



FaxCore eV5 API Guide

Table of Contents

Disclaimers	5
Support	6
U.S. & Europe	6
Asia Pacific	6
FaxCore Web Service	7
Operations	7
<i>ActivateUser</i>	7
<i>ActivateUsers</i>	8
<i>AssignMessage</i>	9
<i>AuthenticateUsers</i>	10
<i>CancelMessage</i>	11
<i>CreateInboundRoute</i>	12
<i>CreateNewUser1</i>	13
<i>CreateNewUser2</i>	15
<i>CreateNewUser3</i>	16
<i>DeactivateUser</i>	18
<i>DeactivateUsers</i>	19
<i>DelegateMessage</i>	20
<i>DeleteMessage</i>	23
<i>DisabledInboudRoute</i>	24
<i>DownloadMessageByte1</i>	24
<i>DownloadMessageByte2</i>	25
<i>DownloadMessageByte3</i>	26
<i>DownloadMessageByte4</i>	27
<i>ForwardToUser</i>	28
<i>GetDomainList</i>	29
<i>GetMessageDetails1</i>	30
<i>GetMessageDetails2</i>	32
<i>GetMessageList3</i>	35
<i>GetMessageList2</i>	36
<i>GetMessageList1</i>	37

<i>GetMessageList8</i>	38
<i>GetMessageList4</i>	39
<i>GetMessageList5</i>	40
<i>GetMessageList6</i>	41
<i>GetMessageList7</i>	42
<i>GetMessageStatus</i>	43
<i>GetMsgFolders</i>	44
<i>GetTrackRecords</i>	45
<i>GetUserDetailsByUserID</i>	46
<i>GetUserDetailsByUserName</i>	47
<i>GetUserFaxSettingByUserID</i>	49
<i>GetUserFaxSettingByUserName</i>	50
<i>GetUserListWithDomainID</i>	51
<i>GetUserListWithDomainName</i>	52
<i>IsMessageDeleted</i>	53
<i>IsMessageRead</i>	54
<i>MarkMsgAsDownloaded</i>	54
<i>MarkMsgAsRead</i>	55
<i>MarkMsgAsUnread</i>	56
<i>MoveMessage</i>	57
<i>RemoveUser</i>	58
<i>RetryFailed</i>	58
<i>SearchIBRouteRules</i>	59
<i>SendCompleteMessage</i>	60
<i>BasicSendMessageWithBytes</i>	63
<i>SendCompleteMessageWithPresetAgents</i>	65
<i>SendCompleteMessageWithPresetAgentsAndByPassRender</i>	68
<i>SendMixedMode</i>	71
<i>ServerStatus</i>	72
<i>UpdateTrackRecord</i>	73
<i>UpdateUser-basic</i>	74
<i>UpdateUser-extra</i>	75

<i>UpdateUser-advance</i>	76
Hardware Requirements	78
Dialogic Brooktrout Fax Boards Supported.....	78
• Fax Board: Fax communication device present	78
Getting started with FaxCore eV5	79
Login to FaxCore eV5 for the first time	79
Assigning Ports Licensing	80

Disclaimers

FaxCore and FaxCore eV5 are registered trademarks of FaxCore, Inc. Windows is a registered trademark of Microsoft Corporation. All other trademarks are the property of their respective owners.

Support

FaxCore customers are encouraged to purchase a FaxCore Support Contract which provides them with unlimited telephone, email and web based product support from Monday to Friday 8.00 am to 5.00 pm Mountain time (GMT -7). FaxCore Partners can also provide on-site support and training when needed. FaxCore Support Contract holders are entitled to all FaxCore product service packs and upgrades at no additional charge.

Web Support: www.faxcore.com

This site contains technical information as well as a list of frequently asked questions.

U.S. & Europe

Email Support: Email support is available at support@faxcore.com

Phone Support: FaxCore support technicians can be reached at +1 (720) 870 2900

Asia Pacific

Email Support: Email support is available at support@faxcore.com.my

Phone Support: FaxCore support technicians can be reached at +60 (3) 3324 3226

FaxCore Web Service

Operations

Below is the list of operations available along with sample XML request and response. Further detail of parameters and response result are available in a separate document wfapi_params_v3.

All parameters required to submit as part of the request, even if the value remains as empty or null.

ActivateUser

Set user account to active state.

Input:

Parameters	Type	Description
adminID	long/bigint	Required. Administrator user unique id.
userID	long/bigint	Required. User ID to activate.

Output:

Parameters	Type	Description
ActivateUserResult	string	Result info: OK – success/completed INVALID_USER – invalid user ID supplied INVALID_ADMIN – invalid admin ID supplied.

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:ActivateUser>
      <ser:adminID>20011</ser:adminID>
      <ser:userID>20011</ser:userID>
    </ser:ActivateUser>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ActivateUserResponse xmlns="http://faxcoreserver/services/">
      <ActivateUserResult>INVALID_ADMIN</ActivateUserResult>
    </ActivateUserResponse>
  </soap:Body>
</soap:Envelope>
```

ActivateUsers

Set a list of users to active state.

Input:

Parameters	Type	Description
adminID	long/bigint	Required. Administrator user unique id.
userIDList	array of long	Required. User IDs to activate.

Output:

Parameters	Type	Description
ActivateUserResult	string	Result info: OK – success/completed SOME_INVALID_USER – invalid user ID supplied INVALID_ADMIN – invalid admin ID supplied.

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:ActivateUsers>
      <ser:adminID>2010211</ser:adminID>
      <ser:userIDList>
        <!--One or more repetitions:-->
        <ser:long>20011</ser:long>
        <ser:long>20012</ser:long>
      </ser:userIDList>
    </ser:ActivateUsers>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ActivateUsersResponse xmlns="http://faxcoreserver/services/">
      <ActivateUsersResult>SOME_INVALID_USER</ActivateUsersResult>
    </ActivateUsersResponse>
  </soap:Body>
</soap:Envelope>
```


AssignMessage

Assign message to users.

Input:

Parameters	Type	Description
userID	string	Required. user unique ID.
msgID	string	Required. Message unique ID.
usernames	array of string	Required. Minimum one username value.

Output:

Parameters	Type	Description
AssignMessageResult	string	Result info: OK – success/completed MESSAGE_NOT_FOUND – invalid message ID supplied. INVALID_USER – invalid user ID supplied.

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:AssignMessage>
      <ser:userID>20011</ser:userID>
      <ser:msgID>208100442777</ser:msgID>
      <ser:usernames>
        <!--One or more repetitions:-->
        <ser:string>admin</ser:string>
        <ser:string>userA</ser:string>
      </ser:usernames>
    </ser:AssignMessage>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <AssignMessageResponse xmlns="http://faxcoreserver/services/">
      <AssignMessageResult>OK</AssignMessageResult>
    </AssignMessageResponse>
  </soap:Body>
</soap:Envelope>
```

AuthenticateUsers

Authenticate user to retrieve a valid ID to use with other operations.

Input:

Parameters	Type	Description
userName	string	Required. A valid username.
password	string	Required. A valid password.

Output:

Parameters	Type	Description
AuthenticateUserResult	long/bigint	Result info: 1 – invalid username/password

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:AuthenticateUser>
      <ser:userName>user_name</ser:userName>
      <ser:password>pass_word</ser:password>
    </ser:AuthenticateUser>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <AuthenticateUserResponse xmlns="http://faxcoreserver/services/">
      <AuthenticateUserResult>200011</AuthenticateUserResult>
    </AuthenticateUserResponse>
  </soap:Body>
</soap:Envelope>
```

CancelMessage

Cancel active transmission.

Input:

Parameters	Type	Description
msgID	string	Required. A valid message ID.

Output:

Parameters	Type	Description
CancelMessageResult	string	Result info: OK – success/completed CANCEL_NOT_ALLOW – message not allow to cancel or message already completed.

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:CancelMessage>
      <ser:msgID>208100442777</ser:msgID>
    </ser:CancelMessage>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CancelMessageResponse xmlns="http://faxcoreserver/services/">
      <CancelMessageResult>CANCEL_NOT_ALLOW</CancelMessageResult>
    </CancelMessageResponse>
  </soap:Body>
</soap:Envelope>
```

CreateInboundRoute

Create inbound routing rule.

Input:

Parameters	Type	Description
userID	long/bigint	Required. A valid user ID.
forwardedUserName	string	Required. A username to forward inbound message to.
condition1	string	Required. enum { 'None', 'BarCode', 'FaxPortNo', 'RoutingDigits', 'RemoteCallerID', 'RemoteCSID', 'ServerName' }
conditionExp1	string	Required. Condition expression/value based on the condition set above.
isAndOperator	boolean	Required. Set to true for 'AND', false for 'OR' Set either one if second condition is not required.
condition2	string	Required. enum { 'None', 'BarCode', 'FaxPortNo', 'RoutingDigits', 'RemoteCallerID', 'RemoteCSID', 'ServerName' } Set to 'None' if condition 2 is not required.
conditionExp2	string	Optional. Condition expression/value based on the condition 2 set above.
isActive	boolean	Set true to activate the rule.
priority	int	Set priority for the rule. Higher value, higher priority.

Output:

Parameters	Type	Description
CreateInboundRouteResult	string	Result info: Empty string – route create success

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:CreateInboundRoute>
      <ser:userID>2010211</ser:userID>
      <ser:forwardedUserName>admin</ser:forwardedUserName>
      <ser:condition1>BarCode</ser:condition1>
      <ser:conditionExp1>121</ser:conditionExp1>
```

```

    <ser:isAndOperator>0</ser:isAndOperator>
    <ser:condition2>None</ser:condition2>
    <ser:conditionExp2></ser:conditionExp2>
    <ser:isActive>false</ser:isActive>
    <ser:priority>9</ser:priority>
  </ser:CreateInboundRoute>
</soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CreateInboundRouteResponse xmlns="http://faxcoreserver/services/">
      <CreateInboundRouteResult/>
    </CreateInboundRouteResponse>
  </soap:Body>
</soap:Envelope>

```

CreateNewUser1

Create new user with default NOR/NOS = false.

Input:

Parameters	Type	Description
userID	long/bigint	Required. A valid admin user ID.
domainName	string	Required. Domain name the new user creates to.
userName	string	Required. new user's username.
password	string	Required. user user's password.
role	string	Required. new user's role. enum { 'SystemAdmin', 'DomainAdmin', 'SuperUser', 'StandardUser' }
isExternalAuth	boolean	Required. Set to 'true' if user require external authentication, 'false' if otherwise.
isActive	boolean	Required. Set account to active/inactive upon creation. 'true' = active.
displayName	string	Set user's display name.
firstName	string	Set user's first name.
middleName	String	Set user's middle name.
lastName	String	Set user's last name.
companyName	String	Set user's company name.
preferAddrType	string	Set user's preferred address type in user profile. enum { 'Fax', 'Email', 'FaxRaw' }
email	string	Set user's email address. This value

		is required when preferAddrType is set to Email.
fax	string	Set user's fax address. This value is required when preferAddrType is set to Fax.
rawFax	string	Set user's raw fax address. This value is required when preferAddrType is set to FaxRaw.
desc	string	Set user's profile description.

Output:

Parameters	Type	Description
CreateNewUser1Result	string	Result info: User ID – user create success

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:CreateNewUser1>
      <ser:userID>200011</ser:userID>
      <ser:domainName>Testdomain</ser:domainName>
      <ser:userName>testuser1</ser:userName>
      <ser:password>?testest</ser:password>
      <ser:role>StandardUser</ser:role>
      <ser:isExternalAuth>>false</ser:isExternalAuth>
      <ser:isActive>>true</ser:isActive>
      <ser:displayName>Test User</ser:displayName>
      <ser:firstName>Test</ser:firstName>
      <ser:middleName></ser:middleName>
      <ser:lastName>User</ser:lastName>
      <ser:companyName>Company A</ser:companyName>
      <ser:preferAddrType>Email</ser:preferAddrType>
      <ser:email>test@domain.com</ser:email>
      <ser:fax>123123123</ser:fax>
      <ser:rawFax></ser:rawFax>
      <ser:desc></ser:desc>
    </ser:CreateNewUser1>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CreateNewUser1Response xmlns="http://faxcoreserver/services/">
      <CreateNewUser1Result>10180032002801248</CreateNewUser1Result>
    </CreateNewUser1Response>
  </soap:Body>
</soap:Envelope>
```

CreateNewUser2

Create new user with NOR and NOS values.

Input:

Parameters	Type	Description
userID	long/bigint	Required. A valid admin user ID.
domainName	string	Required. Domain name the new user creates to.
userName	string	Required. new user's username.
password	string	Required. user user's password.
role	string	Required. new user's role. enum { 'SystemAdmin', 'DomainAdmin', 'SuperUser', 'StandardUser' }
isExternalAuth	boolean	Required. Set to 'true' if user require external authentication, 'false' if otherwise.
isActive	boolean	Required. Set account to active/inactive upon creation. 'true' = active.
displayName	string	Set user's display name.
firstName	string	Set user's first name.
middleName	string	Set user's middle name.
lastName	string	Set user's last name.
companyName	string	Set user's company name.
preferAddrType	string	Set user's preferred address type in user profile. enum { 'Fax', 'Email', 'FaxRaw' }
email	string	Set user's email address. This value is required when preferAddrType is set to Email.
fax	string	Set user's fax address. This value is required when preferAddrType is set to Fax.
rawFax	string	Set user's raw fax address. This value is required when preferAddrType is set to FaxRaw.
NOR	boolean	Set to receive notification on fax received.
NOS	boolean	Set to receive notification on fax sent.
desc	string	Set user's profile description.

Output:

Parameters	Type	Description
CreateNewUser2Result	string	Result info: User ID – user create success

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:CreateNewUser2>
      <ser:userID>200011</ser:userID>
      <ser:domainName>testdomain</ser:domainName>
      <ser:userName>test2</ser:userName>
      <ser:password>test</ser:password>
      <ser:role>DomainAdmin</ser:role>
      <ser:isExternalAuth>false</ser:isExternalAuth>
      <ser:isActive>true</ser:isActive>
      <ser:displayName>Test user</ser:displayName>
      <ser:firstName>Test</ser:firstName>
      <ser:middleName></ser:middleName>
      <ser:lastName>User</ser:lastName>
      <ser:companyName></ser:companyName>
      <ser:preferAddrType>Fax</ser:preferAddrType>
      <ser:email></ser:email>
      <ser:fax>12345678</ser:fax>
      <ser:rawFax></ser:rawFax>
      <ser:NOR>true</ser:NOR>
      <ser:NOS>false</ser:NOS>
      <ser:desc>This is a test user</ser:desc>
    </ser:CreateNewUser2>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CreateNewUser2Response xmlns="http://faxcoreserver/services/">
      <CreateNewUser2Result>10180032002801249</CreateNewUser2Result>
    </CreateNewUser2Response>
  </soap:Body>
</soap:Envelope>
```

CreateNewUser3

Create new user with NOR and NOS values, caller ID and CSID.

Input:

Parameters	Type	Description
userID	long/bigint	Required. A valid admin user ID.
domainName	string	Required. Domain name the new user creates to.
userName	string	Required.

		new user's username.
password	string	Required. user user's password.
role	string	Required. new user's role. enum { 'SystemAdmin', 'DomainAdmin', 'SuperUser', 'StandardUser' }
isExternalAuth	boolean	Required. Set to 'true' if user require external authentication, 'false' if otherwise.
isActive	boolean	Required. Set account to active/inactive upon creation. 'true' = active.
displayName	string	Set user's display name.
firstName	string	Set user's first name.
middleName	string	Set user's middle name.
lastName	string	Set user's last name.
companyName	string	Set user's company name.
preferAddrType	string	Set user's preferred address type in user profile. enum { 'Fax', 'Email', 'FaxRaw' }
email	string	Set user's email address. This value is required when preferAddrType is set to Email.
fax	string	Set user's fax address. This value is required when preferAddrType is set to Fax.
rawFax	string	Set user's raw fax address. This value is required when preferAddrType is set to FaxRaw.
NOR	boolean	Set to receive notification on fax received.
NOS	boolean	Set to receive notification on fax sent.
desc	string	Set user's profile description.
callerID	string	Set user's caller ID info.
csid	string	Set user's CSID value.

Output:

Parameters	Type	Description
CreateNewUser3Result	string	Result info: User ID – user create success

Sample XML request:

<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://faxcoreserver/services/"> <soapenv:Header/> <soapenv:Body> <ser:CreateNewUser3> <ser:userID>200011</ser:userID> <ser:domainName>testdomain</ser:domainName></pre>
--

```

    <ser:userName>test3</ser:userName>
    <ser:password>test</ser:password>
    <ser:role>StandardUser</ser:role>
    <ser:isExternalAuth>>false</ser:isExternalAuth>
    <ser:isActive>true</ser:isActive>
    <ser:displayName>Test 3</ser:displayName>
    <ser:firstName>John</ser:firstName>
    <ser:middleName></ser:middleName>
    <ser:lastName>Doe</ser:lastName>
    <ser:companyName>FFF</ser:companyName>
    <ser:preferAddrType>FaxRaw</ser:preferAddrType>
    <ser:email>test@ddd.com</ser:email>
    <ser:fax></ser:fax>
    <ser:rawFax>12334456</ser:rawFax>
    <ser:NOR>true</ser:NOR>
    <ser:NOS>true</ser:NOS>
    <ser:desc></ser:desc>
    <ser:callerID>2345</ser:callerID>
    <ser:csid>Company A</ser:csid>
  </ser:CreateNewUser3>
</soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CreateNewUser3Response xmlns="http://faxcoreserver/services/">
      <CreateNewUser3Result>10180032002801250</CreateNewUser3Result>
    </CreateNewUser3Response>
  </soap:Body>
</soap:Envelope>

```

DeactivateUser

Set user account to inactive.

Input:

Parameters	Type	Description
adminID	long/bigint	Required. A valid admin user ID.
userID	long/bigint	Required. User ID to deactivate/set inactive.

Output:

Parameters	Type	Description
DeactivateUserResult	string	Result info: OK – user set to inactive

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:DeactivateUser>
      <ser:adminID>200101</ser:adminID>
      <ser:userID>10180032002801250</ser:userID>
    </ser:DeactivateUser>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DeactivateUserResponse xmlns="http://faxcoreserver/services/">
      <DeactivateUserResult>OK</DeactivateUserResult>
    </DeactivateUserResponse>
  </soap:Body>
</soap:Envelope>
```

DeactivateUsers

Set a list of user accounts to inactive.

Input:

Parameters	Type	Description
adminID	long/bigint	Required. A valid admin user ID.
userIDList	array of long	Required. User ID to deactivate/set inactive. Minimum one is required.

Output:

Parameters	Type	Description
DeactivateUsersResult	string	Result info: OK – user set to inactive

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:DeactivateUsers>
      <ser:adminID>200101</ser:adminID>
      <ser:userIDList>
        <!--One or more repetitions:-->
        <ser:long>10180032002801250</ser:long>
        <ser:long>10180032002801249</ser:long>
      </ser:userIDList>
    </ser:DeactivateUsers>
  </soapenv:Body>
</soapenv:Envelope>
```

```

    </ser:userIDList>
  </ser:DeactivateUsers>
</soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DeactivateUsersResponse xmlns="http://faxcoreserver/services/">
      <DeactivateUsersResult>SOME_INVALID_USER</DeactivateUsersResult>
    </DeactivateUsersResponse>
  </soap:Body>
</soap:Envelope>

```

DelegateMessage

Send message on behalf of other user.

Input:

Parameters	Type	Description
delegateUserID	long/bigint	Required. A valid user ID.
userName	string	Required. Username of the message owner.
senderName	string	Required. sender's name.
senderCompName	string	Required. sender's company name.
subject	string	Required. message's subject
note	string	Required. message's note.
billingCode	string	Required. message's billing code info.
scheduleDate	string	Message's schedule date.
priority	int	Message's priority. 70 = high 60 = normal 50 = low
isOnHold	Boolean	Set message to on hold = true
mss	Boolean	Set 'true' to trigger notification when message sent success.
msf	boolean	Set "true" to trigger notification when message sent failed.
trackLabel	array of string	Set tracking info label.
trackValue	array of string	Set tracking info value.
recpName	array of string	Set a list of recipient names.
recpCompany	array of string	Set a list of recipient's company name.
recpAddress	array of string	Set a list of recipient address.
recpRawFax	array of boolean	Set a list of indication of recipient

		address is raw fax format.
recpNotifyAddr	array of string	Set a list of recipient's notification address.
recpIsRRFOn	array of boolean	Set a list of notify sender when recipient receive failed.
recpIsRRSON	array of boolean	Set a list of notify sender when recipient receive success.
documentName	array of string	Set a list of document names.
documentPath	array of string	Set a list of document path.
documentBytes	array of object in byte[]	Set a list of document binary data
documentIsMerge	array of boolean	Set a list of indication document is merge (cover page)

Output:

Parameters	Type	Description
DelegateMssageResult	string	Result info: Message ID – message create success

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:DelegateMessage>
      <ser:delegateUserID>201001</ser:delegateUserID>
      <ser:userName>testuser</ser:userName>
      <ser:senderName>John</ser:senderName>
      <ser:senderCompName>Food Inc</ser:senderCompName>
      <ser:subject>Send on behalf</ser:subject>
      <ser:note>test</ser:note>
      <ser:billingCode></ser:billingCode>
      <ser:scheduleDate></ser:scheduleDate>
      <ser:priority>60</ser:priority>
      <ser:isOnHold>false</ser:isOnHold>
      <ser:mss>true</ser:mss>
      <ser:msf>true</ser:msf>
      <ser:trackLabel>
        <!--Zero or more repetitions:-->
        <ser:string>Track Label 1</ser:string>
        <ser:string>Track Label 2</ser:string>
        <ser:string>Track Label 3</ser:string>
      </ser:trackLabel>
      <ser:trackValue>
        <!--Zero or more repetitions:-->
        <ser:string>Track Value 1</ser:string>
        <ser:string>Track Value 2</ser:string>
        <ser:string>Track Value 3</ser:string>
      </ser:trackValue>
      <ser:recpName>
        <!--One or more repetitions:-->
        <ser:string>Recipient A</ser:string>
      </ser:recpName>
    </ser:DelegateMessage>
  </soapenv:Body>
</soapenv:Envelope>
```

```

    <ser:string>Recipient B</ser:string>
  </ser:recpName>
  <ser:recpCompName>
    <!--One or more repetitions:-->
    <ser:string>Recipient A Company</ser:string>
    <ser:string>Recipient B Company</ser:string>
  </ser:recpCompName>
  <ser:recpAddress>
    <!--One or more repetitions:-->
    <ser:string>123456789</ser:string>
    <ser:string>123456789</ser:string>
  </ser:recpAddress>
  <ser:recpRawFax>
    <!--One or more repetitions:-->
    <ser:boolean>true</ser:boolean>
    <ser:boolean>false</ser:boolean>
  </ser:recpRawFax>
  <ser:recpNotifyAddr>
    <!--One or more repetitions:-->
    <ser:string></ser:string>
    <ser:string></ser:string>
  </ser:recpNotifyAddr>
  <ser:recpIsRRFOn>
    <!--One or more repetitions:-->
    <ser:boolean>false</ser:boolean>
    <ser:boolean>false</ser:boolean>
  </ser:recpIsRRFOn>
  <ser:recpIsRRSON>
    <!--One or more repetitions:-->
    <ser:boolean>false</ser:boolean>
    <ser:boolean>false</ser:boolean>
  </ser:recpIsRRSON>
  <ser:documentName>
    <!--One or more repetitions:-->
    <ser:string>Test.pdf</ser:string>
  </ser:documentName>
  <ser:documentPath>
    <!--One or more repetitions:-->
    <ser:string></ser:string>
  </ser:documentPath>
  <ser:documentBytes>
    <!--One or more repetitions:-->
    <ser:anyType>[file binary]</ser:anyType>
  </ser:documentBytes>
  <ser:documentIsMerge>
    <!--One or more repetitions:-->
    <ser:boolean>false</ser:boolean>
  </ser:documentIsMerge>
</ser:DelegateMessage>
</soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DelegateMessageResponse xmlns="http://faxcoreserver/services/">
      <DelegateMessageResult>INVALID_DELEGATOR</DelegateMessageResult>
    </DelegateMessageResponse>
  </soap:Body>
</soap:Envelope>

```

DeleteMessage

Delete existing message.

Input:

Parameters	Type	Description
msgID	string	Required. A valid message ID.
userID	long/bigint	Required. A valid user ID.

Output:

Parameters	Type	Description
DeleteMessageResult	string	Result info: OK – delete message success

Sample XML request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser>DeleteMessage>
      <ser:msgID>10010001000</ser:msgID>
      <ser:userID>201001</ser:userID>
    </ser>DeleteMessage>
  </soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DeleteMessageResponse xmlns="http://faxcoreserver/services/">
      <DeleteMessageResult>MESSAGE_NOT_FOUND</DeleteMessageResult>
    </DeleteMessageResponse>
  </soap:Body>
</soap:Envelope>

```

DisabledInboudRoute

Disable inbound routing rule.

Input:

Parameters	Type	Description
adminID	long/bigint	Required. A valid admin ID.
routeID	Int	Required. A valid route ID.

Output:

Parameters	Type	Description
DisabledInboudRouteResult	string	Result info: Empty value – disable route success

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:DisabledInboudRoute>
      <ser:adminID>20101</ser:adminID>
      <ser:routeID>004</ser:routeID>
    </ser:DisabledInboudRoute>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DisabledInboudRouteResponse xmlns="http://faxcoreserver/services/">
      <DisabledInboudRouteResult/>
    </DisabledInboudRouteResponse>
  </soap:Body>
</soap:Envelope>
```

DownloadMessageByte1

Download message binary with specific message ID, delivery number, download type and start page index.

Input:

Parameters	Type	Description
messageID	string	Required. A valid message ID.
deliveryNum	int	Required. A delivery number/transaction sequence in a message. For inbound message,

		value always be '1'.
downloadType	string	Required. "TIF" or "PDF"
startPageIndex	int	Required. Start download from a specific page. Zero base index.

Output:

Parameters	Type	Description
DownloadMessageByte1Result	Base64Binary	Result info: Return binary value

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:DownloadMessageByte1>
      <ser:messageID>208100456346</ser:messageID>
      <ser:deliveryNum>1</ser:deliveryNum>
      <ser:downloadType>PDF</ser:downloadType>
      <ser:startPageIndex>0</ser:startPageIndex>
    </ser:DownloadMessageByte1>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DownloadMessageByte1Response xmlns="http://faxcoreserver/services/">
      <DownloadMessageByte1Result/>
      <output>[binary data]</output>
    </DownloadMessageByte1Response>
  </soap:Body>
</soap:Envelope>
```

DownloadMessageByte2

Download message binary with specific message ID, delivery number, download type and start page index and list of excluded pages.

Input:

Parameters	Type	Description
messageID	string	Required. A valid message ID.
deliveryNum	int	Required. A delivery number/transaction sequence in a message. For inbound message, value always be '1'.

downloadType	string	Required. "TIF" or "PDF"
startPageIndex	int	Required. Start download from a specific page. Zero base index.
excludePageList	array of int	Required. List of exclusion pages. Non-zero base index.

Output:

Parameters	Type	Description
DownloadMessageByte2Result	Base64Binary	Result info: Return binary value

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:DownloadMessageInByte2>
      <ser:messageID>208100456346</ser:messageID>
      <ser:deliveryNum>1</ser:deliveryNum>
      <ser:downloadType>TIF</ser:downloadType>
      <ser:startPageIndex>0</ser:startPageIndex>
      <ser:excludePageList>
        <!--One or more repetitions:-->
        <ser:int>1</ser:int>
      </ser:excludePageList>
    </ser:DownloadMessageInByte2>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DownloadMessageInByte2Response xmlns="http://faxcoreserver/services/">
      <DownloadMessageInByte2Result/>
      <output>[binary data]</output>
    </DownloadMessageInByte2Response>
  </soap:Body>
</soap:Envelope>
```

DownloadMessageByte3

Download message binary with specific delivery ID, download type and start page index and list of excluded pages.

Input:

Parameters	Type	Description
------------	------	-------------

deliveryID	string	Required. A delivery ID/transaction ID in a message.
downloadType	string	Required. "TIF" or "PDF"
startPageIndex	int	Required. Start download from a specific page. Zero base index.

Output:

Parameters	Type	Description
DownloadMessageByte3Result	Base64Binary	Result info: Return binary value

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:DownloadMessageInByte3>
      <ser:deliveryID>208100456346</ser:deliveryID>
      <ser:downloadType>TIF</ser:downloadType>
      <ser:startPageIndex>0</ser:startPageIndex>
    </ser:DownloadMessageInByte3>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DownloadMessageInByte3Response xmlns="http://faxcoreserver/services/">
      <DownloadMessageInByte3Result>
        <output>[binary data]</output>
      </DownloadMessageInByte3Result>
    </DownloadMessageInByte3Response>
  </soap:Body>
</soap:Envelope>
```

DownloadMessageByte4

Download message binary with specific delivery ID, download type and start page index and list of excluded pages.

Input:

Parameters	Type	Description
deliveryID	string	Required. A delivery ID/transaction ID in a message.
downloadType	string	Required. "TIF" or "PDF"
startPageIndex	int	Required. Start download from a

		specific page. Zero base index.
excludePageList	array of int	Required. List of exclusion pages. Non-zero base index.

Output:

Parameters	Type	Description
DownloadMessageByte4Result	Base64Binary	Result info: Return binary value

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:DownloadMessageInByte4>
      <ser:deliveryID>208100456346</ser:deliveryID>
      <ser:downloadType>TIF</ser:downloadType>
      <ser:startPageIndex>0</ser:startPageIndex>
      <ser:excludePageList>
        <!--One or more repetitions:-->
        <ser:int>2</ser:int>
      </ser:excludePageList>
    </ser:DownloadMessageInByte4>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DownloadMessageInByte4Response xmlns="http://faxcoreserver/services/">
      <DownloadMessageInByte4Result><output>[binary data]</output></DownloadMessageInByte4Result>
    </DownloadMessageInByte4Response>
  </soap:Body>
</soap:Envelope>
```

ForwardToUser

Forward message to users.

Input:

Parameters	Type	Description
userID	string	Required. A valid user ID.
msgID	String	Required. A message ID of the message you want to forward.
usernames	array of String	Required. Intended usernames as recipient.

Output:

Parameters	Type	Description
ForwardToUserResult	string	Result info: OK – forward success NOT_MESSAGE_OWNER – user ID is not the owner of the message ID.

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:ForwardToUser>
      <ser:userID>201010</ser:userID>
      <ser:msgID>208100456346</ser:msgID>
      <ser:username>
        <!--One or more repetitions:-->
        <ser:string>testuser</ser:string>
      </ser:username>
    </ser:ForwardToUser>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ForwardToUserResponse xmlns="http://faxcoreserver/services/">
      <ForwardToUserResult>NOT_MESSAGE_OWNER</ForwardToUserResult>
    </ForwardToUserResponse>
  </soap:Body>
</soap:Envelope>
```

GetDomainList

Retrieve domains belong to domain admin.

Input:

Parameters	Type	Description
adminID	long/bigint	Required. A valid admin user ID.

Output:

Parameters	Type	Description
------------	------	-------------

GetDomainListResult	array of string	Result info: Return array of string. {domainID domainName}
---------------------	-----------------	---

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetDomainList>
      <ser:adminID>201010</ser:adminID>
    </ser:GetDomainList>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetDomainListResponse xmlns="http://faxcoreserver/services/">
      <GetDomainListResult>
        <string>10||SYSTEM</string>
        <string>11110||TESTDOMAIN</string>
        <string>10600||Department 1</string>
      </GetDomainListResult>
    </GetDomainListResponse>
  </soap:Body>
</soap:Envelope>
```

GetMessageDetails1

Gets message detail in XML string format.

Input:

Parameters	Type	Description
msgID	string	Required. A valid message ID.

Output:

Parameters	Type	Description
GetMessageDetails1Result	string	Result info: Return XML string

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
```

```
<ser:GetMessageDetails1>
  <ser:msgID>208100456346</ser:msgID>
</ser:GetMessageDetails1>
</soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetMessageDetails1Response xmlns="http://faxcoreserver/services/">
      <GetMessageDetails1Result><![CDATA[<?xml version="1.0" encoding="utf-8"?>
<MessageDetails xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <MessageID>208100456346</MessageID>
  <MessageNo>50618</MessageNo>
  <OwnerName>SYS-RECEIVE</OwnerName>
  <OwnerDomain>SYSTEM</OwnerDomain>
  <MessageType>Inbound</MessageType>
  <Subject>Inbound Message</Subject>
  <Tracking />
  <TotalDocuments />
  <TotalTransactions>1</TotalTransactions>
  <TotalComplete>1</TotalComplete>
  <ProcessCount>0</ProcessCount>
  <CreateDate>12/5/2014 4:49:29 PM</CreateDate>
  <ScheduleDate>12/5/2014 4:49:29 PM</ScheduleDate>
  <CompleteDate>12/5/2014 4:49:29 PM</CompleteDate>
  <ApprovalRequired />
  <ApprovalUsers />
  <Barcodes />
  <Transactions>
    <TransactionDetails>
      <TransactionID>3847983590146</TransactionID>
      <DocumentID>3268000932459753</DocumentID>
      <AddressTypeID>2</AddressTypeID>
      <AddressTypeDef>FAX</AddressTypeDef>
      <TransactionSeq>1</TransactionSeq>
      <TransactionTypeID>301</TransactionTypeID>
      <TransactionTypeDef>To</TransactionTypeDef>
      <TransmitErrorID>0</TransmitErrorID>
      <TransmitErrorInfo />
      <RecipientName>SYS-RECEIVE</RecipientName>
      <RecipientCompanyName />
      <MessageDisplayAddress>FAXCORE</MessageDisplayAddress>
      <NumberOfAttempts>1</NumberOfAttempts>
      <PageCount>1</PageCount>
      <PageSent>1</PageSent>
      <TransactionStatusDef>Completed</TransactionStatusDef>
      <LocalCSID />
      <RemoteCSID>ROBOFAX</RemoteCSID>
      <TransmitSpeed>33600</TransmitSpeed>
      <TotalTransmitTime>17 sec</TotalTransmitTime>
      <CreateDate>12/5/2014 4:49:29 PM</CreateDate>
```

```

<CompleteDate>12/5/2014 4:49:30 PM</CompleteDate>
<Attempts>
  <TransmissionDetails>
    <TransmissionID>201800000000331470</TransmissionID>
    <CreateDate>12/5/2014 4:49:29 PM</CreateDate>
    <ScheduleDate>12/5/2014 4:49:29 PM</ScheduleDate>
    <CompleteDate>12/5/2014 4:49:29 PM</CompleteDate>
    <CallingNumber>30327268800</CallingNumber>
    <CalledNumber>30327268804</CalledNumber>
    <RemoteCSID>ROBOFAX</RemoteCSID>
    <RoutingInfo>30327268804</RoutingInfo>
    <FaxProtocol>V34</FaxProtocol>
    <TransmitSpeed>33600</TransmitSpeed>
    <CompressionMode>MH</CompressionMode>
    <CallDuration>17</CallDuration>
    <TransmitDuration>17</TransmitDuration>
    <FaxResolution />
    <TransmitStatus>0</TransmitStatus>
    <TransmitStatusDef>Completed</TransmitStatusDef>
    <AttemptNo>1</AttemptNo>
    <MaxAttemptNo>1</MaxAttemptNo>
    <PageCount>1</PageCount>
    <StartPageNo>1</StartPageNo>
    <PageSent>1</PageSent>
    <FaxAgentID>101009</FaxAgentID>
    <PortNo>0</PortNo>
    <BoardID>M0,0</BoardID>
    <ServerIP>10.15.7.6</ServerIP>
    <ErrorID>0</ErrorID>
    <ErrorInfo />
  </TransmissionDetails>
</Attempts>
</TransactionDetails>
</Transactions>
<Status>Completed</Status>
<BillingCode />
</MessageDetails>]]></GetMessageDetails1Result>
  </GetMessageDetails1Response>
</soap:Body>
</soap:Envelope>

```

GetMessageDetails2

Get message detail in XML string format, include read status. Require user ID

Input:

Parameters	Type	Description
msgID	string	Required. A valid message ID.
userID	long/bigint	Required. A valid user ID which own the message.

Output:

Parameters	Type	Description
GetMessageDetails2Result	string	Result info: Return XML string

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetMessageDetails2>
      <ser:msgID>208100456346</ser:msgID>
      <ser:userID>201011</ser:userID>
    </ser:GetMessageDetails2>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetMessageDetails2Response xmlns="http://faxcoreserver/services/">
      <GetMessageDetails2Result><![CDATA[<?xml version="1.0" encoding="utf-8"?>
<MessageDetails2 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <MessageID>2081004427777</MessageID>
  <MessageNo>48596</MessageNo>
  <OwnerName>SYS-RECEIVE</OwnerName>
  <OwnerDomain>SYSTEM</OwnerDomain>
  <MessageType>Inbound</MessageType>
  <Subject>Inbound Message</Subject>
  <Tracking />
  <TotalDocuments />
  <TotalTransactions>1</TotalTransactions>
  <TotalComplete>1</TotalComplete>
  <ProcessCount>0</ProcessCount>
  <CreateDate>11/13/2014 10:03:20 AM</CreateDate>
  <ScheduleDate>11/13/2014 10:03:20 AM</ScheduleDate>
  <CompleteDate>11/13/2014 10:03:20 AM</CompleteDate>
  <ApprovalRequired />
  <ApprovalUsers />
  <Barcodes />
  <Transactions>
    <TransactionDetails>
      <TransactionID>3847983569763</TransactionID>
      <DocumentID>3268000932453844</DocumentID>
      <AddressTypeID>2</AddressTypeID>
      <AddressTypeDef>FAX</AddressTypeDef>
      <TransactionSeq>1</TransactionSeq>
      <TransactionTypeID>301</TransactionTypeID>
      <TransactionTypeDef>To</TransactionTypeDef>
      <TransmitErrorID>0</TransmitErrorID>
      <TransmitErrorInfo />
    </TransactionDetails>
  </Transactions>
</MessageDetails2>
</GetMessageDetails2Result>
</GetMessageDetails2Response>
</soap:Body>
</soap:Envelope>
```

```
<RecipientName>SYS-RECEIVE</RecipientName>
<RecipientCompanyName />
<MessageDisplayAddress>FAXCORE</MessageDisplayAddress>
<NumberOfAttempts>1</NumberOfAttempts>
<PageCount>1</PageCount>
<PageSent>1</PageSent>
<TransactionStatusDef>Completed</TransactionStatusDef>
<LocalCSID />
<RemoteCSID>60333429136</RemoteCSID>
<TransmitSpeed>9600</TransmitSpeed>
<TotalTransmitTime>31 sec</TotalTransmitTime>
<CreatedDate>11/13/2014 10:03:20 AM</CreatedDate>
<CompleteDate>11/13/2014 10:03:20 AM</CompleteDate>
<Attempts>
  <TransmissionDetails>
    <TransmissionID>201800000000320863</TransmissionID>
    <CreatedDate>11/13/2014 10:03:20 AM</CreatedDate>
    <ScheduleDate>11/13/2014 10:03:20 AM</ScheduleDate>
    <CompleteDate>11/13/2014 10:03:20 AM</CompleteDate>
    <CallingNumber>0333429136</CallingNumber>
    <CalledNumber>0327268913</CalledNumber>
    <RemoteCSID>60333429136</RemoteCSID>
    <RoutingInfo>0327268913</RoutingInfo>
    <FaxProtocol>V29</FaxProtocol>
    <TransmitSpeed>9600</TransmitSpeed>
    <CompressionMode>MH</CompressionMode>
    <CallDuration>31</CallDuration>
    <TransmitDuration>31</TransmitDuration>
    <FaxResolution />
    <TransmitStatus>0</TransmitStatus>
    <TransmitStatusDef>Completed</TransmitStatusDef>
    <AttemptNo>1</AttemptNo>
    <MaxAttemptNo>1</MaxAttemptNo>
    <PageCount>1</PageCount>
    <StartPageNo>1</StartPageNo>
    <PageSent>1</PageSent>
    <FaxAgentID>100009</FaxAgentID>
    <PortNo>0</PortNo>
    <BoardID>M0,0</BoardID>
    <ServerIP>10.165.7.36</ServerIP>
    <ErrorID>0</ErrorID>
    <ErrorInfo />
  </TransmissionDetails>
</Attempts>
</TransactionDetails>
</Transactions>
<Status>Completed</Status>
<BillingCode />
<IsRead>False</IsRead>
</MessageDetails2>]]></GetMessageDetails2Result>
  </GetMessageDetails2Response>
</soap:Body>
</soap:Envelope>
```

GetMessageList3

Get message list based on the folder name and isread/unread flag.

Input:

Parameters	Type	Description
userID	long/bigint	Required. A valid user ID.
folderName	string	Required. FaxCore folder name; Inbox/Outbox/Sent, etc
isRead	boolean	Required. Filter read items. Set 'true' to return read messages.
sortDescending	boolean	Required. Set 'true' to return result in descending order.
maxResults	int	Cap the maximum item returned.

Output:

Parameters	Type	Description
GetMessageList3Result	array of string	Result info: Return array of string

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetMessageList3>
      <ser:userID>201011</ser:userID>
      <ser:folderName>inbox</ser:folderName>
      <ser:isRead>false</ser:isRead>
      <ser:sortDescending>true</ser:sortDescending>
      <ser:maxResults>3</ser:maxResults>
    </ser:GetMessageList3>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetMessageList3Response xmlns="http://faxcoreserver/services/">
      <GetMessageList3Result>
        <string>2108100455920</string>
        <string>2108100455778</string>
        <string>2108100455275</string>
      </GetMessageList3Result>
    </GetMessageList3Response>
  </soap:Body>
</soap:Envelope>
```

GetMessageList2

Get message list based on the folder name, date range based on the message create date and isread/unread flag.

Input:

Parameters	Type	Description
userID	long/bigint	Required. A valid user ID.
folderName	string	Required. FaxCore folder name; Inbox/Outbox/Sent, etc
isRead	boolean	Required. Filter read items. Set 'true' to return read messages.
startDate	string	Required. Begin date.
endDate	string	Required. End date.
sortDescending	boolean	Required. Set 'true' to return result in descending order.
maxResults	int	Cap the maximum item returned.

Output:

Parameters	Type	Description
GetMessageList2Result	Array of string	Result info: Return array of string

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetMessageList2>
      <ser:userID>2010211</ser:userID>
      <ser:folderName>inbox</ser:folderName>
      <ser:isRead>true</ser:isRead>
      <ser:startDate>20141010</ser:startDate>
      <ser:endDate>20141031</ser:endDate>
      <ser:sortDescending>false</ser:sortDescending>
      <ser:maxResults>5</ser:maxResults>
    </ser:GetMessageList2>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetMessageList2Response xmlns="http://faxcoreserver/services/">
      <GetMessageList2Result>
        <string>208100439484</string>
        <string>208100440351</string>
        <string>208100441484</string>
      </GetMessageList2Result>
    </GetMessageList2Response>
  </soap:Body>
</soap:Envelope>
```

```
</GetMessageList2Result>
</GetMessageList2Response>
</soap:Body>
</soap:Envelope>
```

GetMessageList1

Get message list based on the folder name and message create date range.

Input:

Parameters	Type	Description
userID	long/bigint	Required. A valid user ID.
folderName	string	Required. FaxCore folder name; Inbox/Outbox/Sent, etc
startDate	string	Required. Begin date.
endDate	string	Required. End date.
sortDescending	boolean	Required. Set 'true' to return result in descending order.
maxResults	int	Cap the maximum item returned.

Output:

Parameters	Type	Description
GetMessageList1Result	Array of string	Result info: Return array of string

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetMessageList1>
      <ser:userID>2010211</ser:userID>
      <ser:folderName>sent</ser:folderName>
      <ser:startDate>20141010</ser:startDate>
      <ser:endDate>20141031</ser:endDate>
      <ser:sortDescending>false</ser:sortDescending>
      <ser:maxResults>5</ser:maxResults>
    </ser:GetMessageList1>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetMessageList1Response xmlns="http://faxcoreserver/services/">
      <GetMessageList1Result>
        <string>2108100438971</string>
        <string>2108100438972</string>
      </GetMessageList1Result>
    </GetMessageList1Response>
  </soap:Body>
</soap:Envelope>
```

```

        <string>2108100438973</string>
        <string>2108100441281</string>
    </GetMessageList1Result>
</GetMessageList1Response>
</soap:Body>
</soap:Envelope>

```

GetMessageList8

Get message list based on the folder name.

Input:

Parameters	Type	Description
userID	long/bigint	Required. A valid user ID.
folderName	string	Required. FaxCore folder name; Inbox/Outbox/Sent, etc
sortDescending	boolean	Required. Set 'true' to return result in descending order.
maxResults	int	Cap the maximum item returned.

Output:

Parameters	Type	Description
GetMessageList8Result	Array of string	Result info: Return array of string

Sample XML request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetMessageList8>
      <ser:userID>2010211</ser:userID>
      <ser:folderName>sent</ser:folderName>
      <ser:sortDescending>true</ser:sortDescending>
      <ser:maxResults>5</ser:maxResults>
    </ser:GetMessageList8>
  </soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetMessageList8Response xmlns="http://faxcoreserver/services/">
      <GetMessageList8Result>
        <string>2108100454275</string>
        <string>2108100444130</string>
        <string>2108100444129</string>
        <string>2108100443024</string>
      </GetMessageList8Result>
    </GetMessageList8Response>
  </soap:Body>
</soap:Envelope>

```

```

    <string>2108100443011</string>
  </GetMessageList8Result>
</GetMessageList8Response>
</soap:Body>
</soap:Envelope>

```

GetMessageList4

Get message list based on the folder name, users and whether fax is downloaded.

Input:

Parameters	Type	Description
userList	array of long/bigint	Required. A valid user ID.
folderName	string	Required. FaxCore folder name; Inbox/Outbox/Sent, etc
isDownloaded	boolean	Required. Set 'true' to return list of downloaded messages.
sortDescending	boolean	Required. Set 'true' to return result in descending order.
maxResults	bnt	Cap the maximum item returned.

Output:

Parameters	Type	Description
GetMessageList4Result	Array of string	Result info: Return array of string

Sample XML request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetMessageList4>
      <ser:userList>
        <!--One or more repetitions:-->
        <ser:long>2010211</ser:long>
        <ser:long>202022</ser:long>
      </ser:userList>
      <ser:folderName>inbox</ser:folderName>
      <ser:isDownloaded>>false</ser:isDownloaded>
      <ser:sortDescending>true</ser:sortDescending>
      <ser:maxResults>5</ser:maxResults>
    </ser:GetMessageList4>
  </soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>

```

```

<GetMessageList4Response xmlns="http://faxcoreserver/services/">
  <GetMessageList4Result>
    <string>2108100455920</string>
    <string>2108100455778</string>
    <string>2108100455469</string>
    <string>2108100455276</string>
    <string>2108100455275</string>
  </GetMessageList4Result>
</GetMessageList4Response>
</soap:Body>
</soap:Envelope>

```

GetMessageList5

Get message list based on the folder name, users, whether fax is downloaded, is read, and message create date range.

Input:

Parameters	Type	Description
userList	array of long/bigint	Required. A valid user ID.
folderName	string	Required. FaxCore folder name; Inbox/Outbox/Sent, etc
isDownloaded	boolean	Required. Set 'true' to return list of downloaded messages.
isRead	boolean	Required. Filter read items. Set 'true' to return read item list.
startDate	string	Required. Begin date.
endDate	string	Required. End date.
sortDescending	boolean	Required. Set 'true' to return result in descending order.
maxResults	int	Cap the maximum item returned.

Output:

Parameters	Type	Description
GetMessageList5Result	Array of string	Result info: Return array of string

Sample XML request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetMessageList5>
      <ser:userList>
        <!--One or more repetitions:-->
        <ser:long>2010211</ser:long>
        <ser:long>2002</ser:long>
      </ser:userList>
      <ser:folderName>sent</ser:folderName>
      <ser:isDownloaded>>false</ser:isDownloaded>
    </ser:GetMessageList5>
  </soapenv:Body>
</soapenv:Envelope>

```



```

    <ser:isRead>false</ser:isRead>
    <ser:startDate>20141010</ser:startDate>
    <ser:endDate>20141031</ser:endDate>
    <ser:sortDescending>true</ser:sortDescending>
    <ser:maxResults>5</ser:maxResults>
  </ser:GetMessageList5>
</soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetMessageList5Response xmlns="http://faxcoreserver/services/">
      <GetMessageList5Result>
        <string>2108100441281</string>
        <string>2108100438973</string>
        <string>2108100438971</string>
      </GetMessageList5Result>
    </GetMessageList5Response>
  </soap:Body>
</soap:Envelope>

```

GetMessageList6

Get message list based on the folder name, users, whether fax is downloaded, and message create date range.

Input:

Parameters	Type	Description
userList	array of long/bigint	Required. A valid user ID.
folderName	string	Required. FaxCore folder name; Inbox/Outbox/Sent, etc
isDownloaded	boolean	Required. Set 'true' to return list of downloaded messages.
startDate	string	Required. Begin date.
endDate	string	Required. End date.
sortDescending	boolean	Required. Set 'true' to return result in descending order.
maxResults	int	Cap the maximum item returned.

Output:

Parameters	Type	Description
GetMessageList6Result	Array of string	Result info: Return array of string

Sample XML request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"

```

```

xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetMessageList6>
      <ser:userList>
        <!--One or more repetitions:-->
        <ser:long>2010211</ser:long>
        <ser:long>2002</ser:long>
      </ser:userList>
      <ser:folderName>inbox</ser:folderName>
      <ser:isDownloaded>>false</ser:isDownloaded>
      <ser:startDate>20141010</ser:startDate>
      <ser:endDate>20141031</ser:endDate>
      <ser:sortDescending>true</ser:sortDescending>
      <ser:maxResults>5</ser:maxResults>
    </ser:GetMessageList6>
  </soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetMessageList6Response xmlns="http://faxcoreserver/services/">
      <GetMessageList6Result>
        <string>2108100441484</string>
        <string>2108100441482</string>
        <string>2108100441479</string>
        <string>2108100441376</string>
        <string>2108100441348</string>
      </GetMessageList6Result>
    </GetMessageList6Response>
  </soap:Body>
</soap:Envelope>

```

GetMessageList7

Get message list based on the folder name, users, whether fax is downloaded, and is read.

Input:

Parameters	Type	Description
userList	array of long/bigint	Required. A valid user ID.
folderName	string	Required. FaxCore folder name; Inbox/Outbox/Sent, etc
isDownloaded	boolean	Required. Set 'true' to return list of downloaded messages.
isRead	boolean	Required. Filter read items. Set 'true' to return read item list.
sortDescending	boolean	Required. Set 'true' to return result in descending order.
maxResults	int	Cap the maximum item returned.

Output:

Parameters	Type	Description
GetMessageList7Result	Array of string	Result info: Return array of string

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetMessageList7>
      <ser:userList>
        <!--One or more repetitions:-->
        <ser:long>2010211</ser:long>
        <ser:long>2002</ser:long>
      </ser:userList>
      <ser:folderName>inbox</ser:folderName>
      <ser:isDownloaded>>false</ser:isDownloaded>
      <ser:isRead>>false</ser:isRead>
      <ser:sortDescending>>true</ser:sortDescending>
      <ser:maxResults>5</ser:maxResults>
    </ser:GetMessageList7>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetMessageList7Response xmlns="http://faxcoreserver/services/">
      <GetMessageList7Result>
        <string>2108100455920</string>
        <string>2108100455778</string>
        <string>2108100455275</string>
        <string>2108100455270</string>
        <string>2108100455097</string>
      </GetMessageList7Result>
    </GetMessageList7Response>
  </soap:Body>
</soap:Envelope>
```

GetMessageStatus

Get message's status

Input:

Parameters	Type	Description
msgID	string	Required. A valid message ID.

Output:

Parameters	Type	Description
GetMessageStatusResult	string	Result info: Return string

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetMessageStatus>
      <ser:msgID>2108100455920</ser:msgID>
    </ser:GetMessageStatus>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetMessageStatusResponse xmlns="http://faxcoreserver/services/">
      <GetMessageStatusResult>Completed</GetMessageStatusResult>
    </GetMessageStatusResponse>
  </soap:Body>
</soap:Envelope>
```

GetMsgFolders

Retrieve folder list belong to user

Input:

Parameters	Type	Description
userID	long/bigint	Required. A valid user ID.

Output:

Parameters	Type	Description
GetMsgFoldersResult	Array of string	Result info: Return array string

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
```

```
<ser:GetMsgFolders>
  <ser:userID>2010211</ser:userID>
</ser:GetMsgFolders>
</soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetMsgFoldersResponse xmlns="http://faxcoreserver/services/">
      <GetMsgFoldersResult>
        <string>inbox</string>
        <string>outbox</string>
        <string>sent</string>
        <string>cancelled</string>
        <string>hold</string>
        <string>trash</string>
        <string>unrouted</string>
        <string>Processed</string>
      </GetMsgFoldersResult>
    </GetMsgFoldersResponse>
  </soap:Body>
</soap:Envelope>
```

GetTrackRecords

Return the tracking info for a message.

Input:

Parameters	Type	Description
msgID	long/bigint	Required. A valid message ID.

Output:

Parameters	Type	Description
GetTrackRecordsResult	Array of string	Result info: Return array of string. {trackID trackLabel trackValue}

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetTrackRecords>
      <ser:msgID>218100454275</ser:msgID>
    </ser:GetTrackRecords>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetTrackRecordsResponse xmlns="http://faxcoreserver/services/">
      <GetTrackRecordsResult>
        <string>254680|Tracking No.|1</string>
      </GetTrackRecordsResult>
    </GetTrackRecordsResponse>
  </soap:Body>
</soap:Envelope>
```

GetUserDetailsByUserID

Get user's details

Input:

Parameters	Type	Description
adminID	long/bigint	Required. A valid admin user ID.
userID	long/bigint	Required. A valid user ID

Output:

Parameters	Type	Description
GetUserDetailsByUserIDResult	string	Result info: Return string

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetUserDetailsByUserID>
      <ser:adminID>2010211</ser:adminID>
      <ser:userID>20320221</ser:userID>
    </ser:GetUserDetailsByUserID>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetUserDetailsByUserIDResponse xmlns="http://faxcoreserver/services/">
      <GetUserDetailsByUserIDResult><![CDATA[<?xml version="1.0" encoding="utf-8"?>
<UserDetails xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```

<UserID>2010211</UserID>
<UserName>ADNAN</UserName>
<DisplayName>Adnan Yaya</DisplayName>
<FirstName>Adnan</FirstName>
<MiddleName />
<LastName>Yaya</LastName>
<PreferAddressType />
<Addresses>
  <UserAddress>
    <AddressID>10</AddressID>
    <AddressType>EMAIL</AddressType>
    <Address>info@fax.com.my</Address>
    <IsPrimary>true</IsPrimary>
    <NotifyOnReceive>true</NotifyOnReceive>
    <NotifyOnSend>true</NotifyOnSend>
  </UserAddress>
  <UserAddress>
    <AddressID>884</AddressID>
    <AddressType>EMAIL</AddressType>
    <Address>jan@faxcore.com</Address>
    <IsPrimary>false</IsPrimary>
    <NotifyOnReceive>true</NotifyOnReceive>
    <NotifyOnSend>false</NotifyOnSend>
  </UserAddress>
  <UserAddress>
    <AddressID>993</AddressID>
    <AddressType>EMAIL</AddressType>
    <Address>dong@faxcore.com</Address>
    <IsPrimary>false</IsPrimary>
    <NotifyOnReceive>false</NotifyOnReceive>
    <NotifyOnSend>false</NotifyOnSend>
  </UserAddress>
</Addresses>
<Role>DOMAIN ADMIN</Role>
<AuthenticationType>Internal</AuthenticationType>
<IsActive>true</IsActive>
<Description />
</UserDetails>]]></GetUserDetailsByUserIDResult>
  </GetUserDetailsByUserIDResponse>
</soap:Body>
</soap:Envelope>

```

GetUserDetailsByUserName

Get user's details

Input:

Parameters	Type	Description
adminID	long/bigint	Required. A valid admin user ID.
username	string	Required. A valid username.

Output:

Parameters	Type	Description
GetUserDetailsByUserNameResult	string	Result info: Return string

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetUserDetailsByUserName>
      <ser:adminID>2010211</ser:adminID>
      <ser:userName>testuser</ser:userName>
    </ser:GetUserDetailsByUserName>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetUserDetailsByUserNameResponse xmlns="http://faxcoreserver/services/">
      <GetUserDetailsByUserNameResult><![CDATA[<?xml version="1.0" encoding="utf-8"?>
<UserDetails xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <UserID>21121224</UserID>
  <UserName>TESTUSER</UserName>
  <DisplayName>Test User</DisplayName>
  <FirstName>Test</FirstName>
  <MiddleName />
  <LastName>User</LastName>
  <PreferAddressType />
  <Addresses>
    <UserAddress>
      <AddressID>8</AddressID>
      <AddressType>EMAIL</AddressType>
      <Address>testuser@faxcore.com</Address>
      <IsPrimary>true</IsPrimary>
      <NotifyOnReceive>false</NotifyOnReceive>
      <NotifyOnSend>false</NotifyOnSend>
    </UserAddress>
  </Addresses>
  <Role>DOMAIN ADMIN</Role>
  <AuthenticationType>Internal</AuthenticationType>
  <IsActive>true</IsActive>
  <Description />
</UserDetails>]]></GetUserDetailsByUserNameResult>
    </GetUserDetailsByUserNameResponse>
  </soap:Body>
</soap:Envelope>
```


Get user's fax settings

Parameters	Type	Description
adminID	long/bigint	Required. A valid admin user ID.
userID	long/bigint	Required. A valid user ID

Parameters	Type	Description
GetUserFaxSettingByUserIDResult	string	Result info: Return string

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetUserFaxSettingByUserID>
      <ser:adminID>2010211</ser:adminID>
      <ser:userID>21231114</ser:userID>
    </ser:GetUserFaxSettingByUserID>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetUserFaxSettingByUserIDResponse xmlns="http://faxcoreserver/services/">
      <GetUserFaxSettingByUserIDResult><![CDATA[<?xml version="1.0" encoding="utf-8"?>
<UserFaxSetting xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <UserID>21231114</UserID>
  <CountryCode>60</CountryCode>
  <AreaCode />
  <DialPrefix />
  <DialPrefixRule>Use agent configuration only</DialPrefixRule>
  <DialSuffix />
  <DialSuffixRule>Use agent configuration only</DialSuffixRule>
  <LocalCSID>www.faxcore.com.my</LocalCSID>
  <LocalCallerID />
  <LocalNumber />
  <DefaultSendFaxFormat>Fax</DefaultSendFaxFormat>
  <EmailSendNotifyFormat>With Attachment</EmailSendNotifyFormat>
  <EmailSendAttachMaxSize>1000 KB</EmailSendAttachMaxSize>
  <Resolution>Standard (204x98dpi)</Resolution>
  <FaxHeader>%d %t  %-23c  -&gt; %-25n %r %s %p</FaxHeader>
  <PageSendLimit>100</PageSendLimit>
```

```

<AllowFaxRecvPartially>Yes</AllowFaxRecvPartially>
<EmailRecvNotifyFormat>With Attachment</EmailRecvNotifyFormat>
<EmailRecvAttachMaxSize>9 MB</EmailRecvAttachMaxSize>
<UnassignedMessageUser>System-Unrouted ( SYSTEM\SYS-UNROUTED )</UnassignedMessageUser>
<DefaultSendCoverPage>Default CoverPage</DefaultSendCoverPage>
</UserFaxSetting>]]></GetUserFaxSettingByUserIDResult>
  </GetUserFaxSettingByUserIDResponse>
</soap:Body>
</soap:Envelope>

```

GetUserFaxSettingByUserName

Get user's fax settings

Input:

Parameters	Type	Description
adminID	long/bigint	Required. A valid admin user ID.
Username	string	Required. A valid username.

Output:

Parameters	Type	Description
GetUserFaxSettingByUserNameResult	string	Result info: Return string

Sample XML request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetUserFaxSettingByUserName>
      <ser:adminID>2010211</ser:adminID>
      <ser:userName>testuser</ser:userName>
    </ser:GetUserFaxSettingByUserName>
  </soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetUserFaxSettingByUserNameResponse xmlns="http://faxcoreserver/services/">
      <GetUserFaxSettingByUserNameResult><![CDATA[<?xml version="1.0" encoding="utf-8"?>
<UserFaxSetting xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <UserID>21231114</UserID>
  <CountryCode>60</CountryCode>
  <AreaCode />
  <DialPrefix />
  <DialPrefixRule>Use agent configuration only</DialPrefixRule>

```

```

<DialSuffix />
<DialSuffixRule>Use agent configuration only</DialSuffixRule>
<LocalCSID>www.faxcore.com.my</LocalCSID>
<LocalCallerID />
<LocalNumber />
<DefaultSendFaxFormat>Fax</DefaultSendFaxFormat>
<EmailSendNotifyFormat>With Attachment</EmailSendNotifyFormat>
<EmailSendAttachMaxSize>1000 KB</EmailSendAttachMaxSize>
<Resolution>Standard (204x98dpi)</Resolution>
<FaxHeader>%d %t  %-23c  -&gt; % -25n %r %s %p</FaxHeader>
<PageSendLimit>100</PageSendLimit>
<AllowFaxRecvPartially>Yes</AllowFaxRecvPartially>
<EmailRecvNotifyFormat>With Attachment</EmailRecvNotifyFormat>
<EmailRecvAttachMaxSize>9 MB</EmailRecvAttachMaxSize>
<UnassignedMessageUser>System-Unrouted ( SYSTEM\SYS-UNROUTED )</UnassignedMessageUser>
<DefaultSendCoverPage>Default CoverPage</DefaultSendCoverPage>
</UserFaxSetting>]]></GetUserFaxSettingByUserNameResult>
  </GetUserFaxSettingByUserNameResponse>
</soap:Body>
</soap:Envelope>

```

GetUserListWithDomainID

Get user's list with domain ID.

Input:

Parameters	Type	Description
adminID	long/bigint	Required. A valid admin user ID.
domainID	long/bigint	Required. A valid domain ID

Output:

Parameters	Type	Description
GetUserListWithDomainIDResult	Array of string	Result info: Return array of string. {userID userName}

Sample XML request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetUserListWithDomainID>
      <ser:adminID>2010211</ser:adminID>
      <ser:domainID>132021</ser:domainID>
    </ser:GetUserListWithDomainID>
  </soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetUserListWithDomainIDResponse xmlns="http://faxcoreserver/services/">
      <GetUserListWithDomainIDResult>
        <string>2001||ADMIN</string>
        <string>101180032002800548||DANIEL</string>
        <string>200700||EM-CODES</string>
        <string>101800132002800161||ROBOFAX</string>
        <string>101800132002800533||RTONG</string>
        <string>2002002||SYS-MAILER</string>
        <string>2003300||SYS-RECEIVE</string>
        <string>2004430||SYS-STRESS</string>
        <string>205212||SYS-TEST</string>
        <string>200000||SYS-UNROUTED</string>
        <string>101800132002800637||TLARD</string>
      </GetUserListWithDomainIDResult>
    </GetUserListWithDomainIDResponse>
  </soap:Body>
</soap:Envelope>

```

GetUserListWithDomainName

Get user's with domain name.

Input:

Parameters	Type	Description
adminID	long/bigint	Required. A valid admin user ID.
domainName	string	Required. A valid domain name.

Output:

Parameters	Type	Description
GetUserListWithDomainNameResult	Array of string	Result info: Return array of string

Sample XML request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:GetUserListWithDomainName>
      <ser:adminID>2010211</ser:adminID>
      <ser:domainName>testdomain</ser:domainName>
    </ser:GetUserListWithDomainName>
  </soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetUserListWithDomainNameResponse xmlns="http://faxcoreserver/services/">
      <GetUserListWithDomainNameResult>
        <string>11180032002800608||BFOWL</string>
        <string>11180032002801249||TEST2</string>
        <string>11180032002801250||TEST3</string>
        <string>2114432||TESTUSER</string>
        <string>101800320028012481||TESTUSER1</string>
      </GetUserListWithDomainNameResult>
    </GetUserListWithDomainNameResponse>
  </soap:Body>
</soap:Envelope>

```

IsMessageDeleted

Check message delete status.

Input:

Parameters	Type	Description
messageID	string	Required. A valid message ID.
userID	string	Required. A valid user ID

Output:

Parameters	Type	Description
IsMessageDeletedResult	string	Result info: Return true/false in string

Sample XML request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:IsMessageDeleted>
      <ser:messageID>2081100437538</ser:messageID>
      <ser:userID>2010211</ser:userID>
    </ser:IsMessageDeleted>
  </soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <IsMessageDeletedResponse xmlns="http://faxcoreserver/services/">
      <IsMessageDeletedResult>False</IsMessageDeletedResult>
    </IsMessageDeletedResponse>
  </soap:Body>
</soap:Envelope>

```

IsMessageRead

Check message read status.

Input:

Parameters	Type	Description
userID	long/bigint	Required. A valid user ID.
msgID	string	Required. A valid message ID

Output:

Parameters	Type	Description
IsMessageReadResult	string	Result info: Return true/false in string

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:IsMessageRead>
      <ser:userID>2010211</ser:userID>
      <ser:msgID>2081100437538</ser:msgID>
    </ser:IsMessageRead>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <IsMessageReadResponse xmlns="http://faxcoreserver/services/">
      <IsMessageReadResult>true</IsMessageReadResult>
    </IsMessageReadResponse>
  </soap:Body>
</soap:Envelope>
```

MarkMsgAsDownloaded

Set message download status.

Input:

Parameters	Type	Description
msgID	string	Required. A valid message ID.
userID	long/bigint	Required. A valid user ID

Output:

Parameters	Type	Description
MarkMsgAsDownloadedResult	string	Result info: OK = success

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:MarkMsgAsDownloaded>
      <ser:userID>2010211</ser:userID>
      <ser:msgID>2081004375328</ser:msgID>
    </ser:MarkMsgAsDownloaded>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <MarkMsgAsDownloadedResponse xmlns="http://faxcoreserver/services/">
      <MarkMsgAsDownloadedResult>OK</MarkMsgAsDownloadedResult>
    </MarkMsgAsDownloadedResponse>
  </soap:Body>
</soap:Envelope>
```

MarkMsgAsRead

Set message read status.

Input:

Parameters	Type	Description
msgID	string	Required. A valid message ID.
userID	long/bigint	Required. A valid user ID

Output:

Parameters	Type	Description
MarkMsgAsReadResult	string	Result info: OK = success

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
```

```

<soapenv:Header/>
<soapenv:Body>
  <ser:MarkMsgAsRead>
    <ser:userID>2010211</ser:userID>
    <ser:msgID>2081004375328</ser:msgID>
  </ser:MarkMsgAsRead>
</soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <MarkMsgAsReadResponse xmlns="http://faxcoreserver/services/">
      <MarkMsgAsReadResult>OK</MarkMsgAsReadResult>
    </MarkMsgAsReadResponse>
  </soap:Body>
</soap:Envelope>

```

MarkMsgAsUnread

Set message read status.

Input:

Parameters	Type	Description
msgID	String	Required. A valid message ID.
userID	Long/bigint	Required. A valid user ID

Output:

Parameters	Type	Description
MarkMsgAsUnreadResult	string	Result info: OK = success

Sample XML request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:MarkMsgAsUnread>
      <ser:userID>2001</ser:userID>
      <ser:msgID>2081004375338</ser:msgID>
    </ser:MarkMsgAsUnread>
  </soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

```



```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <MarkMsgAsUnreadResponse xmlns="http://faxcoreserver/services/">
      <MarkMsgAsUnreadResult>OK</MarkMsgAsUnreadResult>
    </MarkMsgAsUnreadResponse>
  </soap:Body>
</soap:Envelope>

```

MoveMessage

Move message from one folder to another.

Input:

Parameters	Type	Description
msgID	long/bigint	Required. A valid message ID.
userID	long/bigint	Required. A valid user ID
newFolderName	string	Required. New destination folder name.

Output:

Parameters	Type	Description
MoveMessageResult	string	Result info: OK = success

Sample XML request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:MoveMessage>
      <ser:userID>2010211</ser:userID>
      <ser:msgID>2081004375338</ser:msgID>
      <ser:newFolderName>processed</ser:newFolderName>
    </ser:MoveMessage>
  </soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <MoveMessageResponse xmlns="http://faxcoreserver/services/">
      <MoveMessageResult>MESSAGE_NOT_FOUND</MoveMessageResult>
    </MoveMessageResponse>
  </soap:Body>
</soap:Envelope>

```

RemoveUser

Delete user from system.

Input:

Parameters	Type	Description
adminID	long/bigint	Required. A valid admin user ID.
userID	long/bigint	Required. A valid user ID

Output:

Parameters	Type	Description
RemoveUserResult	string	Result info: OK = success

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:RemoveUser>
      <ser:adminID>2010211</ser:adminID>
      <ser:userID>21212</ser:userID>
    </ser:RemoveUser>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <RemoveUserResponse xmlns="http://faxcoreserver/services/">
      <RemoveUserResult>INVALID_USER</RemoveUserResult>
    </RemoveUserResponse>
  </soap:Body>
</soap:Envelope>
```

RetryFailed

Retry failed message.

Input:

Parameters	Type	Description
msgID	string	Required. A valid message ID.

Output:

Parameters	Type	Description
------------	------	-------------

RetryFailedResult	string	Result info: OK = success
-------------------	--------	------------------------------

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:RetryFailed>
      <ser:msgID>123232122</ser:msgID>
    </ser:RetryFailed>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <RetryFailedResponse xmlns="http://faxcoreserver/services/">
      <RetryFailedResult>MESSAGE_NOT_FOUND</RetryFailedResult>
    </RetryFailedResponse>
  </soap:Body>
</soap:Envelope>
```

SearchIBRouteRules

Search inbound routing rules.

Input:

Parameters	Type	Description
adminID	long/bigint	Required. A valid admin user ID.
Input	string	Required. Search string

Output:

Parameters	Type	Description
SearchIBRouteRulesResult	string	Result info: Routing rules in XML string.

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:SearchIBRouteRules>
      <ser:adminID>2010211</ser:adminID>
      <ser:input>robofax</ser:input>
    </ser:SearchIBRouteRules>
  </soapenv:Body>
</soapenv:Envelope>
```

```
</soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SearchIBRouteRulesResponse xmlns="http://faxcoreserver/services/">
      <SearchIBRouteRulesResult><![CDATA[<?xml version="1.0" encoding="utf-8"?>
<Rules xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <IBRouteRules>
    <IBRouteRule>
      <RouteID>602</RouteID>
      <RouteByString>[Remote CSID] = 'ROBOFAX'</RouteByString>
      <ForwardAddress>ROBOFAX (SYSTEM)</ForwardAddress>
      <IsTimeBasedRouting>>false</IsTimeBasedRouting>
      <IsLoopBack>>false</IsLoopBack>
      <IsActive>>true</IsActive>
      <Priority>110</Priority>
    </IBRouteRule>
    <IBRouteRule>
      <RouteID>69</RouteID>
      <RouteByString>[Routing Digit] = '0327268805'</RouteByString>
      <ForwardAddress>ROBOFAX (SYSTEM)</ForwardAddress>
      <IsTimeBasedRouting>>false</IsTimeBasedRouting>
      <IsLoopBack>>false</IsLoopBack>
      <IsActive>>true</IsActive>
      <Priority>100</Priority>
    </IBRouteRule>
  </IBRouteRules>
</Rules>]]></SearchIBRouteRulesResult>
    </SearchIBRouteRulesResponse>
  </soap:Body>
</soap:Envelope>
```

SendCompleteMessage

Send message to multiple recipients with multiple documents.

Input:

Parameters	Type	Description
userID	long/bigint	Required. A valid user ID
senderName	string	Sender name
senderCompName	string	Sender company name
subject	string	Message subject
note	string	Message note
billingCode	string	Message billing code
scheduleDate	string	Message schedule date
priority	int	Message priority
isOnHold	boolean	Set message to on hold.
mss	boolean	Indicate whether a notification is

		sent to sender when message is sent success.
msf	boolean	Indicate whether a notification is sent to sender when message is sent failed.
trackLabel	array of string	List of tracking label
trackValue	array of string	List of tracking value
recpName	array of string	List of recipient's name
recpCompName	array of string	List of recipient's company name
recpAddress	array of string	List of recipient's address
recpRawFax	array of boolean	List of Boolean to indicate whether the recipient address is raw fax.
recpNotifyAddr	array of string	List of recipient's notify address in email format
recpIsRRFOn	array of boolean	List of Boolean to indicate whether to notify the recipient when message is received failed
recpIsRRSON	array of boolean	List of Boolean to indicate whether to notify the recipient when message is received success
documentName	array of string	List of document name. if the document is a FaxCore document, use the name as stated in FaxCore document store. Else, put the document name along with its extension to represent the attached file.
documentPath	array of string	List of document's path. If the document is stored in FaxCore server, use the full document path, otherwise leave it empty.
documentBytes	array of anytype	List of document byte array witch represent the attached document in binary. Leave it as 'null' if it is not an attachment.
documentIsMerge	array of boolean	List of Boolean to indicates whether the document is a merge file that contain FaxCore tokens. Only applicable for HTML file.

Output:

Parameters	Type	Description
SendCompleteMessageResult	string	Result info: Return message ID.

Sample XML request:

<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://faxcoreserver/services/"> <soapenv:Header/> <soapenv:Body> <ser:SendCompleteMessage> <ser:userID>2010211</ser:userID> </ser:SendCompleteMessage> </soapenv:Body> </soapenv:Envelope></pre>

```

<ser:senderName>John Wick</ser:senderName>
<ser:senderCompName>Food Inc</ser:senderCompName>
<ser:subject>Test</ser:subject>
<ser:note>Note goes here</ser:note>
<ser:billingCode></ser:billingCode>
<ser:scheduleDate></ser:scheduleDate>
<ser:priority>70</ser:priority>
<ser:isOnHold>true</ser:isOnHold>
<ser:mss>false</ser:mss>
<ser:msf>false</ser:msf>
<ser:trackLabel>
  <!--One or more repetitions:-->
  <ser:string>Label 1</ser:string>
  <ser:string>Label 2</ser:string>
</ser:trackLabel>
<ser:trackValue>
  <!--One or more repetitions:-->
  <ser:string>Value 1</ser:string>
  <ser:string>Value 2</ser:string>
</ser:trackValue>
<ser:recpName>
  <!--One or more repetitions:-->
  <ser:string>AAA</ser:string>
  <ser:string>BBB</ser:string>
</ser:recpName>
<ser:recpCompName>
  <!--One or more repetitions:-->
  <ser:string>Comp A</ser:string>
  <ser:string>Comp B</ser:string>
</ser:recpCompName>
<ser:recpAddress>
  <!--One or more repetitions:-->
  <ser:string>720129922</ser:string>
  <ser:string>023323422</ser:string>
</ser:recpAddress>
<ser:recpRawFax>
  <!--One or more repetitions:-->
  <ser:boolean>>false</ser:boolean>
  <ser:boolean>true</ser:boolean>
</ser:recpRawFax>
<ser:recpNotifyAddr>
  <!--One or more repetitions:-->
  <ser:string></ser:string>
  <ser:string></ser:string>
</ser:recpNotifyAddr>
<ser:recpIsRRFOn>
  <!--One or more repetitions:-->
  <ser:boolean>>false</ser:boolean>
  <ser:boolean>>false</ser:boolean>
</ser:recpIsRRFOn>
<ser:recpIsRRSON>
  <!--One or more repetitions:-->
  <ser:boolean>>false</ser:boolean>
  <ser:boolean>>false</ser:boolean>
</ser:recpIsRRSON>
<ser:documentName>

```

```

    <!--One or more repetitions:-->
    <ser:string>Doc.pdf</ser:string>
    <ser:string>Doc.xlsx</ser:string>
  </ser:documentName>
  <ser:documentPath>
    <!--One or more repetitions:-->
    <ser:string></ser:string>
    <ser:string></ser:string>
  </ser:documentPath>
  <ser:documentBytes>
    <!--One or more repetitions:-->
    <ser:anyType>[binary data]</ser:anyType>
    <ser:anyType>[binary data]</ser:anyType>
  </ser:documentBytes>
  <ser:documentIsMerge>
    <!--One or more repetitions:-->
    <ser:boolean>>false</ser:boolean>
    <ser:boolean>true</ser:boolean>
  </ser:documentIsMerge>
</ser:SendMessage>
</soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SendMessageResponse xmlns="http://faxcoreserver/services/">
      <SendMessageResult>20108092123922</SendMessageResult>
    </SendMessageResponse>
  </soap:Body>
</soap:Envelope>

```

BasicSendMessageWithBytes

Basic send message with documents from bytes.

Input:

Parameters	Type	Description
userID	long/bigint	Required. A valid user ID
senderName	string	Sender name
senderCompName	string	Sender company name
Subject	string	Message subject
note	string	Message note
trackLabel	array of string	List of tracking label
trackValue	array of string	List of tracking value
recpName	array of string	List of recipient's name
recpCompName	array of string	List of recipient's company name
recpAddress	array of string	List of recipient's address
recpRawFax	array of boolean	List of Boolean to indicate whether the recipient address is raw fax.
recpNotifyAddr	array of string	List of recipient's notify address in

		email format
recpIsRRFOn	array of boolean	List of Boolean to indicate whether to notify the recipient when message is received failed
recpIsRRSON	array of boolean	List of Boolean to indicate whether to notify the recipient when message is received success
documentName	array of string	List of document name. if the document is a FaxCore document, use the name as stated in FaxCore document store. Else, put the document name along with its extension to represent the attached file.
documentPath	array of string	List of document's path. If the document is stored in FaxCore server, use the full document path, otherwise leave it empty.
documentBytes	array of anytype	List of document byte array witch represent the attached document in binary. Leave it as 'null' if it is not an attachment.
documentIsMerge	array of boolean	List of Boolean to indicates whether the document is a merge file that contain FaxCore tokens. Only applicable for HTML file.

Output:

Parameters	Type	Description
BasicSendMessageWithBytesResult	string	Result info: Return message ID.

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:BasicSendMessageWithBytes>
      <ser:userID>2010211</ser:userID>
      <ser:senderName>Johnny</ser:senderName>
      <ser:senderCompName>Fax Inc</ser:senderCompName>
      <ser:subject>test</ser:subject>
      <ser:note></ser:note>
      <ser:trackLabel>
        <!--One or more repetitions:-->
        <ser:string></ser:string>
      </ser:trackLabel>
      <ser:trackValue>
        <!--One or more repetitions:-->
        <ser:string></ser:string>
      </ser:trackValue>
      <ser:recpName>
```



```

    <!--One or more repetitions:-->
    <ser:string>Dan</ser:string>
  </ser:recpName>
  <ser:recpAddress>
    <!--One or more repetitions:-->
    <ser:string>170923202</ser:string>
  </ser:recpAddress>
  <ser:recpRawFax>
    <!--One or more repetitions:-->
    <ser:boolean>true</ser:boolean>
  </ser:recpRawFax>
  <ser:documentName>
    <!--One or more repetitions:-->
    <ser:string>test.pdf</ser:string>
  </ser:documentName>
  <ser:documentBytes>
    <!--One or more repetitions:-->
    <ser:anyType>[binary data]</ser:anyType>
  </ser:documentBytes>
  <ser:documentIsMerge>
    <!--One or more repetitions:-->
    <ser:boolean>false</ser:boolean>
  </ser:documentIsMerge>
</ser:BasicSendMessageWithBytes>
</soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <BasicSendMessageWithBytesResponse xmlns="http://faxcoreserver/services/">
      <BasicSendMessageWithBytesResult>2010293382221</BasicSendMessageWithBytesResult>
    </BasicSendMessageWithBytesResponse>
  </soap:Body>
</soap:Envelope>

```

SendCompleteMessageWithPresetAgents

Send message to multiple recipients with multiple documents using specific agent/port.

Input:

Parameters	Type	Description
userID	long/bigint	Required. A valid user ID
senderName	string	Sender name
senderCompName	string	Sender company name
subject	string	Message subject
note	string	Message note
billingCode	string	Message billing code
scheduleDate	string	Message schedule date
priority	int	Message priority
isOnHold	boolean	Set message to on hold.

mss	boolean	Indicate whether a notification is sent to sender when message is sent success.
msf	boolean	Indicate whether a notification is sent to sender when message is sent failed.
trackLabel	array of string	List of tracking label
trackValue	array of string	List of tracking value
recpName	array of string	List of recipient's name
recpCompName	array of string	List of recipient's company name
recpAddress	array of string	List of recipient's address
recpRawFax	array of boolean	List of Boolean to indicate whether the recipient address is raw fax.
recpNotifyAddr	array of string	List of recipient's notify address in email format
recpIsRRFOn	array of boolean	List of Boolean to indicate whether to notify the recipient when message is received failed
recpIsRRSON	array of boolean	List of Boolean to indicate whether to notify the recipient when message is received success
documentName	array of string	List of document name, if the document is a FaxCore document, use the name as stated in FaxCore document store. Else, put the document name along with its extension to represent the attached file.
documentPath	array of string	List of document's path. If the document is stored in FaxCore server, use the full document path, otherwise leave it empty.
documentBytes	array of anytype	List of document byte array witch represent the attached document in binary. Leave it as 'null' if it is not an attachment.
documentIsMerge	array of boolean	List of Boolean to indicates whether the document is a merge file that contain FaxCore tokens. Only applicable for HTML file.
agentID	array of int	Required. Valid agent ID
presetType	array of int	Required. 1=port 2=port group
presetValue	array of int	Required. Valid port number.

Output:

Parameters	Type	Description
SendCompleteMessageWithPresetAgentsResult	string	Result info: Return message ID.

Sample XML request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:SendCompleteMessageWithPresetAgents>
      <ser:userID>2010211</ser:userID>
      <ser:senderName>Dan</ser:senderName>
      <ser:senderCompName>FastTrack</ser:senderCompName>
      <ser:subject>test</ser:subject>
      <ser:note></ser:note>
      <ser:billingCode></ser:billingCode>
      <ser:scheduleDate></ser:scheduleDate>
      <ser:priority>70</ser:priority>
      <ser:isOnHold>>false</ser:isOnHold>
      <ser:mss>>true</ser:mss>
      <ser:msf>>true</ser:msf>
      <ser:trackLabel>
        <!--One or more repetitions:-->
        <ser:string></ser:string>
      </ser:trackLabel>
      <ser:trackValue>
        <!--One or more repetitions:-->
        <ser:string></ser:string>
      </ser:trackValue>
      <ser:recpName>
        <!--One or more repetitions:-->
        <ser:string>Ken</ser:string>
      </ser:recpName>
      <ser:recpCompName>
        <!--One or more repetitions:-->
        <ser:string>Industries</ser:string>
      </ser:recpCompName>
      <ser:recpAddress>
        <!--One or more repetitions:-->
        <ser:string>17208704141</ser:string>
      </ser:recpAddress>
      <ser:recpRawFax>
        <!--One or more repetitions:-->
        <ser:boolean>>false</ser:boolean>
      </ser:recpRawFax>
      <ser:recpNotifyAddr>
        <!--One or more repetitions:-->
        <ser:string></ser:string>
      </ser:recpNotifyAddr>
      <ser:recpIsRRFOn>
        <!--One or more repetitions:-->
        <ser:boolean>>false</ser:boolean>
      </ser:recpIsRRFOn>
      <ser:recpIsRRSON>
        <!--One or more repetitions:-->
        <ser:boolean>>false</ser:boolean>
      </ser:recpIsRRSON>
      <ser:documentName>
        <!--One or more repetitions:-->
        <ser:string>Defaultcoverpage</ser:string>
      </ser:documentName>
    </ser:SendCompleteMessageWithPresetAgents>
  </soapenv:Body>
</soapenv:Envelope>

```

```

<ser:documentPath>
  <!--One or more repetitions:-->
  <ser:string></ser:string>
</ser:documentPath>
<ser:documentBytes>
  <!--One or more repetitions:-->
  <ser:anyType></ser:anyType>
</ser:documentBytes>
<ser:documentIsMerge>
  <!--One or more repetitions:-->
  <ser:boolean>true</ser:boolean>
</ser:documentIsMerge>
<ser:agentID>
  <!--One or more repetitions:-->
  <ser:int>100009</ser:int>
</ser:agentID>
<ser:presetType>
  <!--One or more repetitions:-->
  <ser:int>1</ser:int>
</ser:presetType>
<ser:presetValue>
  <!--One or more repetitions:-->
  <ser:int>1</ser:int>
</ser:presetValue>
</ser:SendMessageWithPresetAgents>
</soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SendMessageWithPresetAgentsResponse xmlns="http://faxcoreserver/services/">

    <SendMessageWithPresetAgentsResult>20109801211</SendMessageWithPresetAgentsResult>
    </SendMessageWithPresetAgentsResponse>
  </soap:Body>
</soap:Envelope>

```

SendMessageWithPresetAgentsAndByPassRender

Send message to multiple recipients with multiple document using specific agent/port and bypass rendering process.

Input:

Parameters	Type	Description
userID	long/bigint	Required. A valid user ID
senderName	string	Sender name
senderCompName	string	Sender company name
subject	string	Message subject
note	string	Message note
billingCode	string	Message billing code

scheduleDate	string	Message schedule date
Priority	int	Message priority
isOnHold	boolean	Set message to on hold.
Mss	boolean	Indicate whether a notification is sent to sender when message is sent success.
Msf	boolean	Indicate whether a notification is sent to sender when message is sent failed.
trackLabel	array of string	List of tracking label
trackValue	array of string	List of tracking value
recpName	array of string	List of recipient's name
recpCompName	array of string	List of recipient's company name
recpAddress	array of string	List of recipient's address
recpRawFax	array of boolean	List of Boolean to indicate whether the recipient address is raw fax.
recpNotifyAddr	array of string	List of recipient's notify address in email format
recpIsRRFOn	array of boolean	List of Boolean to indicate whether to notify the recipient when message is received failed
recpIsRRSON	array of boolean	List of Boolean to indicate whether to notify the recipient when message is received success
documentName	array of string	List of document name. if the document is a FaxCore document, use the name as stated in FaxCore document store. Else, put the document name along with its extension to represent the attached file.
documentPath	array of string	List of document's path. If the document is stored in FaxCore server, use the full document path, otherwise leave it empty.
documentBytes	array of anytype	List of document byte array witch represent the attached document in binary. Leave it as 'null' if it is not an attachment.
documentIsMerge	array of boolean	List of Boolean to indicates whether the document is a merge file that contain FaxCore tokens. Only applicable for HTML file.
agentID	array of int	Required. Valid agent ID
presetType	array of int	Required. port = 1 port group =2
presetValue	array of int	Required. Port no
byPassRender	boolean	Required. Set value 'true' to bypass rendering

Output:

Parameters	Type	Description
------------	------	-------------

SendCompleteMessageWithPresetAgentsAndByPassRenderResult	string	Result info: Return message ID.
--	--------	------------------------------------

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:SendCompleteMessageWithPresetAgentsAndByPassRender>
      <ser:userID>2010211</ser:userID>
      <ser:senderName>John</ser:senderName>
      <ser:senderCompName>FTI</ser:senderCompName>
      <ser:subject>test</ser:subject>
      <ser:note></ser:note>
      <ser:billingCode></ser:billingCode>
      <ser:scheduleDate></ser:scheduleDate>
      <ser:priority>50</ser:priority>
      <ser:isOnHold>>false</ser:isOnHold>
      <ser:mss>>false</ser:mss>
      <ser:msf>>false</ser:msf>
      <ser:trackLabel>
        <!--One or more repetitions:-->
        <ser:string></ser:string>
      </ser:trackLabel>
      <ser:trackValue>
        <!--One or more repetitions:-->
        <ser:string></ser:string>
      </ser:trackValue>
      <ser:recpName>
        <!--One or more repetitions:-->
        <ser:string>Nyugen</ser:string>
      </ser:recpName>
      <ser:recpCompName>
        <!--One or more repetitions:-->
        <ser:string>JTI</ser:string>
      </ser:recpCompName>
      <ser:recpAddress>
        <!--Zero or more repetitions:-->
        <ser:string>17208704141</ser:string>
      </ser:recpAddress>
      <ser:recpRawFax>
        <!--One or more repetitions:-->
        <ser:boolean>>true</ser:boolean>
      </ser:recpRawFax>
      <ser:recpNotifyAddr>
        <!--One or more repetitions:-->
        <ser:string></ser:string>
      </ser:recpNotifyAddr>
      <ser:recpIsRRFOn>
        <!--One or more repetitions:-->
        <ser:boolean>>false</ser:boolean>
      </ser:recpIsRRFOn>
      <ser:recpIsRRSOn>
        <!--One or more repetitions:-->
        <ser:boolean>>false</ser:boolean>
    </ser:SendCompleteMessageWithPresetAgentsAndByPassRender>
  </soapenv:Body>
</soapenv:Envelope>
```

```

</ser:recpIsRRSON>
<ser:documentName>
  <!--One or more repetitions:-->
  <ser:string>Default coverpage</ser:string>
</ser:documentName>
<ser:documentPath>
  <!--One or more repetitions:-->
  <ser:string></ser:string>
</ser:documentPath>
<ser:documentBytes>
  <!--One or more repetitions:-->
  <ser:anyType></ser:anyType>
</ser:documentBytes>
<ser:documentIsMerge>
  <!--One or more repetitions:-->
  <ser:boolean>true</ser:boolean>
</ser:documentIsMerge>
<ser:agentID>
  <!--One or more repetitions:-->
  <ser:int>100009</ser:int>
</ser:agentID>
<ser:presetType>
  <!--One or more repetitions:-->
  <ser:int>1</ser:int>
</ser:presetType>
<ser:presetValue>
  <!--One or more repetitions:-->
  <ser:int>2</ser:int>
</ser:presetValue>
<ser:byPassRender>false</ser:byPassRender>
</ser:SendCompleteMessageWithPresetAgentsAndByPassRender>
</soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SendCompleteMessageWithPresetAgentsAndByPassRenderResponse xmlns="http://faxcoreserver/services/">

<SendCompleteMessageWithPresetAgentsAndByPassRenderResult>2010808972121</SendCompleteMessageWith
PresetAgentsAndByPassRenderResult>
    </SendCompleteMessageWithPresetAgentsAndByPassRenderResponse>
  </soap:Body>
</soap:Envelope>

```

SendMixedMode

Send message in mixed mode.[This method only available for .NET implementation]

Input:

Parameters	Type	Description
------------	------	-------------

userID	long/bigint	Required. A valid user ID
senderName	string	Sender name
senderCompName	string	Sender company name
subject	string	Message subject
note	string	Message note
billingCode	string	Message billing code
scheduleDate	string	Message schedule date
priority	int	Message priority
isOnHold	boolean	Set message to on hold.
mss	boolean	Indicate whether a notification is sent to sender when message is sent success.
msf	boolean	Indicate whether a notification is sent to sender when message is sent failed.
useDefaultNotify	boolean	Use default notification setting set in sender's profile
useDefaultCP	boolean	Attach default cover page to message.
Documents	array of MsgDocuments	See sample code
Recipients	array of MsgRecipient	See sample code. AddressType: 2 = fax 1 = email 9 = raw fax
Trackings	array of MsgTracking	See sample code. TrackID will always be empty

Output:

Parameters	Type	Description
SendMixedModeResult	string	Result info: Return message ID.

Sample XML request:

n/a

Sample XML response:

n/a

ServerStatus

Check server status.

Input:

Parameters	Type	Description
n/a		

Output:

Parameters	Type	Description
ServerStatusResult	string	Result info: OK = success

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:ServerStatus/>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ServerStatusResponse xmlns="http://faxcoreserver/services/">
      <ServerStatusResult>OK</ServerStatusResult>
    </ServerStatusResponse>
  </soap:Body>
</soap:Envelope>
```

UpdateTrackRecord

Update tracking information.

Input:

Parameters	Type	Description
msgID	Long/bigint	Required. A valid message ID.
trackID	long/bigint	Required. A valid tracking ID for specified message.
trackValue	string	Required. New tracking value.

Output:

Parameters	Type	Description
UpdateTrackRecordResult	string	Result info: OK = success

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:MoveMessge>
      <ser:msgID>2081004156346</ser:msgID>
      <ser:trackID>1</ser:trackID>
      <ser:trackValue>new value</ser:trackValue>
    </ser:MoveMessge>
  </soapenv:Body>
```

```
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <MoveMessgeResponse xmlns="http://faxcoreserver/services/">
      <MoveMessgeResult>OK</MoveMessgeResult>
    </MoveMessgeResponse>
  </soap:Body>
</soap:Envelope>
```

UpdateUser-basic

Update user basic profile info.

Input:

Parameters	Type	Description
adminID	long/bigint	Required. A valid admin user ID.
userID	long/bigint	Required. A valid user ID.
role	string	Required. enum { 'SystemAdmin', 'DomainAdmin', 'SuperUser', 'StandardUser' }
isExternalAuth	boolean	Set 'true' to indicate user authenticate externally.
isActive	boolean	Set 'true' to activate user account.
displayName	string	
firstName	string	
middleName	string	
lastName	string	
companyName	string	
preferAddrType	string	enum { 'Fax', 'Email', 'FaxRaw' }
desc	string	

Output:

Parameters	Type	Description
UpdateUser-basicResult	string	Result info: OK = success

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:UpdateUser-basic>
      <ser:adminID>2001</ser:adminID>
      <ser:userID>2114</ser:userID>
      <ser:role>DomainAdmin</ser:role>
      <ser:isExternalAuth>true</ser:isExternalAuth>
```

```

    <ser:isActive>true</ser:isActive>
    <ser:displayName>new display name</ser:displayName>
    <ser:firstName>new first name</ser:firstName>
    <ser:middleName>new middle name</ser:middleName>
    <ser:lastName>new last name</ser:lastName>
    <ser:companyName>new company name</ser:companyName>
    <ser:preferAddrType>Email</ser:preferAddrType>
    <ser:desc></ser:desc>
  </ser:UpdateUser-basic>
</soapenv:Body>
</soapenv:Envelope>

```

Sample XML response:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <UpdateUser-basicResponse xmlns="http://faxcoreserver/services/">
      <UpdateUser-basicResult>OK</UpdateUser-basicResult>
    </UpdateUser-basicResponse>
  </soap:Body>
</soap:Envelope>

```

UpdateUser-extra

Update user basic profile info, local CSID and caller ID.

Input:

Parameters	Type	Description
adminID	long/bigint	Required. A valid admin user ID.
userID	long/bigint	Required. A valid user ID.
role	string	Required. enum { 'SystemAdmin', 'DomainAdmin', 'SuperUser', 'StandardUser' }
isExternalAuth	boolean	Set 'true' to indicate user authenticate externally.
isActive	boolean	Set 'true' to activate user account.
displayName	string	
firstName	string	
middleName	string	
lastName	string	
companyName	string	
preferAddrType	string	enum { 'Fax', 'Email', 'FaxRaw' }
desc	string	
localCSID	string	
callerID	string	

Output:

Parameters	Type	Description
UpdateUser-extraResult	string	Result info:

		OK = success
--	--	--------------

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:UpdateUser-extra>
      <ser:adminID>2001</ser:adminID>
      <ser:userID>2114</ser:userID>
      <ser:role>DomainAdmin</ser:role>
      <ser:isExternalAuth>>false</ser:isExternalAuth>
      <ser:isActive>>false</ser:isActive>
      <ser:displayName>new display name</ser:displayName>
      <ser:firstName>new first name</ser:firstName>
      <ser:middleName>new middle name</ser:middleName>
      <ser:lastName>new last name</ser:lastName>
      <ser:companyName>new company name</ser:companyName>
      <ser:preferAddrType>Email</ser:preferAddrType>
      <ser:desc> </ser:desc>
      <ser:localCSID>new local csid</ser:localCSID>
      <ser:callerID>new caller id</ser:callerID>
    </ser:UpdateUser-extra>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <UpdateUser-extraResponse xmlns="http://faxcoreserver/services/">
      <UpdateUser-extraResult>OK</UpdateUser-extraResult>
    </UpdateUser-extraResponse>
  </soap:Body>
</soap:Envelope>
```

UpdateUser-advance

Update user basic profile info.

Input:

Parameters	Type	Description
adminID	long/bigint	Required. A valid admin user ID.
userID	long/bigint	Required. A valid user ID.
role	string	Required. enum { 'SystemAdmin', 'DomainAdmin', 'SuperUser', 'StandardUser' }
isExternalAuth	boolean	Set 'true' to indicate user authenticate externally.
isActive	boolean	Set 'true' to activate user account.
displayName	string	

firstName	string	
middleName	string	
lastName	string	
companyName	string	
preferAddrType	string	enum { 'Fax', 'Email', 'FaxRaw' }
desc	string	
config	array of UserConfig	See sample code. Refer to document SDK-UserConfigList.xls

Output:

Parameters	Type	Description
UpdateUser-advanceResult	string	Result info: OK = success

Sample XML request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://faxcoreserver/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:UpdateUser-advance>
      <ser:adminID>2010211</ser:adminID>
      <ser:userID>2114</ser:userID>
      <ser:role>DomainAdmin</ser:role>
      <ser:isExternalAuth>>false</ser:isExternalAuth>
      <ser:isActive>>true</ser:isActive>
      <ser:displayName>new display name</ser:displayName>
      <ser:firstName>new first name</ser:firstName>
      <ser:middleName>new middle name</ser:middleName>
      <ser:lastName>new last name</ser:lastName>
      <ser:companyName>new company name</ser:companyName>
      <ser:preferAddrType>Email</ser:preferAddrType>
      <ser:desc> </ser:desc>
      <ser:config>
        <!--One or more repetitions:-->
        <ser:UserConfig>
          <ser:ConfigName>USER.MSG.SEND.SUCCESS</ser:ConfigName>
          <ser:ConfigValue>1</ser:ConfigValue>
        </ser:UserConfig>
        <ser:UserConfig>
          <ser:ConfigName>USER.BROADCAST.ALLOW</ser:ConfigName>
          <ser:ConfigValue>0</ser:ConfigValue>
        </ser:UserConfig>
      </ser:config>
    </ser:UpdateUser-advance>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample XML response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <UpdateUser-advanceResponse xmlns="http://faxcoreserver/services/">
      <UpdateUser-advanceResult>OK</UpdateUser-advanceResult>
    </UpdateUser-advanceResponse>
  </soap:Body>
</soap:Envelope>
```

FaxCore eV5 supports the following Microsoft operating systems:

Windows Server 2008R2 x86/x64

- *with Microsoft .Net 3.5 or .Net 2.0 or .Net 4.0 installed*
- VMware Virtual Machine with Windows Server 2008R2 x86/x64 installed. Only applicable if using FaxCore FoIP Drivers or a remote fax agent with a board.
 - ♦ 64 Bit host is only supported for Windows Server 2008 R2, but both FaxCore and the Dialogic Brooktrout drivers run in 32 bit mode.
- *VMware ESX 4.0 and above, Citrix XenServer and Microsoft Hyper-V R2 virtual machines are fully supported for Dialogic Brooktrout SR140 and etherFAX implementations.*

Operating System – Configuration Requirements and Recommendations

- A fresh stand-alone operating system installation **is required** (not an OS upgrade) with all Windows Updates is required. FaxCore eV5 installer will install the other prerequisites automatically
- Microsoft Office installed to facilitate rendering of MS Office based documents.
- Microsoft .Net 3.5 or .Net 2.0 or .Net 4.0 installed
- Static or DHCP reserved IP address
- Recommended that the FaxCore server is not joined to the Domain.
- Recommended that FaxCore is on a standalone server. FaxCore will not be responsible for any issues related to other software loaded on the server.
- **CANNOT BE LOADED ON THE SAME SERVER AS EXCHANGE.**
- The FaxCore directory, and all sub directories, **must be excluded from any virus scan software.**

Client Software Requirements -- Microsoft Internet Explorer 7.0 or later, Firefox, Chrome or Safari

Hardware Requirements

- CPU: Processor (Celeron 2.4 GHz)
- Memory: Minimum 4GB RAM - additional memory recommended for larger deployments
- Disk Space: 20 GB available disk space (Additional disk space will be required for storage of fax messages)

Dialogic Brooktrout Fax Boards Supported

- **Fax Board:** Fax communication device present

For PSTN:

- Dialogic Brooktrout TR1034 Series Fax Board
 - (full length and full height PCI board, board will require 12.5" of room in PCI slot)

- Dialogic Brooktrout TruFax Series Fax Board

For Fax-Over-IP Internal (connection to PSTN still required):

- FaxCore FoIP Drivers *
- Dialogic Diva Analog boards for use with SR140 (full size form factor for older models require 12.5" of room in PCI slot)
- Dialogic Brooktrout TR1034 Series Fax Board (With FoIP Capability Enabled)

To ensure compatibility of Fax over IP (T.38) verify with your phone system or SIP Trunk Provider

Getting started with FaxCore eV5

Login to FaxCore eV5 for the first time

1. When all the above installation process and configuration is completed, reboot the server.
2. When the server boots up, you can access the FaxCore web interface with Internet Explorer 8.0 or later at the following URLs below:

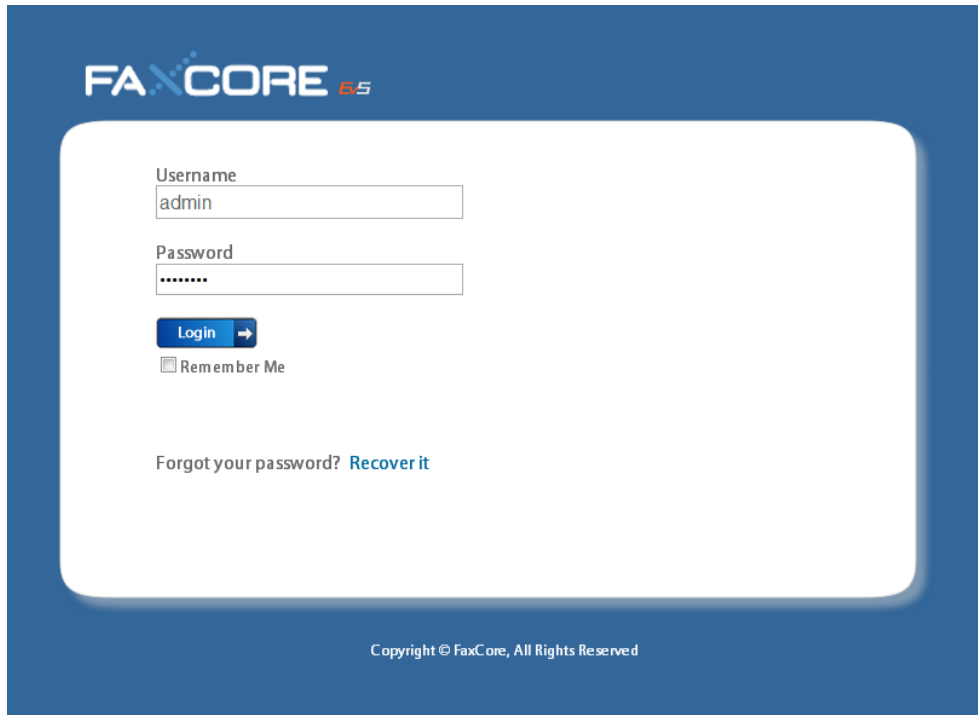
User Web Access: <http://localhost/> or <http://servername/>

Admin Panel Access: <http://localhost/admin/> or <http://servername/admin/>

FaxCore default Admin User Credential:

Username: **admin**

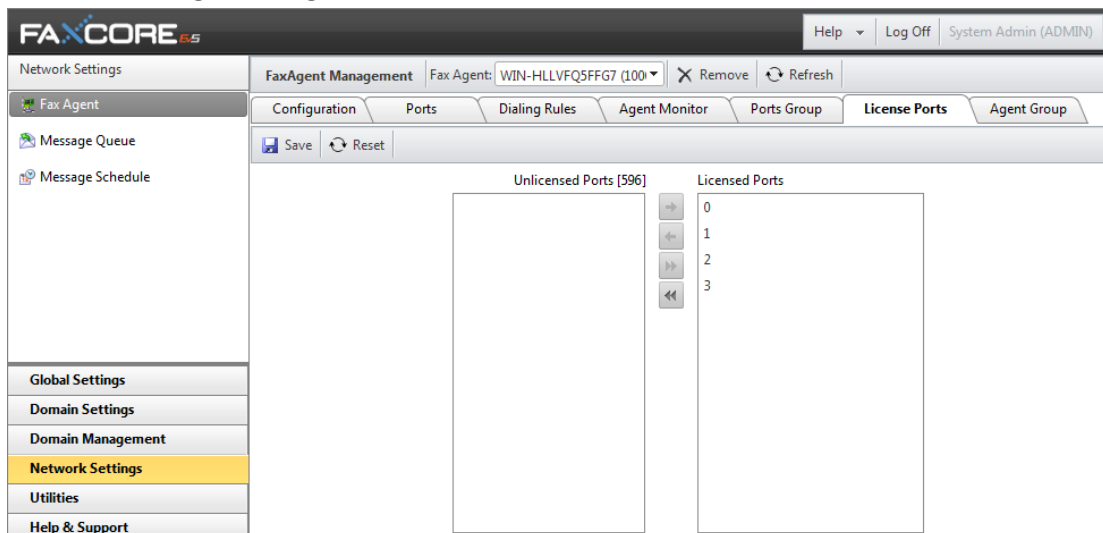
Password: **password**



The image shows the login interface for FaxCore eV5. It features a blue header with the 'FAXCORE eV5' logo. Below the logo is a white rounded rectangle containing the login fields. The 'Username' field is pre-filled with 'admin'. The 'Password' field is masked with dots. There is a blue 'Login' button with a right-pointing arrow, and a checkbox labeled 'Remember Me' below it. At the bottom of the white area, there is a link that says 'Forgot your password? Recover it'. The footer of the page, in a dark blue bar, reads 'Copyright © FaxCore, All Rights Reserved'.

Assigning Ports Licensing

1. If you have successfully activated the software or obtained a trial license by following the steps in “**Activating FaxCore eV5**”, you will now have FaxCore licenses available to use. You will now need to assign these licenses to the available fax ports in the system.
2. To assign the licenses to enable the ports in the system, login to FaxCore eV5 **Admin Panel** > **Network Settings** > **FaxAgent** > **License Ports** tab.



The image shows the 'License Ports' configuration screen in the FaxCore eV5 Admin Panel. The top navigation bar includes 'Help', 'Log Off', and 'System Admin (ADMIN)'. Below this is a 'FaxAgent Management' section with a dropdown menu showing 'WIN-HLLVFQ5FFG7 (100%)' and buttons for 'Remove' and 'Refresh'. The main area has several tabs: 'Configuration', 'Ports', 'Dialing Rules', 'Agent Monitor', 'Ports Group', 'License Ports' (which is selected), and 'Agent Group'. Under the 'License Ports' tab, there are 'Save' and 'Reset' buttons. The central part of the screen is divided into two columns: 'Unlicensed Ports [596]' and 'Licensed Ports'. Between these columns are four arrow buttons: a single right arrow, a single left arrow, a double right arrow, and a double left arrow. The 'Licensed Ports' column currently contains a list of numbers: 0, 1, 2, and 3.

3. Move the ports with >> from Unlicensed Ports section to the License Ports section and click the

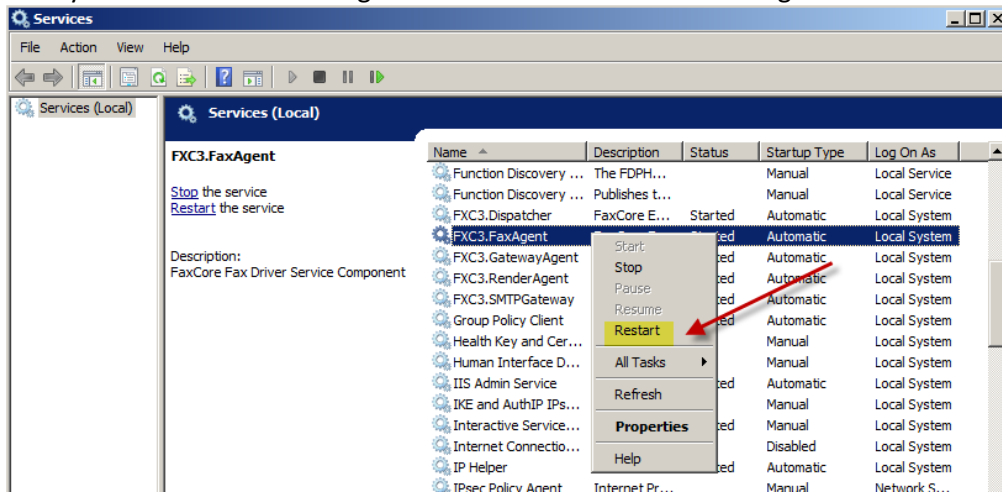
“Save” button.

- Next select the **“Ports”** tab and enable **“Send”** and **“Receive”** for the Ports that you have just assigned licenses to.

The screenshot displays the FAXCORE v5.5 web interface. The top navigation bar includes 'Help', 'Log Off', and 'System Admin (ADMIN)'. Below this, the 'FaxAgent Management' section shows a dropdown menu with 'WIN-HLLVFQ5FFG7 (100)', a 'Remove' button, and a 'Refresh' button. The 'Ports' tab is selected, showing a table of port configurations. The table has columns for 'Module', 'Port', 'Enabled', 'Send Enabled', 'Receive Enabled', and 'Tone Detect Enabled'. The 'Send Enabled' and 'Receive Enabled' columns are checked for all four modules (M0,0 to M0,3). A 'FaxAgent Utilization: 0 %' indicator and a 'Save' button are located above the table. The left sidebar contains links for 'Network Settings', 'Global Settings', 'Domain Settings', 'Domain Management', 'Network Settings' (highlighted), 'Utilities', and 'Help & Support'. The bottom of the interface shows a pagination bar with 'Page size: 25' and '4 items in 1 pages'.

Module	Port	Enabled	Send Enabled	Receive Enabled	Tone Detect Enabled
M0,0	0	Yes	☒	☒	☐
M0,1	1	Yes	☒	☒	☐
M0,2	2	Yes	☒	☒	☐
M0,3	3	Yes	☒	☒	☐

5. Finally restart the “FXC3.FaxAgent” with Windows Services Management console.



For more information, refer to [FaxCore eV5 Administrator's Guide](#) and [FaxCore eV5 User Manual](#).