP SOAP Quickstart Guide

Examples of how to use ListManager's functions with PHP are available in your ListManager's soap directory:

http://yourlistmanagerurl/soap

Your ListManager installation includes a toolkit for Php, Nausea. To use it and run a simple test:

1. Unzip and install the Nausea client on your webserver to a directory (e.g., to /var/www/html/lmapiTest). NuSoap may be found at

http://yourlistmanagerserver/soap/php/nusoap-0.6.7.zip

2. Create a new php file called *test.php* in the lmapiTest directory.

Your directory structure should look like this in your webserver:

```
lmapiTest/
     |
     |_ test.php
     |_ lib/
     |_ samples/
```

3. To run your script, point your browser to the location:

http://[yourwebserver]/lmapiTest/lmapiTest.phphttp://[YOURWEBSERVER]/lmapiTest/lmapiTest.php

If you are using php as a standalone, run

```
php test.php
```

The following is the minimal client script for PHP:

```
-----PHP------
<?
$userName = 'admin';
$password = 'lyris';

// create client
$lmapiClient = new soapclient( $wsdl_location, true );

//set basic authentication
$lmapiClient->setCredentials($userName,$password, 'basic');

//make sure there was no error.
$err= $lmapiClient->getError();
if ($err) {
echo "<h2>Error</h2><pre> $err <hr> $lmapiClient-
>debug_str;\n\n";
return false;
}

$lmapi = $lmapiClient->getProxy();

//set basic authentication
$lmapi->setCredentials($userName,$password, 'basic');

if ( $lmapi->ApiVersion() != "1.5" )
{
echo "<h3>Incorrect API version! Expected: ".$lmapi-
>ApiVersion()."</h3>";
return false;
}
?>
```

Perl SOAP Quickstart Guide

Perl is not yet supported with API v1.5. It may be possible to make it work, but code written to the 1.4.2 API will not work without alteration.

Examples of how to use ListManager's functions with Perl are available in your ListManager's soap directory:

http://yourlistmanagerurl/soap

These instructions show how to use a Perl toolkit, SOAP::Lite, with ListManager.

1. Download SOAP::Lite:
http://www.soaplite.com/

2. Install SOAP::Lite by running:

```
perl -MCPAN -e 'install SOAP::Lite'
```

3. Define a SOAP::Lite function to handle basic authentication. See the SOAP::Lite documentation for more information.

The following is the minimal client script for Perl:

```
----PERL----
use SOAP::Lite  ;

my $USER = 'admin';
my $PASS = 'lyris';
my $WSDL = 'http://yourlistmanagerurl:82/?wsdl';
my $SOAP = SOAP::Lite -> service($WSDL);

my $ApiVersion  = $SOAP->ApiVersion();
if ($ApiVersion ne '1.5') {

    die "Incompatible API version '$ApiVersion' detected!\n"

}

#define Authentication

sub SOAP::Transport::HTTP::Client::get_basic_credentials {
    return $USER => $PASS;

}
```

Java SOAP Quickstart Guide

Examples of how to use ListManager's functions with Java are available in your ListManager's soap directory:

http://yourlistmanagerurl/soap

These instructions show how to use a Java toolkit, Ant, with ListManager.

1. Install Ant:
http://ant.apache.org/

2. Make sure ant is in your path.

3. Install the following software from Sun:

   SOAP with Attachments API for Java (SAAJ)
   Java Architecture for XML Binding (JAXB)
   Java API for XML-based RPC (JAX-RPC)
   Java API for XML Processing (JAXP)
   Java API for XML Registries (JAXR)

4. Install Apache Axis.

5. Create a directory (e.g., c:\temp\lmapiTests).

6. Copy the file *build.xml* from your ListManager server to your lmapiTests directory. This file is available at:

http://*yourlistmanagerserver*/soap/java/build.xml)

Modify the URL above so *yourlistmanagerserver* points to the URL for your ListManager installation.

7. Modify build.xml to reflect where you installed the Java libraries in step 3.

8. In your directory (c:\temp\lmapiTests) run:

```
java org.apache.axis.wsdl.WSDL2Java
http://yourlistmanagerserver/soap/wsdl/ -plmapi
```

Modify the URL above so *yourlistmanagerserver* points to the URL for your ListManager installation.

9. wsdl2Java creates a directory named *lmapi*. In lmapi/, create a file named *Client.java*.

10. build.xml is already set up to use Client.java as the program entry point.

11) In your directory (c:\temp\lmapiTest) type "ant" to run the program.

The following is the minimal client script for Java:

```
----JAVA-----
package lmapi;
import java.util.*;
import org.apache.axis.client.Stub;

public class Client {

     // static means class global variables, it does not change
between instances.
     public static String userName ="admin";
     public static String password = "lyris";
     public static String apiVersion = "1.5";


     public static void main (String [] args) throws Exception
     {
     // Make a service
     service = new LmapiLocator();

           try
           {
                 // Get a stub that implements the service
operations
                 lm = service.getlmapiSoap();

                 // Lyris Api uses authorization
                 ((Stub)lm).setUsername(userName);
                 ((Stub)lm).setPassword(password);

                 // make sure we are using the right version
                 if ( apiVersion.compareTo(lm.apiVersion()) == 0
)
                 {

                       // run code here

                 } else {
                       System.out.println("Bad API version!!
Expected '" + apiVersion + "', but got: '" + lm.apiVersion() +
"'");
                 }
           } catch (Exception ex) {
                 ex.printStackTrace();
           }
     }
}
----END JAVA-----
```

C#.NET SOAP Quickstart Guide

Examples of how to use ListManager's functions with C# .NET are available in your ListManager's soap directory:

http://yourlistmanagerurl/soap

These instructions show how to use Visual Studio with ListManager.

1. Open Visual Studio, and then open the example project.

2. Modify the Web reference http://yourserver/soap/wsdl/http://yourlistmanagerurl:82/?wsdl so *yourlistmanagerurl* points to the URL for your ListManager installation.

Note to .NET 2.0 users:

Any integer or boolean member of a struct class must have the accompanying Specified value set to "true" in order for it to be transmitted to the server. Please refer to example client code.

Function Reference

Member Functions

SelectMembers
SelectSimpleMember
CreateSingleMember
UpdateMemberDemographics
UpdateMemberPassword
UpdateMemberKind
UpdateMemberEmail
UpdateMemberStatus
DeleteMembers
CopyMember
SendMemberDoc
Unsubscribe
CheckMemberPassword
GetMemberID
GetListnameFromMemberID
GetEmailFromMemberID
EmailOnWhatLists
EmailPasswordOnWhatLists
CreateMemberBan

List Functions

CreateListAdmin
UpdateListAdmin
DeleteList
CreateList
UpdateList
SelectList

GetListID

Content Functions
SelectContent

Mailing Functions
SendMailing
SendMailingDirect
ImportContent
MailingStatus
ScheduleMailing

SQL and Misc. Functions
SqlSelect
SqlDelete
SqlInsert
SqlUpdate
ApiVersion
CurrentUserEmailAddress

Segment Functions
SelectSegments

Member Functions

SelectMembers

Description
Selects an array of MemberStructs using FilterCriteriaArray. Example:

```
ar[0] = (ListName = list1); ar[1] = (domain = example.com)
```

Arguments
FilterCriteriaArray - an array of member field/value pairs (values from MemberStruct)

Returns
An array of MemberStructs. The maximum number of rows returned is 750; to return all rows, use SelectSimpleMember.

SelectSimpleMembers

Description
Selects an array of MemberStructs using FilterCriteriaArray. Example:

```
ar[0] = (ListName = list1); ar[1] = (domain = example.com)
```

Arguments

FilterCriteriaArray - an array of member field/value pairs (values from MemberStruct)

Returns

An array of SimpleMemberStructs.

CreateSingleMember

Description

Creates a single member of a list. CreateSingleMember is easier to use programmatically than CreateManyMembers, as it requires three strings instead of the MemberStructArray that CreateManyMembers requires.

Arguments

EmailAddress - The email address of the member (string)

FullName - The full name of the member (string)

ListName - The name of the list the member will be added to (string)

Returns

Success Flag (boolean)

UpdateMemberDemographics

Update demographics for a member. Use field names as defined in your members_ table paired with the correct values in the demographics array.

Arguments

SimpleMemberStruct - A SimpleMemberStruct

KeyValueType - An array of KeyValueType(s) as defined by your database.

Note: In the previous version of the ListManager API, the second argument was **DemographicsArray**. DemographicsArray and KeyValueType are equivalent, so code should convert by altering the type for any variables of this type.

Returns

Success Flag (boolean)

UpdateMemberPassword

Updates the password of a member.

Arguments

SimpleMemberStruct - A SimpleMemberStruct

NewPassword - The new member password (string)

Returns

Success Flag (boolean)

UpdateMemberKind

Updates the "kind" of a member. The member "kind" defines what kind of mail the member receives (e.g., mail or digest; see MemberKindEnum for member kind options). The member kind correlates to the member_.SubType_ field.

Arguments

SimpleMemberStruct - A SimpleMemberStruct

NewMemberKind - The member kind, from MemberKindEnum

Returns

Success Flag (boolean)

UpdateMemberEmail

Updates the email address of a member.

Arguments

SimpleMemberStruct - A SimpleMemberStruct

NewEmailAddress - The new email address of the member (string)

Returns

Success Flag (boolean)

UpdateMemberStatus

Updates the "status" of a member. The member "status" defines the membership status (e.g., normal, unsub, confirm; see MemberStatusEnum for member status options). The member status correlates to the member_.MemberType_ field.

Arguments

SimpleMemberStruct - A SimpleMemberStruct

NewStatus - The member's new status, from MemberStatusEnum

Returns

Success Flag (boolean)

DeleteMembers

Completely removes members based on FilterCriteraArray. Returns number of users deleted.

Arguments

FilterCriteriaArray - an array of member field/value pairs (values from MemberStruct)

Returns

The number of members deleted (int)

CopyMember

Copies a member.

Arguments

SimpleMemberStruct - A SimpleMemberStruct

NewEmailAddress - The email address of the new member (string)

NewFullName - The full name of the new member (string)

NewListName - The name of the list this member should be copied into (string)

Returns

Success Flag (boolean)

SendMemberDoc

Sends a "confirm", "hello", "goodbye", "held", "private" or "delivery" document to any email address or member. Returns Outmail ID.

Arguments

SimpleMemberStruct - A SimpleMemberStruct

DocTypeIn - The message to be sent, from MessageTypeEnum

Returns

The MessageID of the outmail created (int)

Unsubscribe

Unsubscribes members based on FilterCriteraArray. Members who are unsubscribed remain in the database but their MemberType_ = 'unsub'. Returns number of users unsubscribed.

Arguments

SimpleMemberStructArray - An array of SimpleMemberStructs

Returns

The number of members unsubscribed (int)

CheckMemberPassword

Returns true if the user in the SimpleMemberStruct exists with password "Password"

Arguments

SimpleMemberStruct - A SimpleMemberStruct

Password - The password to be checked for this member (string)

Returns

True or false (boolean)

GetMemberID

Returns the MemberID for a given EmailAddress and ListName as defined in a SimpleMemberStruct.

Arguments

SimpleMemberStruct - A SimpleMemberStruct

Returns

The member's ID (int)

GetListnameFromMemberID

Returns the list name for a particular member id.

Arguments

MemberID - The ID of the member (int)

Returns

The list the member is on (string)

GetEmailFromMemberID

Returns the email address for a particular member id.

Arguments

MemberID - The ID of the member (int)

Returns

The email address of the member (string)

EmailOnWhatLists

Returns an array of list names that the specified EmailAddress is subscribed to.

Arguments

EmailAddress - The email address (string)

Returns

Array of lists the email address is subscribed to (ArrayOfString)

EmailPasswordOnWhatLists

Returns an array of list names that the specified EmailAddress and Password is subscribed to.

Arguments

EmailAddress - The email address (string)

Password - The password (string)

Returns

Array of lists the email address/password combination is subscribed to (ArrayOfString)

CreateMemberBan

Creates a member ban. Returns BanID. BanLogic is: A = "accepted" C = "conditionally accepted" R = "banned". Most member bans will be R (banned); other types may prevent others from joining the list.

Arguments

MemberBan - A member to be banned, from MemberBanStruct

Returns

The BanID (int)

UpdateListAdmin

Updates the list admin settings of any member.

Arguments

SimpleMemberStruct - A SimpleMemberStruct

IsListAdmin - Whether this member is a list admin (boolean)

RecieveListAdminMail - Whether the administrator should receive email messages for list admins (boolean)

RecieveModerationNotification - Whether the administrator should receive notification of messages pending moderation (boolean)

BypassListModeration - Whether the administrator should be able to post messages directly to the list, bypassing list moderation (boolean)

Returns

Success Flag (boolean)

CreateListAdmin

Creates a new list administrator. Returns MemberID of successfully created list admin. Note that this function will not be successful if the EmailAddress specified already exists on the list.

Arguments

EmailAddress - The email address of the administrator (string)

Password - The password of the administrator (string)

ListName - The name of the list (string)

FullName - The full name of the administrator (string)

RecieveListAdminMail - Whether the administrator should receive email messages for list admins (boolean)

RecieveModerationNotification - Whether the administrator should receive notification of messages pending moderation (boolean)

BypassListModeration - Whether the administrator should be able to post messages directly to the list, bypassing list moderation (boolean)

Returns

MemberID of successfully created list administrator (int)

List Functions

DeleteList

Deletes a list, and everything associated with it (members, content, etc.).

Arguments

ListName - The name of the list to be deleted (string)

Returns

Success Flag (boolean)

CreateList

Creates a new list and returns the newly created listID.

Arguments

ListType - The type of list to be created, from ListTypeEnum

ListName - The name of the list (string)

ShortDescription - A short description of the list (string)

AdminName - The name of the list administrator (string)

AdminEmail - The email address of the list administrator (string)

AdminPassword - The password of the list administrator (string)

Returns

The ListID (int)

UpdateList

Updates the settings of a list. Returns boolean success.

Arguments

ListStruct - The list to be updated, from ListStruct

Returns

Success Flag (boolean)

SelectList

Select an array of ListStructs.

Arguments

ListName - The name of the list. If blank or null, all lists are selected (string)

SiteName - The name of the site. If entered, selects all lists on that site (string)

Returns

An array of ListStructs

GetListID

Gets the ListID from the ListName for a list.

Arguments

ListName - The name of the list (string)

Returns

The ListID (int)

Content Functions

SelectContent

Select an array of content structs based on FilterCriteriaArray.

Example:

```
ar[0] = (ListName = list1); ar[1] = (Title like %newsletter%);
```

Arguments

FilterCriteriaArray - an array of content field/value pairs (values from ContentStruct)

Returns

An array of ContentStruct

Mailing Functions

SendMailing

Sends a mailing based on MailingStruct by creating a row in the inmail_ table. The message is then subject to list processing and validation as if it were a mailing created by an administrator through the administrator GUI. If SegmentID is blank or 0, the mailing is sent to the entire list as defined in MailngStruct. If successful, returns the InMailID of new mailing.

Arguments

SegmentID - The ID of the segment to be sent to. Sends to the entire list if blank or null (int)

MailingStruct - The contents of the mailing as a MailingStruct

Returns

The InmailID of the inmail created if successful (int)

SendMailingDirect

Sends a mailing based on MailingStruct to any email address or member IDs, or an array of email addresses and member IDs. Creates mailing directly in the OutMail_ table, so does not perform any list processing. The email address(es) defined in the EmailAddressArray do not need to be subscribed to any list. Returns true if message was successfully queued.

Arguments

EmailAddressArray - An array of email addresses to be sent the message (ArrayOfString)

MemberIDArray - An array of member IDs to be sent the message (ArrayOfInt)

MailingStruct - The contents of the mailing, as defined by a MailingStruct

Returns

Outmail ID

ImportContent

Takes a Content ID and returns a SimpleMailingStruct. The

SimpleMailingStruct can be used to create a new mailing by copying all of its elements to a new MailingStruct and sending that.

Arguments

ContentID - The ID of the content to be made into a mailing (int)

Returns

A SimpleMailingStruct, which may be copied into a new MailingStruct.

MailingStatus

Get the status of a mailing sent with list processing (e.g., sent with SendMailing). MailingStatus will not work with a mailing created with SendMailingDirect.

Arguments

InMailID - The ID of the inmail to be checked (int)

Returns

The inmail transaction log (string)

ScheduleMailing

Schedules a mailing for a future date based on MailingStruct. The message is then subject to list processing and validation as if it were a mailing created by an administrator through the administrator interface. If SegmentID is blank or 0, the mailing is sent to the entire list as defined in MailingStruct. If successful, returns the InMail ID of the new mailing.

Arguments

SegmentID - The ID of the segment to be sent to. Sends to the entire list if blank or null (int)

SendDate - The date and time the mailing should be sent /soap/object_reference.tml(datetime)

MailingStructIn - A MailingStruct

Returns

InMailID - The ID of the inmail to be created (int)

SQL and Misc. Functions

SQLSelect

Runs an SQL SELECT Statement, and returns an array of array of strings. The first element of the array contains field names. Only server administrators may access this function.

Arguments

SqlStatement - The SQL SELECT statement to run (string)

Returns

ArrayOfArrayOfString

SqlDelete

Deletes row(s) from a table using a "where" clause. Only server administrators may access this function.

Arguments

Table - The table the rows should be deleted from. See the Programmer's Guide in the Reference Manual for more information about ListManager's tables (string)

SqlWhere - The "where" clause (string)

Returns

Success Flag (boolean)

SqlInsert

Inserts rows into a table. DataArray should contain field names and values. If returnid is true, the id of the row is returned. otherwise functions returns 1. Only server administrators may access this function.

Arguments

Table - The table the rows should be inserted. See the Programmer's Guide in the Reference Manual for more information about ListManager's tables (string)

DataArray - ArrayOfKeyValueType

ReturnID - Whether the ID of the new row should be returned (Boolean)

Returns

If ReturnID is true, then the ID of the new row is returned; if not, the number 1 is returned if successful, or an exception if failed.

SqlUpdate

Updates rows into a Table using an SQL "where" clause. DataArray should contain field names and values. Only server administrators may access this function.

Arguments

Table - The table whose the rows should be updated. See the Programmer's Guide in the Reference Manual for more information about ListManager's tables (string)

DataArray - The data array to update the table (ArrayOfKeyValueType)

SqlWhere - The SQL "where" clause that should be run (string)

Returns

Success Flag (boolean)

ApiVersion

Verifies the version of the API the target ListManager server is running. Different api versions have incompatible clients, so this function ensures the version you expect is the version hosted on the listmanager server.

Arguments

(none)

Returns

The API version (string)

CurrentUserEmailAddress

This function returns the email address associated with the user logged in. For example, this command will return the email address associated with the server administrator "admin".

Arguments

(none)

Returns

The email address (string)

Segment Functions

SelectSegments

Selects an array of SegmentStructs using FilterCriteriaArray.

Example:

```
ar[0] = (SegmentName = aol); ar[1] = (ListName like %lis%);
```

Arguments

FilterCriteriaArray - an array of member field/value pairs (values from SegmentStruct)

Returns

An array of SegmentStructs.

Report Functions

TrackingSummary

Exports a report of delivery success and failures, unique and total opens, and unique clicks per URL.

Arguments

OutmailID - The MessageID_ of an outmail.

Returns

TrackingSummaryStruct

Type Reference

FilterCriteriaArray

Classes and Structs

ContentStruct

CharSetStruct

KeyValueType

- ListStruct
- MailingStruct
- MemberStruct
- MemberBanStruct
- SegmentStruct
- SimpleMailingStruct
- SimpleMemberStruct

Enumerations

- BanLogicEnum
- ConfirmUnsubEnum
- DeliveryReportEnum
- DocTypeEnum
- EnableScriptingEnum
- ListTypeEnum
- LoggingLevelEnum
- MailFormatEnum
- MailSectionEncodingEnum
- MemberKindEnum
- MemberListSecurityEnum
- MemberStatusEnum
- MessageTypeEnum
- ModeratedEnum
- NameRequiredEnum
- NewSubscriberPolicyEnum
- PostPasswordEnum
- RecencyOperatorEnum
- ScriptingLevelEnum
- SegmentTypeEnum
- SubscriptionReportEnum

FilterCriteriaArray

An array of strings used to find data that match the field-value pairs in the array.

Example:

```
ar[0] = (ListName=list1); ar[1] = (domain like %.com)
```

The relationship between the field-value pairs may be expressed as:

= equals

<> does not equal

< less than

> greater than

<= less than or equal

>= greater than or equal

like

A % is used for wildcards.

The data returned matches all of the elements in the array. In SQL programming terms, it is as if the elements of the array were joined with AND. For example, in the example above, members who belonged to list1 AND whose domain contains .com would be selected.

Classes and Structs

ContentStruct

Used to create content.

Name	Type	Description
ContentID	int	Unique ID in Docs_ table
DateCreated	date	Date content was created
Description	string	The description of this content
DocType	DocTypeEnum	The document type
HeaderFrom	string	The From: header
HeaderTo	string	The To: header
IsReadOnly	boolean	If true, web interface will not let user change this document in the web interface
IsTemplate	boolean	If true, this content is a template
NativeTitle	string	Title of the document in its native language
Title	string	Title of the document, in English

Note: Since Microsoft .NET does not support NULL date/times, the date/time "2001-01-01 00:00:00" is considered null.

CharSetStruct

The CharSetStruct defines the character set used by a mailing. The CharSetStruct is used with the MailingStruct.

Name	Type	Description
CharSetID	int	An auto-incrementing ID field that uniquely identifies a record in this table
Description	string	A text description. Informational only
Name	string	The MIME compatible name of the character set

KeyValueType

Used by the UpdateMemberDemographics function.

Name	Type	Description
Value	string	Value to insert
Name	string	Name of the demographic field being updated

ListStruct

The ListStruct describes all of the parameters for a particular list.

Name	Type	Description
ListID	int	The unique identifier for this list
Additional	string	Additional information about this list (not used by ListManager)
AddListNameToSubject	boolean	If true, adds the list's name

		to the "Subject:" header
Admin	string	Administrator responsible for this list
AllowCrossPosting	boolean	If true, this list should accept cross-posted messages (identical messages sent to multiple lists)
AllowDuplicatePosts	boolean	If true, this list should accept duplicate messages
AllowInfo	boolean	If true, this list should allow non-members to read information about it
AnyoneCanPost	boolean	If true, this list should allow anyone to post; if false, it requires those who post to be members
ApproveNum	int	The number of approvals required of new members before they may post to the list without moderation
ArchiveDays	int	The number of days archived messages are kept
ArchiveNum	int	The number of archived messages kept
AutoReleaseHour	int	The time of day messages should be automatically approved and released
BlankSubjectOk	boolean	If true, the list should permit messages with blank "Subject:" headers
Child	boolean	If true, this list should accept postings from other lists
CleanAuto	boolean	If true, this list should automatically put members who bounce too much mail on hold
CleanNotif	int	The length of time (days) a "held" member will be notified
Comment	string	Additional information about this list (not used by ListManager)
CommentsID	string	Comment about this list

		(links to a document)
ConfirmSubscribes	boolean	Whether this list should confirm new subscriptions
ConfirmUnsubscribes	ConfirmUnsubEnum	Whether this list should confirm unsubscribes
DateCreated	dateTime	The date and time this list was created
DefaultFrom	string	The default "From:" header
DefaultSubject	string	The default "Subject:" header
DefaultTo	string	The default "To:" header
DeliveryReports	int	What delivery reports should be sent to administrators
DescLongDocID	string	Long description of this list (links to a document)
DetectHtmlByDefault	boolean	If true, this list should detect HTML capability by default
DetectOpenByDefault	boolean	If true, this list should detect HTML opens by default
DigestFooter	string	The digest footer
DigestHeader	string	The digest header
Disabled	boolean	If true, this list is disabled
DontRewriteMessageIDHeader	boolean	If true, ListManager will not rewrite the Message-ID, but instead leave it alone, and use X-Lyris-Message-ID
EnableScripting	EnableScriptingEnum	What level of scripting the list allows
ErrHold	int	How many bounces before a member is put on hold
From	string	If set, specifies the "From:" header of every message send on this list
HeaderRemove	string	Which headers should be removed
KeepOutmailPostings	int	How long mailings sent to the list should be kept
Keywords	string	Keywords which describe this list

ListName	string	The name of the list
MakePostsAnonymous	boolean	If true, posts to this list will be anonymous
MaxMembers	int	The maximum number of members this list is allowed
MaxMessageSize	int	The maximum size, in bytes, non-administrators may post to the list
MaxMessNum	int	The maximum number of messages that may be sent through the list in a 24-hour period
MaxPostsPerUser	int	The maximum number of posts a single person may post to the list
MaxQuoting	int	The maximum number of contiguous lines that may be quoted at once in a single message
MemberListSecurity	MemberListSecurityEnum	Who may access the list of members
MergeCapOverride	int	Whether this list should override the server scripting levels
MessageFooter	string	The default message footer
MessageHeader	string	The default message header
Moderated	ModeratedEnum	What level of moderation this list requires
ModifyHeaderDate	boolean	Whether the "Date:" header should automatically be rewritten to be the date and time a message is approved
MRIVisibility	MRIVisibilityEnum	Whether the Discussion Forum Interface for this list should be (v)isible, (h)idden or (i)nvisible
NameRequired	NameRequiredEnum	Whether the subscriber must provide a name when joining
NewSubscriberSecurity	NewSubscriberPolicyEnum	What security controls who may join the list
NoArchive	boolean	If true, this list maintains a message archive

NoBodyOk	boolean	If true, messages posted to this list may be blank
NoEmail	boolean	If true, this list rejects messages posted via email, and requires postings through the ListManager web interface
NoEmailSubscriptions	boolean	If true, this list prohibits subscriptions via email, and requires them through the ListManager web interface
NoListHeader	boolean	If true, the list-help headers should not be included
NoNNTP	boolean	If true, the list will not be accessible via NNTP
NoSearch	boolean	If true, the list's archives, if available, will not be searchable
NotifyHeldInterval	int	How frequently held members should be notified
OnlyAllowAdminSend	boolean	If true the list will only allow administrators to send
PasswordRequired	int	Whether a password is required when joining
PostPassword	PostPasswordEnum	Whether a member's password must be included in the message in order to be posted
PrivApprov	string	Sets rules for automatically approving applications to a private list
PurgeExpiredInterval	int	How long expired members should be kept in the database before being purged, in days
PurgeHeldInterval	int	How long held members should be kept in the database before being purged, in days
PurgeReferredInterval	int	How long referred members should be kept in the database before being purged, in days
PurgeUnapprovedInterval	int	How long unapproved

		members should be kept in the database before being purged, in days
PurgeUnconfirmedInterval	int	How long unconfirmed members should be kept in the database before being purged, in days
PurgeUnsubInterval	int	How long unsubscribed members should be kept in the database before being purged, in days
RecencyDayCount	int	How many days to check for recent mailings
RecencyEmailEnabled	boolean	If ture, recency checking/purging is enabled
RecencyMailCount	int	The number of mailings that should trigger recency limits
RecencyOperator	RecencyOperatorEnum	When checking for recency, should the number of mailings be (m)ore than, (e)qual to or (l)ess than the number in RecencyMailCount
RecencySequentialEnabled	boolean	If true, recency limits should be applied to sequential mailings
RecencyTriggeredEnabled	boolean	If true, recency limits should be applied to triggered mailings
RecencyWebEnabled	boolean	If true, recency limits should be applied to web mailings
RecipientLoggingLevel	LoggingLevelEnum	What level of delivery logging should there be? (N)o logging, non-SMTP errors (m), all (e)rrors, errors and successes (f)
ReferralsPerDay	int	The number of referrals allowed per member per day
ReleasePending	int	The length of time unapproved messages should be kept before being sent automatically

RemoveDuplicateCrossPostings	boolean	If true, if a list allows cross-postings, it should clean out duplicate recipients so that members of multiple lists only receive one copy of a message
ReplyTo	string	The "Reply-To:" header
RunProgAfterPosting	string	Program to execute after posting
RunProgAfterSub	string	Program to execute after subscribing
RunProgAfterUnsub	string	Program to execute after unsubscribing
RunProgBeforePosting	string	Program to execute before posting a message
RunProgBeforeSub	string	Program to execute before processing a subscribe
RunProgBeforeUnsub	string	Program to execute before processing an unsubscribe
ScriptingLevel	ScriptingLevelEnum	What level of scripting should this list allow
ShortDescription	string	A short description of this list
SimpleSub	boolean	If true, this list should only allow name, email address, or allow members to set other options (such as noack, norepro)
SMTPFrom	string	The SMTP "MAIL FROM:" header
SMTPHeaders	string	Text that should be included in the SMTP header of every message
SponsOrgID	string	Organization that is sponsoring this list (link to document, type is organization)
SubscribePassword	string	If this list is 'password' protected for new subscribers, what is the password?
SubscriptionReports	SubscriptionReportEnum	What subscription reports should be emailed to list administrators?
To	string	The default "To:" header

Visitors	boolean	If true, the list should allow non-members to see messages in the Discussion Forum Interface

Note: Since Microsoft .NET does not support NULL date/times, the date/time "2001-01-01 00:00:00" is considered null.

MailingStruct

The MailingStruct defines a mailing.

Name	**Type**	**Description**
AdditionalHeaders	string	Any additional headers (e.g., X-type headers)
Attachments	string	Any attachments
BypassModeration	boolean	If true, the mailing may bypass moderation
Campaign	string	What campaign the mailing should belong to
CharSetID	int	The ID of the charset for this mailing (from CharSetStruct)
DetectHtml	boolean	If true, the mailing should detect if the recipient has the ability to read HTML
DontAttemptAfterDate	dateTime	The "kill date" after which this mailing should no longer be delivered
EnableRecency	boolean	If true, recency checking should be done for this mailing
From	string	The header "From:"
HtmlMessage	string	The HTML portion of the mailing
HtmlSectionEncoding	int	The HTML message encoding
IsHtmlSectionEncoded	boolean	If true, the HTML portion of the mailing has been encoded
IsTextSectionEncoded	boolean	If true, the text portion of the mailing has been

		encoded
ListName	string	The list this mailing is posted to
MailingName	string	The name of the mailing
RecencyNumberOfMailings	int	If enabled, the number of mailings that should trigger recency limits.
RecencyWhich	string	When checking for recency, should the number of mailings be more than, equal to or less than the number in RecencyNumberof Mailings (from RecencyWhichEnum)
ReplyTo	string	The "Reply To:" header
ResendAfterDays	int	If set, means that when the message is approved, it should be resubmitted for automatic approval after so many days
SampleSize	int	The maximum number of recipients to receive this mailing. If not set, there is no maximum
ScheduledMailingDate	dateTime	The date and time the mailing is scheduled to be sent
Subject	string	The "Subject:" header
TextMessage	string	The text portion of the mailing
TextSectionEncoding	int	The encoding of the text section
Title	string	The title of the mailing
To	string	The "To:" header
TrackOpens	boolean	If true, this mailing should detect if recipients open it
RewriteDateWhenSent	boolean	If true, the mailing should have its "Date:" header rewritten to be the time it was actually sent, removing the original "Date:" header

Note: Since Microsoft .NET does not support NULL date/times, the date/time "2001-01-01 00:00:00" is considered null.

MemberStruct

Most member functions use the SimpleMemberStruct. The MemberStruct provides more information about members. It is used in the CreateManyMembers, and is returned by SelectMembers.

Name	Type	Description
MemberID	integer	Unique member ID
Additional	string	Additional information about this member (not used by ListManager)
ApprovalNeeded	boolean	Can this member bypass approval to send messages?
CanApprovePending	boolean	If true, and if this member is an admin, can this member approve pending (moderated) messages?
CleanAuto	boolean	If true, do not mark as 'held' this member if they bounce too much email (i.e.: let them bounce all they want)
Comment	string	Additional information about this member (not used by ListManager)
DateBounce	dateTime	The date of the most recent bounce
DateConfirm	dateTime	Date the member was last sent a 'confirm' message
DateHeld	dateTime	Date the member was held
DateJoined	dateTime	Date when member joined this list
DateUnsubscribed	dateTime	Date when member unsubscribed from this list
Demographics	ArrayOfDemographics	An array of KeyValueType(s) as defined by your database
EmailAddress	string	The email address of the member
ExpireDate	dateTime	The date the membership expired

FullName	string	The full name of the member
IsListAdmin	boolean	If true, the member is a list administrator
ListName	string	The name of the list this member belongs to
MailFormat	MailFormatEnum	What format does the user want to receive mail, (T)ext, (M)ultipart, or (H)TML.
MembershipKind	MemberKindEnum	The kind of subscription this member has ('mail', 'nomail', 'digest', 'mimedigest', 'index')
MemberStatus	MemberStatusEnum	What is the status of this member? ('normal', 'confirm', 'private', 'expired', 'held', 'unsub', 'needs-goodbye', 'needs-hello', 'needs-confirm')
NoRepro	boolean	If true, the member should receive a copy of his own postings
NotifyError	boolean	If true, the member (if a list administrator) should receive list error mail
NotifySubmission	boolean	If true, the member (if a list administrator) should receive notification of messages pending moderation
NumApprovalsNeeded	integer	The number of approvals this member needs before posting to the list unapproved
NumberOfBounces	integer	The number of bounces this member has had
Password	string	The member's password
ReadsHtml	boolean	If true, this member has been detected to read HTML
ReceiveAcknowlegment	boolean	If true, this member receives a message acknowledging a posting to the list
ReceiveAdminEmail	boolean	If true, this member (if a

		list administrator) should receive messages for list administrators
UserID	string	Additional information about this member (not used by ListManager)

Note: Since Microsoft .NET does not support NULL date/times, the date/time "2001-01-01 00:00:00" is considered null.

MemberBanStruct

The MemberBanStruct defines a member ban, which controls who may or may not join lists.

Name	Type	Description
BanLogic	BanLogicEnum	The type of ban this is
Domain	string	The domain this ban applies to
ListName	string	The list this ban applies to; NULL for a site or server ban
SiteName	string	The site this ban applies to; NULL for a list or server ban
UserName	string	The user name this ban applies to, if any

SegmentStruct

The SegmentStruct defines a segment (SQL subset) of a list.

Name	Type	Description
SegmentID	int	Unique identifier of this segment
Description	string	A description of this segment
ListName	string	The name of the list this segment is associated with

NumTestRecords	int	The number of records to retrieve when testing this segment
SegmentName	string	The name of this segment
SegmentType	SegmentTypeEnum	The type of segment, normal, triggered or sequential

SimpleMailingStruct

The SimpleMailingStruct is created from the ImportContent function. It may then be used in SendMailing or SendMailingDirect function.

Name	**Type**	**Description**
AdditionalHeaders	string	Any additional SMTP headers for the mailing
Attachments	string	Any attachments to the mailing (Note: this field should only be set from the results of ImportContent; it should not be set by the user)
CharSetID	int	The charset ID
From	string	The From header of the mailing
HtmlMessage	string	The HTML message
HtmlSectionEncoding	int	The encoding of the HTML section
IsHtmlSectionEncoded	boolean	If true, the HTML section is encoded
IsTextSectionEncoded	boolean	If true, the text section is encoded
Subject	string	The Subject header of the mailing
TextMessage	string	The text message
TextSectionEncoding	int	The encoding of the text section
To	string	The To header
Title	string	The title of the mailing

SimpleMemberStruct

The SimpleMemberStruct requires minimal information about a member, and is used in many member functions.

Name	Type	Description
MemberID	int	The unique identifier for this member
EmailAddress	string	The email address for this member
ListName	string	The list this member belongs to

TrackingSummaryStruct

The TrackingSummaryStruct is returned by the TrackingSummary function, and gives basic delivery and tracking information for a mailing.

Name	Type	Description
TransientFailure	string	Temporary failures due to non-permanent SMTP or DNS failures
Success	int	The number of successful recipients
Expired	int	The expiration date to send the mailing was reached before all attempts were completed
Paused	int	The recipient was not sent to because the mailing was paused by an administrator
MailMergeSkipped	int	The recipient was not processed due to the skip command in mailmerge
Active	int	The number of recipients currently receiving mailing
Opens	int	The total number of times a message is opened
Created	date	The date that the mailing

		was created as an outmail record for sending
NotAttempted	int	The number of recipients not yet attempted
Clickthroughs	int	The total number of tracked clickthroughs for a message
Title	string	The title of the mailing
TotalRecipients	int	The total number of recipients
PermanentFailure	int	Permanent failures due to permanent SMTP or DNS failures, receipt of a bounce message, invalid email address or failure to retrieve the message body from the database
TotalUndelivered	int	The total number of recipients who have been attempted but failed
MailMergeAbort	int	The recipient was not sent to because the mailing was aborted during mail merging by the abort command
UniqueOpens	int	The total number of recipients tracked opening a message
Clickstreams	int	The total number of tracked clickstreams for a message
Pending	int	The pending number of recipients
URLs	Array of UrlTrackingStruct	All tracked URLs
MailingID	int	The outmail_.MessageID_ value
Retry	int	The number of recipients waiting for mailings who have already been tried but who did not receive due to a transient error

URLTrackingStruct

The URLTrackingStruct is returned as part of the TrackingSummaryStruct.

Name	Type	Description
UniqueOpens	string	Number of unique clicks for this URL.
Opens	string	Total number of clicks for this URL
URL	string	The URL

Enumerations

BanLogicEnum

Specifies the kind of ban in the MemberBanStruct.

Key	Description
A	Accepted. Bans addresses other than accepted addresses from joining.
C	Conditionally accepted. Bans other addresses from joining, except those explicitly rejected with a "Reject" ban.
R	Banned. This member or domain may not join the list.

ConfirmUnsubEnum

Specifies whether unsubscribe requests should be confirmed.

Key	Description
0	Require unsubscribe confirmation when the

	request is questionable (e.g., the request is from a different address than the member's email address)
1	Always require unsubscribe confirmation
2	Never require unsubscribe confirmation

DeliveryReportEnum

Specifies at what interval(s) delivery reports should be emailed to administrators.

Key	**Value**
0	NoReports
1	ReportAfter1Hour
2	ReportAfter4Hours
4	ReportAfter12Hours
8	ReportAfter24Hours
16	ReportAfter48Hours

DocTypeEnum

Specifies the content document type.

Key	**Description**
BILLING	Billing invoice data
CONTENT	Content created to be sent to a list
MESSAGE	An internal message sometimes sent to a user or an administrator

EnableScriptingEnum

Specifies who may use scripting for a list.

Key	Description
0	Only administrators may use scripting
1	Any list member may use scripting
2	Scripting disabled

ListTypeEnum

Specifies the kind of list being created.

Key	Description
marketing	All list settings will be appropriate for a list intended for email marketing. The list will be configured so only administrators may send email to the list. If members reply to the message, their responses will go to the message's author. Open and Clickthrough tracking is on by default for all new mailings sent from an email marketing list. Full personalization of messages (conditional content, mail merge) is possible because full scripting is enabled for administrators. The list will accept messages from other lists as a child list. Mailing and clickthrough data will be kept forever. Messages sent to the list must be approved before they will be sent. The list settings may be changed later in the Utilities: List Settings menu.
announcement-moderated	All list settings will be appropriate for a one-way announcement list. The list will be configured so only administrators may send email to the list. If members reply to the message, their responses will go to the message's author. Open and Clickthrough tracking is off by default for all new mailings sent from an announcement list. Messages can be personalized because safe scripting is enabled for administrators. The list will not accept messages from other lists as a child list. Mailing and clickthrough data will be kept for 35 days. Messages sent to the list must be approved before they will be sent. The list settings may be changed later in the Utilities: List Settings menu.

discussion-moderated	All list settings will be appropriate for a moderated discussion list. The list will be configured so that all list members may send email to the list. If members reply to the message, their responses will go to the list. Messages sent to the list must be moderated by an administrator.
discussion-unmoderated	All list settings will be appropriate for an unmoderated discussion list. The list will be configured so that all list members may send email to the list. If members reply to the message, their responses will go to the list.

LoggingLevelEnum

Specifies the level of delivery logging for a list.

Key	**Description**
N	No logs
M	Displays errors but not SMTP transaction info
E	All errors, including SMTP transaction info
F	Errors and successes

MailFormatEnum

Specifies the mail format a member prefers.

Key	**Description**
T	Text only
H	HTML
M	Multipart-alternative

MailSectionEncodingEnum

Specifies the encoding for a section of a message.

Key	**Description**
8bit	8 bit encoding

quoted-printable	quoted-printable encoding
base64	base 64 encoding
7bit	7 bit encoding

MemberKindEnum

Specifies what kind of messages a member should receive from the list.

Key	**Description**
digest	A digest
mimedigest	A MIME digest
index	Subject line index
nomail	No mail
mail	Normal list mail

MemberListSecurityEnum

Specifies who may access the list of members.

Key	**Description**
0	Only administrators may access the list of members
1	List members may access the list of members
2	Anyone may access the list of members

MemberStatusEnum

Specifies the status of the member. Only members with status = normal will typically receive mail from a list.

Key	**Description**
normal	A normal member who receives mailings from the list
confirm	The person has requested to join this mailing list, has received an email message asking them to confirm

	their membership, but has not replied to the confirmation message.
private	The person has requested to join this mailing list, but because this mailing list is set to private, their request to join is waiting for an administrator to approve it by changing their membership to normal.
expired	The member had their membership set to expire on a certain date, and that date has past.
held	The member's email address was invalid, or bounced mailing list messages for several days.
unsub	The member was a member of the list at one time, but has since unsubscribed
referred	The member was invited to join using a referral, but has not yet joined the list.
needs-confirm	The member will receive a message asking them to confirm their membership, and then their status will be set to Confirm.
needs-hello	The member will receive a message welcoming them to the list, and their status will be Normal.
needs-goodbye	The member will receive a message informing them they have left the list, and their status will be Unsubscribed.

MessageTypeEnum

Specifies the type of document that should be sent.

Key	Description
confirm	The document which will be sent via email to new subscribers when they join the mailing list, and the list is set to confirm new subscribers.
hello	The document which will be sent via email to new members of this mailing list when they join. If blank, a default Hello message will be sent.
goodbye	The document which will be sent via email to members of this mailing list when they leave the mailing list. If blank, a default Goodbye message will be sent.
held	The document which will be sent via email to subscribers whose subscriptions are being held due to delivery problems. If blank, a default Held message will be sent.
private	The document which will be sent via email to new members who join but cannot receive any messages

until an administrator subscribes them to the list.

ModeratedEnum

Specifies the kind of moderation required by a list.

Key	**Description**
no	No moderation required
all	All messages must be moderated
number	The first *n* number of messages for a member must be moderated. This number is defined by the value for ApproveNum in the ListStruct

MriVisibilityEnum

Specifies whether the Discussion Forum Interface is visible, hidden, or unavailable. Used by the ListStruct.

Key	**Description**
V	List is visible
H	List is hidden; users may access the list by going to the list URL directly
I	List is invisible and inaccessible

NameRequiredEnum

Specifies whether a prospective member must provide a name when joining a list.

Key	**Description**
0	Providing a name is optional

1	Do not ask for a name
2	Providing a name is required

NewSubscriberPolicyEnum

Specifies what requirements there are (if any) for new members.

Key	Description
open	Anyone may join
private	Administrator must approve new members
closed	New members may only be added by administrator
password	Members may join if they supply the list password

PostPasswordEnum

Specifies whether the user's password must be included in a message to be posted.

Key	Description
0	Password required
1	Password not required

RecencyOperatorEnum

Specifies whether a list's recency limit should be for more, fewer or exactly the number specified by the RecencyMailCount in the ListStruct.

Key	Description
m	More than the number of messages specified
f	Fewer than the number of messages specified

e	Exactly the number of messages specified

RecencyWhichEnum

Specifies whether a mailing's recency limit should be for more, fewer or exactly the number specified by the RecencyNumberOfMailings in the MailingStruct.

Key	Description
m	More than the number of messages specified
f	Fewer than the number of messages specified
e	Exactly the number of messages specified

ScriptingLevelEnum

Specifies the scripting level for the list.

Key	Description
0	No scripting permitted
1	Safe scripting permitted
2	Full scripting permitted
3	Use server settings

SegmentTypeEnum

Specifies the type of segment.

Key	Description
normal	Normal segment
triggered	Triggered segment
sequential	Sequential segment

SubscriptionReportEnum

Specifies the type of subscription report that should be mailed to list administrators.

Key	Description
none	Do not email sub/unsub delivery reports to list admins
monthly	Monthly: send a standard monthly report on the 1st of the month
monthly-email	Monthly: send a detailed monthly report on the 1st of the month
monthly-nochart	Monthly: send a short monthly report on the 1st of the month
daily	Daily: send a standard report every day
daily-email	Daily: send a detailed report every day
daily-nochart	Daily: send a short report every day
monday	Weekly: every Monday send a standard weekly report
monday-email	Weekly: every Monday send a detailed weekly report
monday-nochart	Weekly: every Monday send a short weekly report
tuesday	Weekly: every Tuesday send a standard weekly report
tuesday-email	Weekly: every Tuesday send a detailed weekly report
tuesday-nochart	Weekly: every Tuesday send a short weekly report
wednesday	Weekly: every Wednesday send a standard weekly report
wednesday-email	Weekly: every Wednesday send a detailed weekly report
wednesday-nochart	Weekly: every Wednesday send a short weekly report
thursday	Weekly: every Thursday send a standard weekly report
thursday-email	Weekly: every Thursday send a detailed weekly report
thursday-nochart	Weekly: every Thursday send a short weekly report

friday	Weekly: every Friday send a standard weekly report
friday-email	Weekly: every Friday send a detailed weekly report
friday-nochart	Weekly: every Friday send a short weekly report
saturday	Weekly: every Saturday send a standard weekly report
saturday-email	Weekly: every Saturday send a detailed weekly report
saturday-nochart	Weekly: every Saturday send a short weekly report
sunday	Weekly: every Sunday send a standard weekly report
sunday-email	Weekly: every Sunday send a detailed weekly report
sunday-nochart	Weekly: every Sunday send a short weekly report

INDEX