# 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

## STS User Management System

TBD

## 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: Otp/RequestOTP.asmx/WsRequestOtp

VerifyOTP: Otp/ValidateOTP.asmx/WsValidateOtp

Secure Trading Registration Service: /User/StsEndUserRegistration.asmx/WsStsEndUserRegistration

Usage and billing information: /Otp/UsageBillingService.asmx/WsUsageBillingService

# 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:JS

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
3. End user’s email address (required): EmailAddress:tdavis@mobileauthcorp.com
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.
2. Who is making the request (optional, used for resolving errors): API:JS

* 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>

<Action>Sent</Action>

<RequestId>5351a84074846919d8f97e17</RequestId>

<DeliveryMethod>SMS</DeliveryMethod>

</Reply>

</macResponse>

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.

### 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.

Action – The “Action” element contains the verb defining the action taken on the request.

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.

# 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)

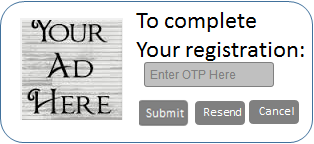


End User Registration

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” – Identifies the calling system in this case STS.
  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
   1. The Request OTP is only used in the “Fully Integrated Solution”
      1. Request details
         1. Example:

Request:SendOtp|CID:53ebb04d7484691d4ce32a27|UserId:480268407612345678901234567890|EndUserIpAddress:::1|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 “:::1” 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) if included the client or user is special handling of the AdPass feature for this request, see AdPass, Request Options later in this document.
        7. Key “API” Value “STS” – Id of the requesting system
    1. Response details
       1. Successful response detail
          1. Example:

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

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

<Reply>RequestId:5446d6637484691328eab102|DeliveryMethod:Sms|EnterOTPAd:3C64697620646174612D61642D69643D27456E7465724F54505F4731273E3C61207461726765743D275F626C616E6B2720687265663D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F52656469722E617370783F693D4731273E3C696D67207372633D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F6164732F676F6C662F676F6C662D6164312E6A706727207374796C653D276D61782D77696474683A2033333570782021696D706F7274616E743B77696474683A20313030252021696D706F7274616E743B2720626F726465723D2730273E3C2F613E3C2F6469763E|ContentAd:3C64697620646174612D61642D69643D27456E7465724F54505F4731273E3C61207461726765743D275F626C616E6B2720687265663D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F52656469722E617370783F693D4731273E3C696D67207372633D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F6164732F676F6C662F676F6C662D6164312E6A706727207374796C653D276D61782D77696474683A2033333570782021696D706F7274616E743B77696474683A20313030252021696D706F7274616E743B2720626F726465723D2730273E3C2F613E3C2F6469763E </Reply>

<Details>SendOtp|Scottsdale Golf Store|Sms</Details>

</macResponse>

Note: Ad text is hexadecimal encoded to avoid special character conflicts, see section on Ad text details.

* + - * 1. Reply
        2. Key “RequestId” value “5446d6637484691328eab102” – Request Id used as a correlation number for the request.
        3. Key “DeliveryMethod” value “Sms” – How the OTP message was delivered options: Sms, Voice or Email.
        4. Key “EnterOTPAd” Value – “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.
        5. Key “ContentAd” Value – “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.
        6. Details: SendOtp|Scottsdale Golf Store|Sms
      1. Unsuccessful response details
         1. Invalid Request

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

<Error>RequestOtp Invalid request for this service[SendOtpabc]</Error>

</macResponse>

* + - * 1. Invalid End User

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

<Error>RequestOtp Open, OpenEndUserServices invalid end user</Error>

</macResponse>

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

Request:VerifyOtp|CID:53ebb04d7484691d4ce32a27|RequestId:5446d01f74846913287bf412|OTP:856121|EndUserIpAddress:::1|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 must be the same as the client 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 “:::1” End User’s IP address (not the IP address of the client’s web site).
      6. Key “API” Value “STS” – Id of the requesting system
  1. Verify OTP Response
     1. Successful response details
        1. Example:

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

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

<Details>|RequestId:5446d01f74846913287bf412|OTPRetriesMax:3|OTPRetriesCurrent:0|OTPExpiredTime:10/21/2014 10:05:49 PM|ClientName:Scottsdale Golf Store</Details>

</macResponse>

* + - 1. Details:
         1. Key “RequestId” Value “5446d01f74846913287bf412” – Request correlation number.
         2. Key “OTPRetriesMax” Value “3” – Maximum retries.
         3. Key “OTPRetriesCurrent” value “0” – Current
         4. Key “OTPExpiredTime” value “10/21/2014 10:05:49 PM” – Time OTP will expire.
         5. Key “ClientName” value “Scottsdale Golf Store” – Name of client.
    1. Unsuccessful response details
       1. Same format as other service.

## 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:192.168.0.5|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 “192.168.0.5” 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) if included the client or user is special handling of the AdPass feature for this request, see AdPass, Request Options later in this document.
      7. Key “API” Value “STS” – Id of the requesting system
  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|DeliveryMethod:Sms|EnterOTPAd:3C64697620646174612D61642D69643D27456E7465724F54505F4731273E3C61207461726765743D275F626C616E6B2720687265663D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F52656469722E617370783F693D4731273E3C696D67207372633D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F6164732F676F6C662F676F6C662D6164312E6A706727207374796C653D276D61782D77696474683A2033333570782021696D706F7274616E743B77696474683A20313030252021696D706F7274616E743B2720626F726465723D2730273E3C2F613E3C2F6469763E|ContentAd:3C64697620646174612D61642D69643D27456E7465724F54505F4731273E3C61207461726765743D275F626C616E6B2720687265663D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F52656469722E617370783F693D4731273E3C696D67207372633D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F6164732F676F6C662F676F6C662D6164312E6A706727207374796C653D276D61782D77696474683A2033333570782021696D706F7274616E743B77696474683A20313030252021696D706F7274616E743B2720626F726465723D2730273E3C2F613E3C2F6469763E </Reply>

<Details>SendOtp|Scottsdale Golf Store|Sms</Details>

</macResponse>

Note: Ad text is hexadecimal encoded to avoid special character conflicts, see section on Ad text details.

* + - * 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 – “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 – “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.

Details: SendOtp|Scottsdale Golf Store|Sms

* + 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 must be the same as the client 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” – Id of the requesting system
  1. Verify OTP Response
     1. Successful response details
        1. Example:

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

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

<Details>|RequestId:5446d01f74846913287bf412|OTPRetriesMax:3|OTPRetriesCurrent:0|OTPExpiredTime:10/21/2014 10:05:49 PM|ClientName:Scottsdale Golf Store</Details>

</macResponse>

* + - * 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” – Time OTP will expire.

Key “ClientName” value “Scottsdale Golf Store” – Name of client.

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

## 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)

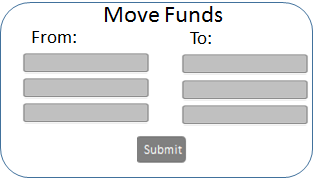


Transaction Authorization / Verification

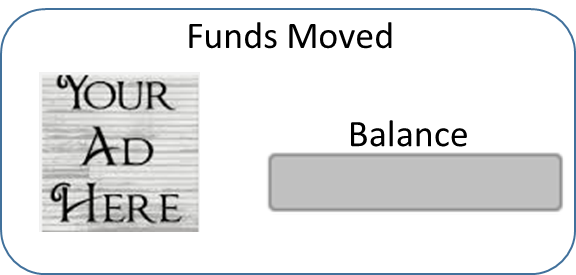
(Funds Movement)

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 “:::1” 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 is Transaction details are text lines separated by the pipe “|” character and 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” – Id of the requesting system
  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|DeliveryMethod:Sms|Ad:3C64697620646174612D61642D69643D27456E7465724F54505F4731273E3C61207461726765743D275F626C616E6B2720687265663D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F52656469722E617370783F693D4731273E3C696D67207372633D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F6164732F676F6C662F676F6C662D6164312E6A706727207374796C653D276D61782D77696474683A2033333570782021696D706F7274616E743B77696474683A20313030252021696D706F7274616E743B2720626F726465723D2730273E3C2F613E3C2F6469763E|ContentAd:3C64697620646174612D61642D69643D27456E7465724F54505F4731273E3C61207461726765743D275F626C616E6B2720687265663D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F52656469722E617370783F693D4731273E3C696D67207372633D27687474703A2F2F6C6F63616C686F73743A383031302F64656D6F732F6164732F676F6C662F676F6C662D6164312E6A706727207374796C653D276D61782D77696474683A2033333570782021696D706F7274616E743B77696474683A20313030252021696D706F7274616E743B2720626F726465723D2730273E3C2F613E3C2F6469763E </Reply>

<Details>SendOtp|Scottsdale Golf Store|Sms</Details>

</macResponse>

Note: Ad text is hexadecimal encoded to avoid special character conflicts, see section on Ad text details.

* + - * 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 – “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 – “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.

Details: SendOtp|Scottsdale Golf Store|Sms

* + 1. Unsuccessful response details
       1. TBS

1. Verify OTP request
   * 1. Successful response details
        1. Example:

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

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

<Details>|RequestId:5446d01f74846913287bf412|OTPRetriesMax:3|OTPRetriesCurrent:0|OTPExpiredTime:10/21/2014 10:05:49 PM|ClientName:Scottsdale Golf Store

</Details>

</macResponse>

* + - * 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” – Time OTP will expire.

Key “ClientName” value “Scottsdale Golf Store” – Name of client.

* + 1. Unsuccessful response details
       1. TBS

# 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:GetUsageBillingDateRange|CID:53ebb04d7484691d4ce32a27|StartingDate:10/20/2014|EndingDate:10/20/2014|API:STS

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 “StartingDate” value “10/20/2014” as a string.
      4. Key “EndingDate” value “10/20/2014” as a string.
      5. Key “ForMonth” value “10/2014” as a string.
      6. Key “GroupId” value “53ebb04d7484691d4ce32a27” – If for a group that contains multiple groups or clients.
  1. Response Details
     1. Example response for single client request

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

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

<Details>

<Client> Name=“Scotts Golf Shop” Cid=”5446d01f74846913287bf412”>

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

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

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

</Client>

</Details>

</macResponse>

* + 1. Example response for group request

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

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

<Details>

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

<Client Name=“Scotts Golf Shop” Cid=”5446d01f74846913287bf412”>

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

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

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

</Client>

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

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

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

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

</Client>

</Group>

</Details>

</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. CoName – Company Name
5. DOB – Date Of Birth
6. SSN4 – Last 4 digits of SSN
7. Street – First part of street address
8. Street2 – Second part of street address
9. City – Name of city
10. State – 2 digit state code
11. Country –Country name or code
12. ZipCode – 5 digit zip code
13. DriverLic – Driver’s License number
14. DriverLicSt – State code where driver’s license was issued

# 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>