# Introduction

This document contains the integration details as outlined in the “MAC STS Statement of Work” document that covers the integration of the Secure Trading System’s End User Management System “STS” and the Mobile Authentication Corporation’s “MAC” One-time Password “OTP” System. This document covers both the “Fully Integrated Solution” and the “User Registration Solution”.

Note: As defined in the SOW, items will contain a solution indicator where appropriate as follows:

“F” if only needed for the Fully Integrated Solution

“R” if only needed for the User Registration Solution

“B” if needed by both solutions

## MAC’s OTP System

The OTP system is implemented as a set of Web Services running on an IIS server using a “REST” communication protocol over HTTPS. Requests are MAC formatted data either encrypted using ??? or hexadecimal encoding depending on the security on the connection. Responses are formatted XML. Requests and responses are detailed later in this document.

The MAC OTP system has been integrated with the “Secure Ads” system and is capable of sending “Message Ad” in the text message and “Content Ads” / “Enter OTP Ads” in the responses. The Content Ads and Enter OTP Ads are displayed in the web pages being presented to the End User. The Ad format is detailed in the responses where appropriate.

# Reference Documents

* Mobile Authentication Corporation: MAC STS Statement of Work (latest)
* Secure Trading System: Operator Integration Workflow 1\_v7.0.1
* Mobile Authentication Corporation: OTP System Overview
* Secure Ads: related documents (tbs)

# OTP Services

There are four (4) Services that support End User Registration, One-Time Password and send Message functionality, RequestOTP Service and VerifyOTP Service and the Usage / Billing data Service. The base URL (where the Services are running) combined with the Method URL make up the Fully Qualified Domain Name “FQDN” for each service.

Service URIs:

* RequestOTP service: Otp/RequestOTP.asmx/WsRequestOtp
* VerifyOTP service: Otp/ValidateOTP.asmx/WsValidateOtp
* Secure Trading Registration service: /User/StsEndUserRegistration.asmx/WsStsEndUserRegistration
* End User Management service: [/User/EndUserManagement.asmx/WsEndUserManagement](http://corp.mobileauthcorp.com/macservices/User/EndUserManagement.asmx/WsEndUserManagement)
* Usage and billing information service: /AdminServices/UsageBilling.asmx/WsUsageBilling

# MAC Service Request and Response details

## General Format Details for MAC Services Requests

The MAC System service format requests as a string containing key value pairs separated by a pipe “|” character as the *Item* separator. The Key and Values are separated by a colon “:” character as the *Key-Value* Separator.

Note: The keys must be unique.

Note: If values contain a *Key-Value* separator, they must be hexadecimal encoded.

Example: key1:value1|key2:value2|key3:value3….

## Request Format Details

The parameters for a request are assembled in an ASCII string as key value pairs with each key/value separated by the pipe character “|” and the keys are separated from the values by the colon character “:”. Some values, such as the Transaction details may contain special character that would cause problems in the request process. These values must be converted to a hexadecimal string before it is added to the request parameter list. (See the coding examples section for details).

**Note**: The following example is for a “Client Managed End User” request. The client is responsible for supplying the end user’s phone number and email address. MAC’s OTP system does not maintain or verify the end user information. For details of a “Registered End User” request refer to the coding examples.

**Note**: This request has the Ad Pass Opt-out option set to do not send Ad. See the Ad Pass section for more details.

* Example of JavaScript function call

function RequestOtpClientManagedEndUser(

pClientId, // Client Id (required)

pEndUserPhoneNumber, // End user’s phone number (required, format is validated)

pEndUserEmail, // End user’s email address(required, format is validated)

pEndUserIp, // End user’s machine’s IP address (optional)

pTransactionType, // OTP Message type (optional, default is 0)

pTransactionDetails, // Transaction Details (optional, included in OTP message)

pAdPass, // Ad Pass Option (optional), see the Ad Pass section for details

pCallbackFunction)

* Example before hexadecimal encoding:

**Note**: Key value pairs with keys in red and values in blue (key value separator is in black).

Request:SendOtp|CID:5351674c74846919ec735074|PhoneNumber:4802684076|EmailAddress:tdavis@mobileauthcorp.com|EndUserIpAddress:192.168.168.1|TrxType:2|TrxDetails:4861742031372e39397c4a61636b657420243135302e39387c546f74616c20243136382e3937|{AdPass Details}|API:???

Where:

1. The request (required): Request:SendOtp

**Note**: In this example send OTP to a Client managed end user.

1. Client Id (required): <CID:5351674c74846919ec735074>
2. End user’s mobile phone number (required): PhoneNumber:4802684076
   1. Validated using the following regex expression: ^\(?\d{3}\)?-? \*\d{3}-? \*-?\d{4}
3. End user’s email address (required): EmailAddress:tdavis@mobileauthcorp.com
   1. Validated using the following regex expression:^([a-zA-Z0-9\_\-\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([a-zA-Z0-9\-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\]?)$
4. End user’s machine IP address (optional): EndUserIpAddress:192.168.168.1
5. Transaction type (optional default is 0 ‘OTP’): TrxType:2
6. Transaction details (optional, default is no details in OTP message): TrxDetails:4861742031372e39397c4a61636b657420243135302e39387c546f74616c20243136382e3937

**Note 1**: Transaction details (the value) is hexadecimal encoding to avoid issues with special character

**Note 2**: see transaction details encoding for formatting details.

1. AdPassDetails (Optional) if included, See AdPass section, Request Details later in this document.
   1. ApOpt:AdDisable – Example of opting out of an ad for this request.
2. Who is making the request (optional, used for resolving errors): API:???

* Example after hexadecimal encoding (complete data packet):

Data=

* Break down of components:

Http post header: Data=

Hexadecimal encoded indicator: 99

Length of client id: 24

Client id (as issued by MAC): 5351674C74846919EC735074

Request data (Hexadecimal encoded): 

### Transaction Type (TrxType) Encoding

The transaction type parameter is used by the Send OTP function to select the text message formatting template.

The templates are setup in when the client is registered with the system.

Transaction types are:

0: (TrxType:0) is for *notification* messages “no OTP will be generated or included in message”,

1: (TrxType:1) is for *authentication* nominally used in the login process “no transaction details”,

2: (TrxType:2) is for *transaction verification* normally includes transaction details that get passes as a hexadecimal encoded string. The message assembly function decodes and formats based on the template.

### Transaction Details Format and Encoding

The transaction details could contain new lines and characters that can’t be sent as ASCII characters.

1. The new lines in the transaction details and in the send message text must be replaced by the pipe character “|”. The massage assembly logic replaces the pipe character with the appropriate new line sequence for the message delivery channel.
2. The transaction details and the send message body are hexadecimal encoded.

* Example before encoding:

Hat $17.99|Jacket $150.98|Shirt $33.98|Total $202.95

**Note:** The example will be displayed in the OTP message as 4 lines.

## General Format Details for MAC Service Responses

The MAC service responses are XML formatted with pre-defined *Element* names.

Examples: (response to the Request OTP request)

Example of Successful response:

<?xml version="1.0" encoding="utf-8" ?>

<macResponse>

<calledMethod>WsRequestOtp()</calledMethod>

<Reply> Note: Pipe “|” separated fields, Colon “:’ separated key value pairs

RequestId:5446d6637484691328eab102 // Request Id

|TLM:5 // OTP time to live minutes

|DeliveryMethod:Sms // Deliver method (Sms/Email/Voice)

|EnterOTPAd:{Hex Encoded Ad}\* // Optional ad for OTP Page (Hexadecimal encoded)

|ContentAd:{Hex Encoded ad}\* // Optional ad for Content Page (Hexadecimal encoded)

</Reply>

<Details>

Request:SendOtp

|ClientName:Scottsdale Golf Store

|TLM:5

|OTPRetriesMax:3

|OTPExpiredTime:10/21/2014 10:05:49 PM

</Details>

</macResponse>

\* See AdPass section for additional information

Example of error response:

<?xml version="1.0" encoding="utf-8"?>

<macResponse totalProcessTime="1ms">

<calledMethod>FinalizeXmlResponseWithError()</calledMethod>

<Error>Invalid [CID:53ed325e74846912e08d57ad1</Error](CID:53ed325e74846912e08d57ad1%3c/Error)>

</macResponse>

Note: There are several error responses that could be returned by the MAC OTP Services. Each specific to the service and the error.

* Example of STOP Error response:

<?xml version="1.0" encoding="utf-8"?>

<macResponse totalProcessTime="1ms">

<calledMethod>FinalizeXmlResponseWithError()</calledMethod>

<Error>Not sent, Blocked user replied '**STOP**' (FromNumber=ShortCode)</Error>

</macResponse>

**Note**: The STOP error indicates that the end user has replied or sent a STOP to the short code the system is using. The End User will not receive the OTP message until they send the opt-in text “OTTIN” to the short code.

### Element Names

macResponse – Wraps all XML formatted responses

calledMethod – Element in every response

Error – If present the “Error” element contains the error details or code.

Reply – If the response was processed, the “Reply” element contains the status. Success, Failure, Valid, Invalid, etc.

Details – If present the “Details” element contains the text with details of the request processing.

### Fixed Element Values

Success – Value in response to Request OTP indicating that the request was successful.

Sent – Value in response to the Request OTP indicating the OTP was sent.

Validated – Value in response to Verify OTP request indicating that the OTP was validated. Once validated the OTP is disabled.

Invalid – Value in response to Verify OTP request indicating that the OTP was invalid. Details contain retry status, number of retries or disabled.

Timeout - Value in response to Verify OTP request indicating that the OTP has timed out.

Resent – Value in response to the Resend OTP request indicating the OTP message was successfully resent.

Registered – Value in response to the Register End User request indicated the end user was successful registered.

TLM – OTP time to live minutes

# End User Registration

The “End User” registration process will be used in “Both” solutions, however, the calling parameters are different. The following diagram illustrates the flow between the STS system registration process and the MAC custom service developed to register end users into the MAC OTP System. The reference numbers in red are used to reference the details of the requests and responses need to register an End User.

STS End User Registration

Service

Request OTP

Service

Verify OTP

Service

MAC OTP System

Secure Trading System

End

User

Enrollment

Request(Register)

Response(Registered / Failed)

Request(Request OTP)

Response(Request Id, Ad Info / Error)

Request(Verify OTP)

Response(Validated / Invalid)

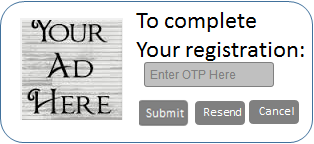


End User Registration

End User Verification

Service

Detail User Information



1

2

3

### Reference Number Details

1. STS Sends “Register End User” request to MAC’s Registration service
   1. Request details
      1. Example:

Request:EndUserRegister|CID:53ebb04d7484691d4ce32a27|FirstName:John|LastName:Doe|PhoneNumber:5555551212|EmailAddress:jdoe@gmail.com|RegType:OpenRegister|UserId:480268407612345678901234567890|API:STS

Note: UserId supplied by STS, must be unique across all groups and clients and users registered by STS.

Note: This example only shows the required fields. The end user registration request supports additional user information, See optional user information section later in this document.

* + - 1. B: Key CID Value 53ebb04d7484691d4ce32a27 – The Id of the client that is registering the end user.
      2. B: Key FirstName Value John – First name of end user.  
         Key LastName Value Doe – Last name of end user (R: used in the generation of user Id).
      3. B: Key PhoneNumber Value 5555551212 – End user’s mobile phone.
      4. B: Key EmailAddress Value doe@gmail.com – End user email (R: used in the generation of user Id).
      5. B: Key RegType Value OpenRegister – Registration Type options are:
         1. ClientRegister – Only client that registered user can use this user.
         2. GroupRegister – Only clients in this group can use this user. Group Id must be present in data.
         3. OpenRegister – User can be used by any client in the OTP system if client is enabled for Open access (client configuration).
      6. F: Key UserId Value 480268407612345678901234567890 – The unique user id generated by the STS system.
      7. B: Key API Value STS – requesting API tag
  1. Response details
     1. Successful response details

Note: If used is already registered the sane response is returned.

* + - 1. Example:

<?xml version="1.0" encoding="utf-8" ?>

<macResponse totalProcessTime='18ms'><calledMethod>WsStsEndUserRegistration()</calledMethod>

<Reply>Registered</Reply>

</macResponse>

* + 1. Unsuccessful response details

Note: There are several other errors that could be returned.

* + - 1. Example: Invalid Client Id

<?xml version="1.0" encoding="utf-8"?><macResponse totalProcessTime="1ms"><calledMethod>FinalizeXmlResponseWithError()</calledMethod>

<Error>Invalid [CID:53ed325e74846912e08d57ad1</Error](CID:53ed325e74846912e08d57ad1%3c/Error)>

</macResponse>

* + - 1. User Exists

<?xml version="1.0" encoding="utf-8"?><macResponse totalProcessTime="0ms"><calledMethod>FinalizeXmlResponseWithError()</calledMethod>

<Error>StsEndUserRegistration End User Exists</Error>

</macResponse>

* + - 1. Invalid Phone Number

<?xml version="1.0" encoding="utf-8"?><macResponse totalProcessTime="0ms"><calledMethod>FinalizeXmlResponseWithError()</calledMethod>

<Error>StsEndUserRegistration , End User's Phone Number is invalid!</Error>

</macResponse>

* + - 1. Invalid Phone Number and Email Address

<?xml version="1.0" encoding="utf-8"?><macResponse totalProcessTime="0ms"><calledMethod>FinalizeXmlResponseWithError()</calledMethod>

<Error>StsEndUserRegistration , End User's Phone Number is invalid!, End User's email is invalid!</Error>

</macResponse>

1. Request OTP

Note: See Request OTP under End User Authentication (Login) later in this document

1. Verify OTP

Note: See Verify OTP under End User Authentication (Login) later in this document

## End User Authentication (Login)

The end user account login uses the “End User Authorization” process in the “Fully Integrated Solution”.

Request OTP

Service

Verify OTP

Service

Secure Trading System

End

User

Login

Request(Request OTP) 1

Response(Request Id, Ad div)

Request(Verify OTP) 2

Response(Validated, Ad Text/Invalid)



End User Authentication



**Welcome**



MAC OTP System

### Reference Number Details

1. Request OTP

The Request OTP is only used in the “Fully Integrated Solution”

* 1. Request details
     1. Example:

Request:SendOtp|CID:53ebb04d7484691d4ce32a27|UserId:480268407612345678901234567890|EndUserIpAddress: 27.132.75.30|TrxType:1|{AdPass Request Options}|API:STS

* + - 1. Key Request Value SendOtp – Request to the Request OTP Service.
      2. Key CID Value 53ebb04d7484691d4ce32a27 – The Id of the client that is registering the end user.
      3. Key UserId Value 480268407612345678901234567890 – The unique user id generated by the STS system.
      4. Key EndUserIpAddress Value 27.132.75.30 End User’s IP address (not the IP address of the client’s web site).
      5. Key TrxType Value 1 – Type of transaction “Authentication” used to select message format.
      6. AdPass Request Options – (Optional) see AdPass, Request Options later in this document.
      7. Key API Value STS – requesting API tag
  1. Response details
     1. Successful response details
        1. Example:

<?xml version="1.0" encoding="utf-8"?>

<macResponse totalProcessTime="841ms">

<calledMethod>WsRequestOtp()</calledMethod>

<Reply>

RequestId:5446d6637484691328eab102

|EnterOTPAd:3C64697620646174612D61642D69643D27456E7465724F54505F4731273E3C61207461726765743D275F626C616E6B2720687265663D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F52656469722E617370783F693D4731273E3C696D67207372633D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F6164732F676F6C662F676F6C662D6164312E6A706727207374796C653D276D61782D77696474683A2033333570782021696D706F7274616E743B77696474683A20313030252021696D706F7274616E743B2720626F726465723D2730273E3C2F613E3C2F6469763E

|ContentAd:3C64697620646174612D61642D69643D27456E7465724F54505F4731273E3C61207461726765743D275F626C616E6B2720687265663D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F52656469722E617370783F693D4731273E3C696D67207372633D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F6164732F676F6C662F676F6C662D6164312E6A706727207374796C653D276D61782D77696474683A2033333570782021696D706F7274616E743B77696474683A20313030252021696D706F7274616E743B2720626F726465723D2730273E3C2F613E3C2F6469763E

</Reply>

<Details>

Request:SendOtp

|ClientName:Client A

|DeliveryMethod:Sms

|TLM:5

|OTPRetriesMax:3

|OTPExpiredTime:10/21/2014 10:05:49 PM

</Details>

</macResponse>

* + - * 1. MAC System Header
        2. Reply

Key RequestId value 5446d6637484691328eab102 – Request Id used as a correlation number for the request.

Key DeliveryMethod Value Sms – How the OTP message was delivered options: Sms, Voice or Email.

Key EnterOTPAd Value 3C64697… – Optional depending on Clients configuration and ad campaign setup. Contains a hexadecimal encoded string to avoid conflicts with imbedded special characters, see details later in this document.

Key ContentAd Value 3C64697… – Optional depending on Clients configuration and ad campaign setup. Contains a hexadecimal encoded string to avoid conflicts with imbedded special characters, see details later in this document.

* + - * 1. Details

Key Request Value SendOtp – Request that sent the OTP

Key ClientName Value Client A – Name of client associated with Client Id of request.

Key TLM Value 5 – Time in minutes the OTP has to live, as configured in the client/ OTP settings.

Note: Resend does reset the timeout.

Key OTPRetriesMax Value 3 – Number of retries attempts the user gets before the OTP is disabled, As configured in the client/OTP settings.

Key OTPExpiredTime Value 10/21/2014 10:05:49 PM – UTC Time OTP will expire

* + 1. Unsuccessful response details
       1. Same format as other services

1. Verify OTP request
   1. Request details
      1. Example:

Request:VerifyOtp|CID:53ebb04d7484691d4ce32a27|RequestId:5446d01f74846913287bf412|OTP:856121|EndUserIpAddress: 192.168.0.5|API:STS

Note: RequestId as sent in Request OTP response.

* + - 1. Key Request Value VerifyOtp – Request to the VerifyOTP service
      2. Key CID Value 53ebb04d7484691d4ce32a27 – The client Id, Note: must be the same as the client id that requested the OTP.
      3. Key RequestId Value 5446d01f74846913287bf412 – As returned in the response of the OTP request.
      4. Key OTP Value 856121 – The OTP entered by the end user.
      5. Key EndUserIpAddress Value 192.168.0.5 – End User’s IP address (not the IP address of the client’s web site).
      6. Key API Value STS – requesting API tag
  1. Verify OTP Response
     1. Successful response details
        1. Example:

<?xml version="1.0" encoding="utf-8"?>

<macResponse totalProcessTime="361ms"><calledMethod>WsValidateOtp()</calledMethod>

<Reply>

Validated Note: see Replies.

<Reply>

<Details>

|RequestId:5446d01f74846913287bf412

|OTPRetriesMax:3|OTPRetriesCurrent:0

|OTPExpiredTime:10/21/2014 10:05:49 PM

|ClientName:Client A

</Details>

</macResponse>

* + - * 1. Replies:

Validated – The OTP was valid and market used.

Invalid – The OTP dis not match the OTP sent, Retry count incremented.

Disabled – OTP disabled because of too many retries.

Timeout – OTP disabled because of timeout.

Inactive – Request Id references an OTP record that is no longer active.

* + - * 1. Details:

Key RequestId Value “5446d01f74846913287bf412” – Request correlation number.

Key OTPRetriesMax Value 3 – Maximum retries.

Key OTPRetriesCurrent Value 0 – Current

Key OTPExpiredTime Value 10/21/2014 10:05:49 PM – UTC Time OTP will expire.

Key ClientName Value Client A– Name of client.

* + 1. Unsuccessful response details
       1. Same format as other services see Error Responses for possible errors.

## Transaction Authorization / Verification

The Transaction Authorization and verification is used in the “Fully Integrated Solution”. The major difference is in the request parameters that include the transaction details.

Request OTP

Service

Verify OTP

Service

Secure Trading System

Request(RequestOTP) 1

Response(Request Id, Ad Text)

Request(Verify OTP) 2

Response(Validated/Invalid, , Ad Info)

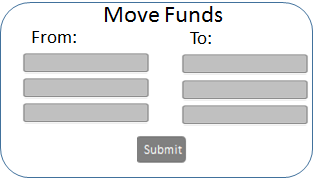


Transaction Authorization / Verification

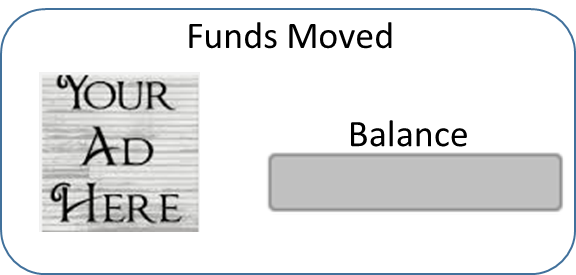
(Funds Movement)

Funds

Movement



Note: Request contains a different transaction type and transaction details



### Reference Number Details

1. Request OTP (Transaction Verification)
   1. Request details
      1. Example: Request:SendOtp|CID:53ebb04d7484691d4ce32a27|UserId:480268407612345678901234567890|EndUserIpAddress:::1|TrxType:2|TrxDetails:4861743A24372E39397C4A61636B65743A243135302E39387C5368697070696E673A24302E30307C546F74616C3A243135382E3937|{AdPass Request Options}|API:STS

Note: UserId, as registered.

* + - 1. Key Request Value SendOtp – Request to the Request OTP Service.
      2. Key CID value 53ebb04d7484691d4ce32a27 – The Id of the client that is registering the end user.
      3. Key UserId Value 480268407612345678901234567890 – The unique user id generated by the STS system.
      4. Key EndUserIpAddress Value 192.168.0.5 – End User’s IP address (not the IP address of the client’s web site).
      5. Key TrxType Value 2 – Type of transaction “Transaction verification and authorization” used to select OTP message format.
      6. Key TrxDetails Value 4861743A2437… – Transaction details are text lines separated by the pipe “|” character, Hexadecimal encoded transaction details to avoid conflicts with special characters
      7. AdPass Request Options – (Optional) if included the client or user is special handling of the AdPass feature for this request, see AdPass, Request Options later in this document.
      8. Key API value STS – requesting API tag
  1. Response details
     1. Successful response details
        1. Example:

<?xml version="1.0" encoding="utf-8"?>

<macResponse totalProcessTime="841ms"><calledMethod>WsRequestOtp()</calledMethod>

<Reply>

RequestId:5446d6637484691328eab102

|TLM:5

|DeliveryMethod:Sms

|EnterOTPAd:3C64697620646174612D61642D69643D27456E7465724F54505F4731273E3C61207461726765743D275F626C616E6B2720687265663D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F52656469722E617370783F693D4731273E3C696D67207372633D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F6164732F676F6C662F676F6C662D6164312E6A706727207374796C653D276D61782D77696474683A2033333570782021696D706F7274616E743B77696474683A20313030252021696D706F7274616E743B2720626F726465723D2730273E3C2F613E3C2F6469763E

|ContentAd:3C64697620646174612D61642D69643D27456E7465724F54505F4731273E3C61207461726765743D275F626C616E6B2720687265663D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F52656469722E617370783F693D4731273E3C696D67207372633D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F6164732F676F6C662F676F6C662D6164312E6A706727207374796C653D276D61782D77696474683A2033333570782021696D706F7274616E743B77696474683A20313030252021696D706F7274616E743B2720626F726465723D2730273E3C2F613E3C2F6469763E

</Reply>

<Details>

Request:SendOtp|ClientName:Client A

</Details>

</macResponse>

* + - * 1. MAC System Header
        2. Reply

Key RequestId value 5446d6637484691328eab102 – Request Id used as a correlation number for the request.

Key DeliveryMethod value Sms – How the OTP message was delivered options: Sms, Voice or Email.

Key EnterOTPAd Value 3C64697620646… – Optional, depending on Clients configuration and ad campaign setup. Contains a hexadecimal encoded string to avoid conflicts with imbedded special characters, see details later in this document.

Key ContentAd Value 3C64697620… – Optional, depending on Clients configuration and ad campaign setup. Contains a hexadecimal encoded string to avoid conflicts with imbedded special characters, see details later in this document.

* + - * 1. Details

Key Request Value SendOtp – Request that sent the OTP

Key ClientName Value Client A – Name of client associated with Client Id of request.

* + 1. Unsuccessful response details
       1. <Error>Details<Error>
          1. See Error Responses for details.

1. Verify OTP request

See Verify OTP request in the User Authentication Section

# End User Management

The End User Management service is provided so that the End User’s status and information can be updated by the client application. Access to the end user record is restricted to the client that registered the end user regardless of how the end user was registered, i.e. Client Registered, Group Registered or Open Registered.

The End User management service allows the registering client to disable, enable and delete the end user. The service also allows the client to change the end user information, such as phone number, address information, etc. There are several parts of the end user information that can’t be changed, any part of the end user’s name and the end user’s email address. In order to change that information the end user would have to be deleted and re-registered with the new information.

1. End User Management
   1. Request details
      1. Requests supported:
         1. **CheckEndUserRegistration**
         2. **ActivateEndUser**
         3. **DeactivateEndUser**
         4. **UpdateEndUser**
         5. **DeleteEndUser**
         6. **SetAdPassOption, AdPassEnable (Opt-in), AdPassDisable (Opt-out)**
      2. Examples: Request:ActivateEndUser|CID:53ebb04d7484691d4ce32a27|UserId:480268407612345678901234567890|API:STS

Request:DeactivateEndUser|CID:53ebb04d7484691d4ce32a27|UserId:480268407612345678901234567890|API:STS

Request:SetAdPassOption|CID:53ebb04d7484691d4ce32a27|UserId:480268407612345678901234567890|

AdPassOption:AdPassDisable|API:STS

Request:UpdateEndUser|CID:53ebb04d7484691d4ce32a27|UserId:480268407612345678901234567890

|OldPhoneNumber:5553886666|PhoneNumber:5553881212|API:STS

Note: UserId, as registered.

* + - 1. Key Request Value UpdateEndUser – Request to Update the end user information.
      2. Key CID value 53ebb04d7484691d4ce32a27 – The Id of the client that registered the end user.
      3. Key UserId Value 480268407612345678901234567890 – The unique user id generated by the STS system.
      4. Key OldPhoneNumber Value 555388666 – (Optional) End User’s old phone number. Logged if included for tracking the change.
      5. Key PhoneNumber Value 5553881212 – End User’s new phone number.
      6. Key API value STS – requesting API tag
    1. Other end user information that can be updated without deleting and re-registering
       1. Primary street address, Street:New Street
       2. Secondary street address, Street2:New Secondary
       3. Unit, Suite, Apartment number, Unit:
       4. City, City:Boston
       5. State, State:MA
       6. ZipCode, ZipCode:12345
       7. Country, Country:US
  1. Response details
     1. Successful response details
        1. Example:

<?xml version="1.0" encoding="utf-8"?>

<macResponse totalProcessTime="841ms"><calledMethod>WsRequestOtp()</calledMethod>

<Reply>

Success

</Reply>

<Details>

Request:SendOtp|ClientName:Client A

</Details>

</macResponse>

* + - * 1. MAC System Header
        2. Reply

Key RequestId value 5446d6637484691328eab102” – Request Id used as a correlation number for the request.

Key DeliveryMethod value Sms – How the OTP message was delivered options: Sms, Voice or Email.

Key EnterOTPAd Value 3C64697620646… – Optional, depending on Clients configuration and ad campaign setup. Contains a hexadecimal encoded string to avoid conflicts with imbedded special characters, see details later in this document.

Key ContentAd Value 3C64697620… – Optional, depending on Clients configuration and ad campaign setup. Contains a hexadecimal encoded string to avoid conflicts with imbedded special characters, see details later in this document.

* + - * 1. Details

Key Request Value SendOtp – Request that sent the OTP

Key ClientName Value Client A – Name of client associated with Client Id of request.

* + 1. Unsuccessful response details
       1. <Error>Details<Error>
          1. See Error Responses for details.

# Usage And Billing Request

The Usage and Billing request calls a MAC OTP System service the creates and XML response containing an element containing the usage numbers and the amount due for each Group and client associated with the request.

1. Usage and Billing request
   1. Request details
      1. Example:

Request:GetUsageBillingForMonth|CID:53ebb04d7484691d4ce32a27|ForMonth:10/2014|API:STS

* + - 1. Key Request Value GetUsageBilling – request to service
      2. Key CID value 53ebb04d7484691d4ce32a27 – If for a single client
      3. Key ForMonth value 10/2014 as a string.
      4. Key GroupId value 53ebb04d7484691d4ce32a27 – If for a group that contains multiple groups or clients.
  1. Response Details
     1. Example response for single client request

<macResponse totalProcessTime="259ms">

<calledMethod>WsGetBill()</calledMethod>

<Reply>

<Client name="!MAC Default Client" id="530f6e8e675c9b1854a6970b">

<DateCreated>2/3/2015 10:50:13 PM</DateCreated>

<DateDue>3/5/2015 10:50:13 PM</DateDue>

<SubTotal>$963,724.71</SubTotal>

<TaxRate>0.0931</TaxRate>

<SalesTax>$89,722.77</SalesTax>

<Total>$1,053,447.48</Total>

<Details>

<OTPSent>

<Sms count="2,546,698" price="$0.08" amount="$203,735.84"/>

<Email count="0" price="$0.00" amount="$0.00"/>

<Voice count="0" price="$0.00" amount="$0.00"/>

</OTPSent>

<Ads>

<Sent count="9,244,212" price="$0.08" amount="$739,536.96"/>

<Clicked count="0" price="$0.00" amount="$0.00"/>

</Ads>

<EndUser>

<Registrations count="206,775" price="$0.08" amount="$16,542.00"/>

<Verifications count="0" price="$0.00" amount="$0.00"/>

</EndUser>

<Addendums count="2" amount="$3,909.91">

<Addendum>

<Amount>$500.00</Amount>

<Notes>Monthly service charge.</Notes>

</Addendum>

<Addendum>

<Amount>$463.04</Amount>

<Notes>$740,000.00 monthly Advertising minimum - adj.</Notes>

</Addendum>

</Addendums>

</Details>

</Client>

</Reply>

</macResponse>

* + 1. Example response for group request

<?xml version="1.0" encoding="utf-8"?>

<macResponse totalProcessTime="361ms"><calledMethod>WsValidateOtp()</calledMethod>

<Reply>

<Group Name=”My Group” GroupId=”5446d01f74846913287bf000”>

<Client Name=“Client 1” Cid=”5446d01f74846913287bf412”>

<OTPSent>

<Sms count=”12345” AmountDue=”10234.00”/>

<Email count=”12345” AmountDue=”10234.00”/>

<Voice count=”12345” AmountDue=”10234.00”/>

</OTPSent>

<Ads>

<Sent count=”123456” AmountDue=”10234.00”/>

<Clicked count=”123456” AmountDue=”10234.00”/>

</Ads>

<EndUser>

<Registrations count”1234” AmountDue=”10234.00”/>

</EndUser>

</Client>

<Client Name=“Client 2” Cid=”5446d01f74846913287bf413”>

<OTPSent>

<Sms count=”12345” AmountDue=”10234.00”/>

<Email count=”12345” AmountDue=”10234.00”/>

<Voice count=”12345” AmountDue=”10234.00”/>

</OTPSent>

<Ads>

<Sent count=”123456” AmountDue=”10234.00”/>

<Clicked count=”123456” AmountDue=”10234.00”/>

</Ads>

<EndUser>

<Registrations count”1234” AmountDue=”10234.00”/>

</EndUser>

</Client>

</Group>

</ Reply >

</macResponse>

# Optional User Information

As an option the end user registration request supports the passing of additional user information. This data will get recorded but has no use cases in the “Fully Integrated Solution”. Should be supplied in the “End User Registration Solution”.

1. Prefix – Name prefix
2. MiddleName – Middle name
3. Suffix – Suffix
4. DOB – Date Of Birth
5. SSN4 – Last 4 digits of SSN
6. Street – First part of street address
7. Street2 – Second part of street address
8. Unit – House number, suite number, apartment number
9. City – Name of city
10. State – 2 digit state code
11. Country –Country name or code
12. ZipCode – 5 digit zip code

# AdPass

AdPass is an feature that allows ads to be delivered to the end user and part of the OTP solution.

The client can configure Ad Campaigns in the Ad System and specify when and how the ads can be delivered and how they are displayed to the user. In the context of this document only the “EnterOTPAd” and the “ContentAd” will be discussed in detail as that is all STS will have to allow for in the integration of the “Fully Integrated Solution”.

## AdPass Request Options

The AdPass feature supports several “On Request Options”. These options are supplied by the client on each request. The MAC OTP System does not maintain these option on a per user basis. The Option are as follows:

1. User or Client Opt-out – This allows the the client to configure a user opt-out feature. If not present in the request the ads are enabled:
   1. Key ApOpt Value AdDisable – The MAC OTP system will not send an Ad for this request regardless of client’s configuration.
2. Ad selection parameters – the Ad Server accepts several Ad Selection Parameters that help the Ad Server select the correct ads to be delivered to the end user. The following is a list of Ad Selection Parameters:
   1. AdNumber
   2. AdAge
   3. AdCity
   4. AdEthnicity
   5. AdGender
   6. AdHomeowner
   7. AdHouseholdIncome
   8. AdMaritalStatus
   9. AdState
   10. AdUserIp
   11. AdType

## Ad Response Details

The Enter OTP and Content Ads are delivered from the ad server as text. The format is HTML and formatted as a ‘div’ that can be inserted directly into the page being delivered to the client. The MAC OTP system encodes the text as a hexadecimal string to avoid conflicts created by special characters used in the ads.

Example of an Ad: - Note: the content of the ad “div” may change based on the Ad Campaign setup:

<div data-ad-id='EnterOTP\_G2'><a id='adURL' target='\_blank' href='http://localhost:8010/demos/Redir.aspx?i=G2' ><img src='http://localhost:8010/demos/ads/golf/golf-ad2.jpg' style='max-width: 300px !important;width: 100% !important;' border='0'></a></div>

# Error Responses

There are 4 categories for errors, If the response contains a <Error> element, this element will contain the details.

1. Protocol errors – the service could not process the data or the data was corrupted.
   1. Corrupt request data, the service could not decode request.
   2. Request type required, the service could not find the request in the data after decoding.
2. API errors
   1. Invalid request, the service does not support the request, normally caused by the request being sent to the wrong service
   2. Invalid Client Id, the service could not find the client using the Id supplied in the request data
   3. Invalid Client name, the service could not find the client using the client’s name supplied in the request data. Normally caused by a difference in spelling or case.
   4. Invalid Group Id, the service could not find the group using the Id supplied in the request data.
   5. Invalid Group name, the service could not find the group using the group’s name supplied in the request data. Normally caused by a difference in spelling or case.
   6. RequestId (Invalid), service could not find the OTP record in the database using the supplied request id. Normally caused be the requester using an invalid request id.
   7. Invalid end user, Service could not find the end user using the parameters supplied in the request data. Normally caused by a difference in spelling in the unique user information(such as email), incorrect UserId supplied in the request.
   8. No End User to deleted, Normally caused by a difference in spelling in the unique user information(such as email), incorrect UserId supplied in the request.
   9. RequestId (Invalid), service could not find the OTP record in the database using the supplied request id. Normally caused be the requester using an invalid request id.
   10. Disabled, the request id supplied in the request data is for an OTP record that was already used. Normally caused be the requester using an invalid request id.
   11. "Client Id (Invalid for this OTP), Normally caused by the requester using a client id that was different than the client id used in the request OTP.
   12. OTP missing in request data, Normally caused by the requester.
   13. OTP can't be of zero length, Normally caused by the requester.
   14. Stop Error, Not sent, Blocked user replied '**STOP**' (FromNumber=Short Code), Normally caused by end user replying or texting “STOP”.
3. System Errors
   1. Exceptions (General), The service failed when processing the request. Unplanned service issue, report to MAC for a resolution.
4. Normal Responses, The <Reply> element contains Success if operation was successful, The <Details> element contains the details of the response:
   1. End User Exists, The UserId was not unique and the user is already in the database.
   2. Invalid OTP, Normally caused by the user entered an incorrect OTP,
      1. <Reply>Invalid
      2. <Details>|RequestId:5474b0ec74846901f8540c74|OTPCode:060021|OTPRetriesMax:3|OTPRetriesCurrent:2|OTPExpiredTime:11/25/2014 4:50:12 PM|ClientName:The Client
   3. Timeout, the user waited too long to enter the OTP. Note: the time allotted to enter the OTP is in the details of the reply of the RequestOTP response.

# Text Messaging and Short Code Considerations

The carriers and gateways (aggregators) have several restrictions they place on companies that use the text messaging service to send text message to a end user base. This section covers the ones that are specific to sending One Time Passwords.

## Reply messages

Reply messages are the text messages that the End User sends from their mobile device to the system via the system’s short code. There are only 2 reply messages that the OTP system processes the “HELP” message and “STOP” message all other text messages are ignored.

### HELP Reply message

When the user replies or sends a HELP message to the short code the gateway will respond with a canned help message. This message must contain a 1-800 number and a link to a web site that provides online help that contains instruction on how to use the OTP system. When using a MAC provided short code MAC provides these services.

### STOP Reply Message

When the end user replies or send a STOP message to the short code the gateway puts a “Block” on the end user’s number. The OTP system will not be able to send a text message to number from the short code until the block is removed. To remove the block the end user must text OPTIN to the short code. The Request OTP service replies with a STOP error when a “SendOTP” or “SendMessage” request is received for a number that is blocked because the gateway received a “STOP” message.

When the Client receives a STOP error the end user should be told that they have to text “OPTIN” to the short code before the system can deliver an OTP message. See the “Example of STOP Error response:” on page 4.