

MachPanel Exchange API

Specification

Contents

Overview:	2
Getting Started:.....	3
Authentication:	3
Service Methods:	3
Service Responses:.....	7
Sample Code:	8
Authentication Header:	8
Customer:.....	8
Organization:.....	10
Domain:.....	10
Mailbox:	12
Mail Contacts:	27
Distribution Lists/Groups:	30
Service Queue/Jobs:	35
Reference Material:	36
General Guide Lines:	36
Response Codes:	36

Overview:

MachPanel API is basically a web service which uses SOAP messages to communicate to the Control Server and the client application. The service is hosted on same server as Control Server and same Port as Control Server. The URL of the service is as

<http://cpaddress/webservices/machpanelservice.asmx>

<https://cpaddress/webservices/machpanelservice.asmx>

To start communication with the control server through API, first we need the API to be enabled by the provider and/or reseller under whose authority the requests will be entertained. The API needs to be enabled by the provider and/or reseller from within the Control Panel after login.

The settings for MachPanel API are located under company profile, from there a provider and/or reseller can enable/disable the API and also can set valid authentication credentials (API Login & API Password) for the API.

The snapshot below highlights the navigation path for MachPanel API Settings:



MachPanel Exchange API is the extension to the existing MachPanel API and is a powerful tool that allows the developers to perform Exchange related tasks such as Add Exchange Organization, Add Exchange Domain, Add Exchange Mailbox etc. The developer can consume this API by using any of the current technologies such as ASP.NET, PHP and java. It enables the service providers

to manage their Exchange subscriptions from application other than the MachPanel and hence enables to customize user interface for Exchange service management.

Getting Started:

Authentication:

Every request for an operation through MachPanel API is authenticated with the provided credentials by the client application. These credentials are saved through Control Server as shown above. Only the authenticated request of an operation is entertained, otherwise an **“AuthenticationException”** is thrown.

Once the authentication process is successful, a call to the API service can be made to access the service methods. All the detailed code samples are added in “Sample Code” section.

Following is the list of service methods and their description:

Service Methods:

Following is the detailed description of the service methods section wise.

Customer:

CreateCustomer	Creates customer
SubscribeExchangeHosting	Subscribes Exchange hosting for a specific customer

Organization:

GetExchangeOrganizations	Gets list of all the organizations in the subscription
AddExchangeOrganization	Adds exchange organization in a specific subscription

RemoveExchangeOrganization	Removes exchange organization from the subscription
----------------------------	---

Domain:

GetExchangeDomains	Gets list of all domains in an organization
AddExchangeDomain	Adds domain to a specific organization
SetExchangeDomainAsDefault	Sets domain as organization default domain
UpdateExchangeDomainType	Updates domain type of a domain belonging to the specific organization
RemoveExchangeDomain	Removes domain from an exchange organization

Mailbox:

GetMailboxLanguages	Gets list of all mailbox languages
GetMailboxTimeZones	Gets list of all mailbox Time Zones
GetMailboxTypes	Gets Mailbox types
GetMailboxDatabases	Gets list of all mailbox databases
GetMailboxStatus	Gets list of all provided mailbox statuses
GetMailboxTemplates	Gets list of mailbox templates
GetMailboxListing	Gets list of all mailboxes in an organization
AddExchangeMailbox	Adds mailbox to a specific organization
DisableExchangeMailbox	Disables exchange mailbox
EnableExchangeMailbox	Enables exchange mailbox
RemoveExchangeMailbox	Remove mailbox from specific organization
GetMailboxGeneralSettings	Gets mailbox general settings
UpdateMailboxGeneralSettings	Updates mailbox general settings
GetMailboxEmailAddresses	Gets mailbox email addresses
AddMailboxEmailAddress	Adds mailbox email address
SetMailboxEmailAddressAsPrimary	Sets email address as Primary for a mailbox

SetMailboxEmailAddressAsPrimaryAndUPN	Sets email address as Primary and the UPN for a specific mailbox
RemoveMailboxEmailAddress	Removes mailbox email address
GetMailboxForwarding	Gets mailbox forwarding settings
SetMailboxForwarding	Sets mailbox forwarding settings
GetMailflowAddressListing	Gets mailbox mailflow address listing
GetMailboxMailFlowSettings	Gets mailbox mailflow settings
SetMailboxMailFlowSettings	Sets mailbox mailflow settings
GetMailboxAdvancedSettings	Gets mailbox advanced settings
SetMailboxAdvancedSettings	Sets mailbox advanced settings
SetMailboxLanguageAndTimeZone	Sets language and time zone of mailbox
GetMailboxSendOnBehalfOptionsList	Gets mailbox SendOnBehalf options list
GetMailboxSendAsOptionsList	Gets mailbox SendAs options list
GetMailboxFullAccessOptionsList	Gets mailbox FullAccess options list
GetMailboxSendOnBehalfSettings	Gets mailbox SendOnBehalf settings
GetMailboxSendAsSettings	Gets mailbox SendAs settings
GetMailboxFullAccessSettings	Gets mailbox FullAccess settings
SetMailboxSendOnBehalfSettings	Sets mailbox SendOnBehalf settings
SetMailboxSendAsSettings	Sets mailbox SendAs settings
SetMailboxFullAccessSettings	Sets mailbox FullAccess settings
UpdateMailboxPassword	Updates password for a mailbox
GetMailboxDatabase	Gets mailbox database
SetMailboxDatabase	Sets mailbox database
GetMailboxArchiveSettings	Gets mailbox archive settings
SetMailboxArchiveSettings	Sets mailbox archive settings
GetMailboxUMPolicies	Gets mailbox UM policies
GetMailboxUMSettings	Gets mailbox UM settings
EnableMailboxUMSettings	Enables mailbox UM settings
DisableMailboxUMSettings	Disables mailbox UM settings

SetMailboxUMSettings	Sets mailbox UM settings
SetMailboxUnifiedMessagingPIN	Sets mailbox UM PIN
GetAvailableGroupsForMailboxMembership	Gets available groups for mailbox membership
GetMailboxGroupMembership	Gets mailbox group membership
SetMailboxGroupMembership	Sets mailbox group membership
GetMailboxJournaling	Gets mailbox journaling settings
SetMailboxJournalingSettings	Sets mailbox journaling settings
EnableMailboxLitigationHold	Enables mailbox litigation hold
DisableMailboxLitigationHold	Disables mailbox litigation hold

Mail Contacts:

GetMailContacts	Gets mail contacts of an exchange organization
GetMailContactOptionalInformation	Gets optional information of a mail contact
AddMailContact	Adds mail contact for an organization
RemoveMailContact	Removes mail contact from an organization
GetMailContactGeneralSettings	Gets mail contact general settings
SetMailContactGeneralSettings	Sets mail contact general settings
GetMailContactMailFlowSettings	Gets mail contact mailflow settings
SetMailContactMailFlowSettings	Sets mail contact mailflow settings
GetAvailableMailContactAcceptRejectList	Gets available accept & reject list for a mail contact
EnableMailContact	Enables mail contact
DisableMailContact	Disables mail contact

Distribution Lists/Groups:

GetGroupListing	Gets distribution list/group listing
CreateGroup	Creates distribution list/group

RemoveGroup	Removes distribution list/group
GetGroupMembers	Gets distribution list/group members
AddGroupMember	Adds member to a distribution list/group
RemoveGroupMember	Removes member from distribution list/group
GetAvailableGroupMembers	Gets available distribution list/group members
GetGroupEmails	Gets emails in a distribution list/group
AddGroupEmail	Adds email to a distribution list/group
RemoveGroupEmail	Removes email from distribution list/group
GetGroupAdvancedSettings	Gets advanced settings of distribution list/group
SetGroupAdvancedSettings	Sets advanced settings of distribution list/group
GetGroupSendAsSettings	Gets SendAs settings of distribution list/group
SetGroupSendAsSettings	Sets SendAS settings of distribution list/group
GetGroupAvailableSendAs	Gets available SendAS listing of a distribution list/group
GetGroupMessageApprovalSettings	Gets message approval settings of a distribution list/group
SetGroupMessageApprovalSettings	Sets message approval settings of a distribution list/group

Service Queue/Jobs:

GetJobStatus	Gets status and complete detail including tries count, last try etc. of a service queue job or Exchange group jobs
--------------	--

Service Responses:

A detailed and correct response is very important in an API while using its services. A response should clearly describe the status and the completeness of the request. In case of any exception is occurred, the response should specify the exception type, its code and the detail description of the error occurred. For this purpose, MachPanel API includes a “ResponseArguments” object that is returned as a service response. It includes the following parameters:

EventRequested:	Name of the Event/Operation Requested
ResponseCode:	Response Code returned from an Operation
ResponseMessage:	Response Message from an Operation
ResponseStatus:	Response Status of an Operation
AssociatedKey:	Any Key/Identifier associated with operation performed

Sample Code:

Authentication Header:

```
MPAuthenticationHeader MPHeader = new MPAuthenticationHeader();  
MPHeader.UserName = "CompanyLogin@somedomain.com";  
MPHeader.UserPassword = "CompanyPassword";
```

Customer:

CreateCustomer

Creates a customer for the panel

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
Customer c = new Customer();
c.Address1 = "H # 123 ABC";
c.Address2 = "";
c.City = "New City";
c.CustomerNumber = "Cust 1";
c.DateCreated = DateTime.Today;
c.Fax = "";
c.FirstName = "Michal";
c.IsTaxable = true;
c.LastName = "Clark";
c.Mobile = "+123 456 789";
c.PassportLogin = "mylogin@mydomain.com";
c.PassportPassword = "myLoginPassw0rd";
c.Phone = "";
c.PhoneCountryCode = "";
c.PrimaryEmail = "myemail@mydomain.com";
c.Province = "Prov 1";
c.ResellerId = 0;
c.SecondaryEmail = "mysecondemail@mydomain.com";
c.Zip = "1234";
c.Country = new CountryInfo();
c.Country.CountryId = 5; // Can be obtained from svc.GetCountries()
c.State = new StateInfo();
c.State.StateId = 10; // Can be obtained from svc.GetStates(CountryId)
c.LocaleId = 20; // Can be obtained from svc.GetLocales()
ResponseArguments Args = svc.CreateCustomer(c);

```

SubscribeExchangeHosting

Subscribes Exchange Hosting for a customer

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ExchangeHosting Eh = new ExchangeHosting();
Eh.OU = new OrganizationalUnit();
Eh.OU.OrganizationName = "Test Organization";
Eh.Comments = "My Comments";
Eh.CustomerId = 15; // Where 15 is customer id
Eh.CycleId = 12; // Where 12 is payment cycle id
Eh.DomainName = "MyTestDomain.com";
Eh.HostingTypeId = 16; // Where 16 is hosting type id for Exchange
Eh.PackageId = 123; // Where 123 is package id
Eh.PaymentGroupId = 56;
Eh.ServerId = 71;
Eh.SignUpDate = DateTime.Today;
Eh.Status = ServiceStatus.Pending;
ResponseArguments Args = svc.SubscribeExchangeHosting(Eh, true);
// true is a flag to specify whether to send invoice email or not

```

Organization:

GetExchangeOrganizations

Use this function to fetch list of all organizations of specific Exchange subscription

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ExchangeOrganization[] Orgs = svc.GetExchangeOrganizations(123);
// Where 123 is subscription id
```

AddExchangeOrganization

Use this function to add an organization to the specific exchange subscription

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ExchangeOrganization Org = new ExchangeOrganization();
Org.OrganizationName = "Test Org";
Org.DomainName = "tetsdomain.com";
Org.CountryName = "United States";
ResponseArguments RA = svc.AddExchangeOrganization(Org, 123);
// Where 123 is subscription id
```

RemoveExchangeOrganization

Use this function to remove an organization from a specific Exchange subscription

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.RemoveExchangeOrganization(456);
// Where 456 is organization id
```

Domain:

GetExchangeDomains

Use this function to fetch list of all domains of an exchange organization

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ExchangeDomain[] Domains = svc.GetExchangeDomains(456); // Where 456 is organization id
```

AddExchangeDomain

Use this function to add mail domain to the exchange organization

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ExchangeDomain ED = new ExchangeDomain();
ED.DomainName = "testdomain.com";
ED.DomainType = DomainType.Authoritative;
ED.IsDefaultDomain = true;
ED.OrganizationId = 456; // Where 456 is organization id
ResponseArguments RA = svc.AddExchangeDomain(ED);
```

SetExchangeDomainAsDefault

Use this function to set mail domain as the default domain of an organization belonging to exchange subscription

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ExchangeDomain ED = new ExchangeDomain();
ED.DomainName = "testdomain.com";
ED.OrganizationId = 456; // Where 456 is organization id
ResponseArguments RA = svc.SetExchangeDomainAsDefault(ED);
```

UpdateExchangeDomainType

Use this function to update domain type of a mail domain of specific organization belonging to exchange subscription

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ExchangeDomain ED = new ExchangeDomain();
ED.DomainName = "testdomain.com";
ED.DomainType = DomainType.ExternalRelay;
ED.OrganizationId = 456; // Where 456 is organization id
ResponseArguments RA = svc.UpdateExchangeDomainType(ED);
```

RemoveExchangeDomain

Use this function to remove a specific mail domain of an organization belonging to exchange subscription

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ExchangeDomain ED = new ExchangeDomain();
ED.DomainName = "testdomain.com";
ED.OrganizationId = 456; // Where 456 is organization id
ResponseArguments RA = svc.RemoveExchangeDomain(ED);
```

Mailbox:

GetMailboxLanguages

Use this function to fetch list of all available languages to be used as the default language of the mailbox while adding or updating the mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] Languages = svc.GetMailboxLanguages();
```

GetMailboxTimeZones

Use this function to fetch list of all available time zones to be used as the time zone of the mailbox while adding or updating the mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] TimeZones = svc.GetMailboxTimeZones();
```

GetMailboxTypes

Use this function to fetch list of exchange mailbox types

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] MailboxTypes = svc.GetMailboxTypes();
```

GetMailboxDatabases

Use this function to fetch list of all available mailbox databases to be used while creating or updating exchange mailbox for a specific organization

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] Databases = svc.GetMailboxDatabases(456); // Where 456 is organization id
```

GetMailboxStatus

Use this function to get list of all mailbox status types

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] StatusList = svc.GetMailboxStatus();
```

GetMailboxTemplates

Use this function to get list of all available mailbox templates of an exchange organization in order to assign it to a mailbox while adding or updating the organization mailboxes

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] Templates = svc.GetMailboxTemplates(456); // Where 456 is organization id
```

GetMailboxListing

Use this function to get list of all mailboxes in an exchange organization

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ExchangeMailbox[] Mailboxes = svc.GetMailboxListing(456, "test", "");
// Where 456 is organization id, "test" is optional value for mailbox display name

```

AddExchangeMailbox

Use this function to add any type of mailbox to an exchange organization

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ExchangeMailbox EM = new ExchangeMailbox();
EM.MailboxType = MailboxTypes.User_Mailbox;
EM.DisplayName = "Test Mailbox";
EM.PrimarySMTPAddress = "email@testdomain.com";
EM.Language = "en-US";
EM.TimeZone = "Pacific Standard Time";
ResponseArguments RA = svc.AddExchangeMailbox(EM, 456); // Where 456 is organization id

```

DisableExchangeMailbox

Use this function to disable any mailbox belonging to exchange organization

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.DisableExchangeMailbox(456, "upn@testdomain.com", true);
// Where 456 is organization id, "upn@testdomain.com" is mailbox UPN
// and 'true' parameter is either add job or process immediately

```

EnableExchangeMailbox

Use this function to enable any mailbox belonging to exchange organization

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.EnableExchangeMailbox(456, "upn@testdomain.com", true);
// Where 456 is organization id, "upn@testdomain.com" is mailbox UPN
// and 'true' parameter is either add job or process immediately

```

RemoveExchangeMailbox

Use this function to remove any mailbox from an organization belonging to exchange subscription

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.RemoveExchangeMailbox(456, "upn@testdomain.com", true);
// Where 456 is organization id, "upn@testdomain.com" is mailbox UPN
// and 'true' parameter is either add job or process immediately
```

GetMailboxGeneralSettings

Use this function to fetch mailbox general settings

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MailboxGeneralSettings MGS = svc.GetMailboxGeneralSettings(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

UpdateMailboxGeneralSettings

Use this function to update mailbox general settings

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MailboxGeneralSettings MGS = new MailboxGeneralSettings();
MGS.DisplayName = "Test Name";
MGS.HideFromAddressList = true;
ResponseArguments RA = svc.UpdateMailboxGeneralSettings(456, "upn@testdomain.com", MGS);
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

GetMailboxEmailAddresses

Use this function to fetch list of all email addresses of a mailbox belonging to exchange organization


```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MailboxEmailAddress[] MEA = svc.GetMailboxEmailAddresses(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN

```

AddMailboxEmailAddress

Use this function to add email address to a mailbox

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MailboxEmailAddress MEA = new MailboxEmailAddress();
MEA.EmailAddress = "email1@testdomain.com";
ResponseArguments RA = svc.AddMailboxEmailAddress(456, "upn@testdomain.com", MEA);
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN

```

SetMailboxEmailAddressAsPrimary

Use this function to set provided email address as the primary address of the mailbox

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.SetMailboxEmailAddressAsPrimary(456, "upn@testdomain.com",
"email1@testdomain.com");
// Where 456 is organization id, "upn@testdomain.com" is mailbox UPN
// and "email1@testdomain.com" is the email address

```

SetMailboxEmailAddressAsPrimaryAndUPN

Use this function to set provided email address as the primary address and UPN of the mailbox

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.SetMailboxEmailAddressAsPrimaryAndUPN(456, "upn@domain.com",
"email1@testdomain.com");
// Where 456 is organization id, "upn@domain.com" is mailbox UPN
// and "email1@testdomain.com" is the email address

```

RemoveMailboxEmailAddress

Use this function to remove email address belonging to the specific exchange mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MailboxEmailAddress MEA = new MailboxEmailAddress();
MEA.EmailAddress = "email1@testdomain.com";
ResponseArguments RA = svc.RemoveMailboxEmailAddress(456, "upn@testdomain.com", MEA);
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

GetMailboxForwarding

Use this function to fetch mailbox forwarding settings of a specific mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
Forwarding FWD = svc.GetMailboxForwarding(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

SetMailboxForwarding

Use this function to update mailbox forwarding settings of a specific mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
Forwarding FWD = new Forwarding();
FWD.Enabled = true;
FWD.DeliverAndForward = true;
FWD.ForwardTo = "group1@testdomain.com";
ResponseArguments RA = svc.SetMailboxForwarding(456, "upn@testdomain.com", FWD);
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

GetMailflowAddressListing

Use this function to fetch list of mailflow addresses of a mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MailflowAddress[] MFA = svc.GetMailflowAddressListing(456, "upn@testdomain.com", true,
false);
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

GetMailboxMailFlowSettings

Use this function to fetch mailflow settings of an exchange mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MailflowSettings MFS = svc.GetMailboxMailFlowSettings(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

SetMailboxMailFlowSettings

Use this function to update mailflow settings of an exchange mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MailflowSettings MFS = new MailflowSettings();
MFS.MaxSendingSizeMB = 20;
MFS.MaxReceivingSizeMB = 10;
ResponseArguments RA = svc.SetMailboxMailFlowSettings(456, "upn@testdomain.com", MFS);
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

GetMailboxAdvancedSettings

Use this function to fetch mailbox advanced settings

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MailboxAdvancedSettings MAS = svc.GetMailboxAdvancedSettings(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

SetMailboxAdvancedSettings

Use this function to update mailbox advanced settings

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MailboxAdvancedSettings MAS = new MailboxAdvancedSettings();
MAS.IssueWarningAtMB = 20;
MAS.ProhibitSendAtMB = 30;
MAS.ProhibitSendAndReceiveAtMB = 40;
MAS.OWAPublicFolderEnabled = true;
ResponseArguments RA = svc.SetMailboxAdvancedSettings(456, "upn@testdomain.com", MAS);
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

SetMailboxLanguageAndTimeZone

Use this function to update language and time zone of an exchange mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.SetMailboxLanguageAndTimeZone(456, "upn@testdomain.com",
"en-US", "Pacific Standard Time");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

GetMailboxSendOnBehalfOptionsList

Use this function to fetch list of all available SendOnBehalf options including mailbox, group etc. to be used while updating mailbox SendOnBehalf permission settings

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] OptionList = svc.GetMailboxSendOnBehalfOptionsList(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

GetMailboxSendAsOptionsList

Use this function to fetch list of all available SendAs options to be used while updating mailbox SendAs permission settings

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] OptionList = svc.GetMailboxSendAsOptionsList(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

GetMailboxFullAccessOptionsList

Use this function to fetch list of all available FullAccess options to be used while updating mailbox FullAccess permission settings

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] OptionList = svc.GetMailboxFullAccessOptionsList(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

GetMailboxSendOnBehalfSettings

Use this function to fetch mailbox SendOnBehalf permission settings

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] OptionList = svc.GetMailboxSendOnBehalfSettings(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

GetMailboxSendAsSettings

Use this function to fetch mailbox SendAs permission settings

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] OptionList = svc.GetMailboxSendAsSettings(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

GetMailboxFullAccessSettings

Use this function to fetch mailbox FullAccess permission settings

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] OptionList = svc.GetMailboxFullAccessSettings(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN

```

SetMailboxSendOnBehalfSettings

Use this function to update mailbox SendOnBehalf permission settings

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] OptionsList = new string[2];
OptionsList[0] = "mailbox1@testdomain.com";
OptionsList[1] = "mailbox2@testdomain.com";
ResponseArguments RA = svc.SetMailboxSendOnBehalfSettings(456, "upn@testdomain.com",
OptionsList);
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN

```

SetMailboxSendAsSettings

Use this function to update mailbox SendAs permission settings

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] OptionsList = new string[2];
OptionsList[0] = "mailbox1@testdomain.com";
OptionsList[1] = "mailbox2@testdomain.com";
ResponseArguments RA = svc.SetMailboxSendAsSettings(456, "upn@testdomain.com",
OptionsList);
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN

```

SetMailboxFullAccessSettings

Use this function to update mailbox FullAccess permission settings

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] OptionsList = new string[2];
OptionsList[0] = "mailbox1@testdomain.com";
OptionsList[1] = "mailbox2@testdomain.com";
ResponseArguments RA = svc.SetMailboxFullAccessSettings(456, "upn@testdomain.com",
OptionsList);
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN

```

UpdateMailboxPassword

Use this function to update mailbox password

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.UpdateMailboxPassword(456, "upn@testdomain.com",
"newpassword");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN

```

GetMailboxDatabase

Use this function to fetch database name assigned to the specific mailbox

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string Database = svc.GetMailboxDatabase(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN

```

SetMailboxDatabase

Use this function to update mailbox database

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.SetMailboxDatabase(456, "upn@testdomain.com", "DBName", true,
10);
// Where 456 is organization id, "upn@testdomain.com" is mailbox UPN
// "DBName" is database name, "true" is the flag representing SkipCorruptedMessages
// and "10" represents number of messages to skip

```

GetMailboxArchiveSettings

Use this function to fetch archive settings of a specific exchange mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ArchiveSettings ASettings = svc.GetMailboxArchiveSettings(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

SetMailboxArchiveSettings

Use this function to update archive settings of a specific exchange mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ArchiveSettings ASettings = new ArchiveSettings();
ASettings.ArchiveEnabled = true;
ASettings.ArchiveDBName = "ArchiveDBName";
ASettings.ArchiveQuotaMB = 10;
ASettings.WarningQuotaMB = 10;
ResponseArguments RA = svc.SetMailboxArchiveSettings(456, "upn@testdomain.com",
ASettings);
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

GetMailboxUMPolicies

Use this function to fetch list of all available unified messaging policies for a specific exchange mailbox which can be used later for updating UM Policies of that mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] Policies = svc.GetMailboxUMPolicies(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

GetMailboxUMSettings

Use this function to fetch mailbox Unified Messaging settings


```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
UMSettings UMS = svc.GetMailboxUMSettings(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

EnableMailboxUMSettings

Use this function to enable Unified Messaging for a specific exchange mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
UMSettings UMS = new UMSettings();
UMS.UMEnabled = true;
UMS.UMPolicy = "Default Policy";
UMS.MailboxExtension = 123;
UMS.RequirePINResetOnLogin = true;
ResponseArguments RA = svc.EnableMailboxUMSettings(456, "upn@testdomain.com", UMS);
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

DisableMailboxUMSettings

Use this function to disable Unified Messaging for a specific exchange mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.DisableMailboxUMSettings(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

SetMailboxUMSettings

Use this function to update Unified Messaging settings for a specific exchange mailbox

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
UMSettings UMS = new UMSettings();
UMS.UMEnabled = true;
UMS.UMPolicy = "Default Policy";
UMS.MailboxExtension = 123;
UMS.RequirePINResetOnLogin = true;
ResponseArguments RA = svc.SetMailboxUMSettings(456, "upn@testdomain.com", UMS);
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN

```

SetMailboxUnifiedMessagingPIN

Use this function to update PIN for Unified Messaging settings of a specific exchange mailbox

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.SetMailboxUnifiedMessagingPIN(456, "upn@testdomain.com", 345,
true);
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN

```

GetAvailableGroupsForMailboxMembership

Use this function to fetch all available distribution lists/groups to be used for mailbox membership with groups

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] Groups = svc.GetAvailableGroupsForMailboxMembership(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN

```

GetMailboxGroupMembership

Use this function to fetch list of all associated distribution lists/groups of a specific exchange mailbox

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] Groups = svc.GetMailboxGroupMembership(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN

```

SetMailboxGroupMembership

Use this function to update mailbox association with distribution lists/groups for a specific exchange mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] GroupEmails = new string[2];
GroupEmails[0] = "email1@testdomain.com";
GroupEmails[1] = "email2@testdomain.com";
ResponseArguments RA = svc.SetMailboxGroupMembership(456, "upn@testdomain.com",
GroupEmails);
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

GetMailboxJournaling

Use this function to fetch journaling settings of a specific exchange mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
JournalingSettings JS = svc.GetMailboxJournaling(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

SetMailboxJournalingSettings

Use this function to update journaling settings for a specific exchange mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
JournalingSettings JS = new JournalingSettings();
JS.JournalingEnabled = true;
JS.Scope = "Internal";
JS.SendEmailTo = "email3@testdomain.com";
ResponseArguments RA = svc.SetMailboxJournalingSettings(456, "upn@testdomain.com", JS);
// Where 456 is organization id, "upn@testdomain.com" is mailbox UPN
```

EnableMailboxLitigationHold

Use this function to enable mailbox litigation hold for a specific exchange mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.EnableMailboxLitigationHold(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

DisableMailboxLitigationHold

Use this function to disable mailbox litigation hold for a specific exchange mailbox

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.DisableMailboxLitigationHold(456, "upn@testdomain.com");
// Where 456 is organization id and "upn@testdomain.com" is mailbox UPN
```

Mail Contacts:

GetMailContacts

Use this function to fetch list of all mail contacts in an exchange organization

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MailContact[] MC = svc.GetMailContacts(456); // Where 456 is organization id
```

GetMailContactOptionalInformation

Use this function to fetch optional information of a specific mail contact

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ContactInformation CI = svc.GetMailContactOptionalInformation(456,
"contact1@testdomain.com");
// Where 456 is organization id and "contact1@testdomain.com" is external email address
```

AddMailContact

Use this function to add mail contact for a specific exchange organization

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MailContact MC = new MailContact();
MC.DisplayName = "Test Contact";
MC.ExternalEmail = "testcontact@testdomain.com";
MC.HideFromAddressLists = true;
ResponseArguments RA = svc.AddMailContact(456, MC); // Where 456 is organization id
```

RemoveMailContact

Use this function to remove mail contact of a specific exchange organization

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.RemoveMailContact(456, "contact1@testdomain.com");
// Where 456 is organization id and "contact1@testdomain.com" is external email
```

GetMailContactGeneralSettings

Use this function to fetch general settings of a mail contact

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MailContact MC = svc.GetMailContactGeneralSettings(456, "contact1@testdomain.com");
// Where 456 is organization id and "contact1@testdomain.com" is external email
```

SetMailContactGeneralSettings

Use this function to update general settings of a mail contact

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MailContact MCGS = new MailContact();
MCGS.DisplayName = "new display name";
MCGS.ExternalEmail = "newcontact1@testdomain.com";
MCGS.SIPAddress = "newcontact1sip@testdomain.com";
MCGS.HideFromAddressLists = true;
ResponseArguments RA = svc.SetMailContactGeneralSettings(456, "Contact1",
"contact1@testdomain.com", MCGS);
// Where 456 is organization id, "Contact1" is name and "contact1@testdomain.com" is
// external email

```

GetMailContactMailFlowSettings

Use this function to fetch mail flow settings of a mail contact

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MailContactMailflowSettings MMS = svc.GetMailContactMailFlowSettings(456,
"contact1@testdomain.com");
// Where 456 is organization id and "contact1@testdomain.com" is external email

```

SetMailContactMailFlowSettings

Use this function to update mail flow settings of a mail contact

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MailContactMailflowSettings MMF = new MailContactMailflowSettings();
MMF.MaxReceivingSizeMB = 100;
MMF.AcceptMessageFrom = new string[] { "mailbox1@testdomain.com" };
MMF.RejectMessageFrom = new string[] { "mailbox2@testdomain.com" };
ResponseArguments RA = svc.SetMailContactMailFlowSettings(456,
"contact1@testdomain.com", MMF);
// Where 456 is organization id and "contact1@testdomain.com" is external email

```

GetAvailableMailContactAcceptRejectList

Use this function to fetch all available members list that can be used as Accept and Reject option list while updating mail contact mail flow settings

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] MembersList = svc.GetAvailableMailContactAcceptRejectList(456);
// Where 456 is organization id
```

EnableMailContact

Use this function to enable a specific mail contact of an exchange organization

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.EnableMailContact(456, "contact1@testdomain.com");
// Where 456 is organization id and "contact1@testdomain.com" is external email
```

DisableMailContact

Use this function to disable a specific mail contact of an exchange organization

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.DisableMailContact(456, "contact1@testdomain.com");
// Where 456 is organization id and "contact1@testdomain.com" is external email
```

Distribution Lists/Groups:

GetGroupListing

Use this function to fetch list of all distribution lists/groups in an exchange organization

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
DistributionList[] DLs = svc.GetGroupListing(456); // Where 456 is organization id
```

CreateGroup

Use this function to create a distribution list/group for a specific exchange organization

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
DistributionList DL = new DistributionList();
DL.ListType = DistributionListType.SimpleDistributionGroup;
DL.DisplayName = "First Group";
DL.PrimaryEmail = "dl1@testdomain.com";
DL.RequireSenderAuthEnabled = true;
DL.ManagedBy = new string[] { "mailbox1@testdomain.com" };
DL.ListMembers = new string[] { "contact1@testdomain.com" };
ResponseArguments RA = svc.CreateGroup(456, DL); // Where 456 is organization id

```

RemoveGroup

Use this function to remove a distribution list/group from a specific organization

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.RemoveGroup(456, "dl1@testdomain.com");
// Where 456 is organization id and "dl1@testdomain.com" is email address

```

GetGroupMembers

Use this function to fetch list of all the members in a specific distribution list/group

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] Members = svc.GetGroupMembers(456, "dl1@testdomain.com");
// Where 456 is organization id and "dl1@testdomain.com" is email address

```

AddGroupMember

Use this function to add a member in a specific distribution list/group belonging to an exchange organization


```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.AddGroupMember(456, "dl1@testdomain.com",
"member1@testdomain.com");
// Where 456 is organization id, "dl1@testdomain.com" is Group email
// and "member1@testdomain.com" is member email

```

RemoveGroupMember

Use this function to remove any member from a specific distribution list/group of exchange organization

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.RemoveGroupMember(456, "dl1@testdomain.com",
"member1@testdomain.com");
// Where 456 is organization id, "dl1@testdomain.com" is Group email
// and "member1@testdomain.com" is member email

```

GetAvailableGroupMembers

Use this function to fetch list of all available members filtered by the distribution list type, to be used while updating distribution list/group

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] AvailableMembers = svc.GetAvailableGroupMembers(456,
DistributionListType.SimpleDistributionGroup); // Where 456 is organization id

```

GetGroupEmails

Use this function to fetch list of all emails of a specific distribution list/group

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
GroupEmailAddress[] Emails = svc.GetGroupEmails(456, "dl1@testdomain.com");
// Where 456 is organization id and "dl1@testdomain.com" is group email

```

AddGroupEmail

Use this function to add email address in a specific distribution list/group of an exchange organization

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
GroupEmailAddress GEA = new GroupEmailAddress();
GEA.EmailAddress = "email1@testdomain.com";
GEA.IsPrimary = true;
ResponseArguments RA = svc.AddGroupEmail(456, "dl1@testdomain.com", GEA);
// Where 456 is organization id and "dl1@testdomain.com" is Group email
```

RemoveGroupEmail

Use this function to remove email address from a specific distribution list/group of an exchange organization

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RA = svc.RemoveGroupEmail(456, "dl1@testdomain.com",
"email1@testdomain.com");
// Where 456 is organization id, "dl1@testdomain.com" is Group email
// and "email1@testdomain.com" is email address
```

GetGroupAdvancedSettings

Use this function to fetch advanced settings of a distribution list/group of an exchange organization

```
MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
GroupAdvancedSettings Settings = svc.GetGroupAdvancedSettings(456,
"dl1@testdomain.com");
// Where 456 is organization id and "dl1@testdomain.com" is group email
```

SetGroupAdvancedSettings

Use this function to update advanced settings of a distribution list/group of an exchange organization

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
GroupAdvancedSettings DLAS = new GroupAdvancedSettings();
DLAS.DisplayName = "Test Name";
DLAS.MaxReceivingSizeMB = 200;
DLAS.HideFromALs = true;
DLAS.OOFReplyToOriginator = true;
DLAS.RequireSenderAuthEnabled = true;
DLAS.DReportOption = DeliveryReports.SendToMessageOriginator;
DLAS.ManagedBy = new string[] { "mailbox1@testdomain.com" };
DLAS.AcceptMessageFrom = new string[] { "contact1@testdomain.com" };
DLAS.RejectMessageFrom = new string[] { "contact2@testdomain.com" };
ResponseArguments RA = svc.SetGroupAdvancedSettings(456, "dl1@testdomain.com", DLAS);
// Where 456 is organization id and "dl1@testdomain.com" is Group email

```

GetGroupSendAsSettings

Use this function to fetch SendAs settings of a distribution list/group of an exchange organization

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] SendAsSettings = svc.GetGroupSendAsSettings(456, "dl1@testdomain.com");
// Where 456 is organization id and "dl1@testdomain.com" is group email

```

SetGroupSendAsSettings

Use this function to update SendAs settings of a distribution list/group of an exchange organization

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] SendAs = new string[] { "mailbox5@testdomain.com" };
ResponseArguments RA = svc.SetGroupSendAsSettings(456, "dl1@testdomain.com", SendAs);
// Where 456 is organization id and "dl1@testdomain.com" is group email

```

GetGroupAvailableSendAs

Use this function to fetch list of all available SendAs options to be used while updating SendAs members of the distribution list/group

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
string[] SendAsSettings = svc.GetGroupAvailableSendAs(456, "dl1@testdomain.com");
// Where 456 is organization id and "dl1@testdomain.com" is group email

```

GetGroupMessageApprovalSettings

Use this function to fetch message approval settings of a distribution list/group

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MessageApprovalSettings MAS = svc.GetGroupMessageApprovalSettings(456,
"d11@testdomain.com");
// Where 456 is organization id and "dl1@testdomain.com" is group email

```

SetGroupMessageApprovalSettings

Use this function to update message approval settings of a distribution list/group

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
MessageApprovalSettings MAS = new MessageApprovalSettings();
MAS.ApprovalEnabled = true;
MAS.Moderators = new string[] { "mailbox1@testdomain.com" };
MAS.SendersDontRequireApproval = new string[] {"mailbox6@testdomain.com" };
MAS.NotifyOption = ApprovalNotification.NotifyOrganizationSenders;
ResponseArguments RA = svc.SetGroupMessageApprovalSettings(456, "dl1@testdomain.com",
MAS); // Where 456 is organization id and "dl1@testdomain.com" is group email

```

Service Queue/Jobs:

GetJobStatus

Use this function to get job status and complete detail including tries count, last try etc. of a service queue job or Exchange group jobs

```

MachPanelService svc = new MachPanelService();
svc.Url = "http://localhost:1234";
svc.MPAAuthenticationHeaderValue = MPHeader;
ResponseArguments RAg = svc.GetJobStatus(204, "11");
// Where 204 is Subscription/Hosting ID & '11' is Queue or Exchange Group Job ID

```

Reference Material:

General Guide Lines:

For general guidelines and sample asp.net test application, please follow below links:

<http://kb.machsol.com/Knowledgebase/Article/50310>

<http://kb.machsol.com/Knowledgebase/Article/50309>

For complete reference of the classes used and member functions detail, please check web version of the MachPanel API Documentation.

Response Codes:

Codes	Description
Success	The process was successful
Authentication_Failed	Authentication failed for the provided user name and password.
Required_Field_Missing	Some mandatory field is missing in the request.
Invalid_Email_Address	Email address is not in a correct format.
Invalid_Domain_Address	Domain address is not in a correct format.
Domain_Already_Exists	Failed to add the domain. The requested Domain already exists in the system
Organizational_Unit_Already_Exists	Failed to add Organization unit. It already exists.
Organization_User_Already_Exists	User already exists in the organization.
Database_Error	Error while connecting to the database.
Unknown_Error	Unknown error occurred. Please see detail attached.
Read_Only_Property	The property can't be set. It is read-only.

Customer_Already_Exists	Failed to add customer. Customer already exists.
Invalid_Portal_Name	Portal name is incorrect.
Invalid_CRM_Organization_Unique_Name	Organization Unique Name is not in a correct format.
Invalid_CRM_Organization_Display_Name	Organization Display Name is not in a correct format.
Duplicate_Record	A similar record already exists.
Invalid_Operation_Request	The requested operation is not supported.
Credit_Card_Already_Exists	Adding new credit card failed. Card already exists.
Invalid_PhoneFaxMob_Number	Invalid format for the Phone, Fax or Mobile number.
Invalid_Credit_Card_Customer	Credit Card Customer is not valid
Client_Limit_Reached	Quota limit exceeded for the desired request.
Invalid_Field_Value	Provided value is not valid for the relevant field or property.
Record_Not_Found	Record was not found in the database.
Exempted_Domain	Provider/Reseller has added this domain into exempted domains list.
Domain_Exists_For_Another_Customer	Domain already exists under another Customer.
Provider_Domain_Should_Be_False	Provider domain flag should be false.
Domain_Quota_Reached	Domain cannot be added, quota is consumed.
Domain_DNS_Provisioning_Failed	DNS provisioning failed for Domain.
Domain_Add_Failed	Adding domain is failed.
Domain_Set_Default_Failed	Set domain as default is failed.
Domain_Update_Type_Failed	Updating domain type is failed.
Domain_In_Use	Removing domain is failed because domain is in use.
Domain_Remove_Failed	Removing domain is failed due to the attached issue detail.
Default_Domain_Remove_Failed	Default domain cannot be removed, set any other domain as default and try again.
Subscription_Not_Active	Subscription is not active for management.
Domain_Not_Exist	Domain doesn't exist for management.
Organization_Quota_Reached	Organization cannot be added, quota is consumed.
No_Active_Mailbox_Server	No active server found for mailbox creation.
Provider_Domain_Required	Provider access domain is required
Organization_Add_Failed	Adding organization is failed, please see detail attached.

Subscription_Not_Found	Subscription not found for management.
Organization_Remove_Failed	Removing organization is failed, please see detail attached.
Quota_Exceeded	Quota exceeded for specific resource.
Password_Policy_Not_Matched	Password is not matched to the password policy.
UPN_Already_Exists	UPN already exists.
MailContact_Already_Exists	Mail Contact already exists.