The 8MS API Guide

Issue 31.0

Contents

Contents

8MS API Overview

Somos has released a new API, known as the Somos Registry API, intended to replace their legacy MGI interface. 8MS was built on the MGI interface. We are in the process of converting 8MS API calls from MGI-based to Registry-based calls.

*If you are an existing 8MS API customer, please be sure to read the **Somos Registry Impact** section of this guide as you may be impacted by these changes.*

The 8MS API allows application programs to generate messages to SMS/800, and receive replies. The API provides a means to automate SMS/800 provisioning over the MGI, without the need of extensive MGI knowledge. An application program initiates a request of the 8MS API, which then sends the appropriate MGI message or messages. Responses to the messages are gathered up, and presented as one result to the caller. The application program then interprets the result, and takes whatever action is appropriate in its environment.

Calls to the 8MS API are the form of HTTP requests, sent via TCP/IP, to a web server running the 8MS application. The URL defining the HTTP request will contain the name of the appropriate server, and its *post data* will define the parameters of the request. Minimally, the type of the request will be provided, along with access information (user ID and password). Other information relevant to a particular request type may also be provided. These *request parameters* will vary from request to request.

The result of an 8MS API request is one or more responses from SMS/800. The SMS response messages are gathered up and returned in the form of an XML document.

8MS API requests are grouped into the following categories:

Message Requests

Batch Message Requests

Carrier Express Requests

ROC (Resp Org Change) Requests

Miscellaneous Requests

Message requests are those involving a single toll-free number. Batch message requests are those that allow the same operation to be performed on arbitrarily large groups of numbers. The Miscellaneous section contains requests that do not result in messages to SMS/800.

Each of the sections listed above contain a description of the request and the parameters it requires. After reading the details below, which describe how the API functions regardless of request type, you can reference the sections above to discover how to construct any of the 8MS API request calls.

*Note that all required parameters are marked with an asterix (*) character to the left of the parameter name. In the following example **dn** is a required field while **ed** is not.*

*	**dn**	The number to be queried.
	ed	The effective date of the record to be queried.

Interface Details

8MS API service is provided by a Java *servlet* running within the web server that also supports the 8MS GUI. This servlet is accessed by passing one or more of the following parameters as part of an HTTP request. The URL of this request will be of the form **https://***8msServer.8msweb.com***/8ms/api**, where *8msServer* is the name of the 8MS web server machine.

request	The type of request; the possible values are those listed in the three sections described above.
userid	The login of the person making the request (for authentication and logging purposes).
loguserid	This parameter works in conjunction with the **userid** parameter for activity logging purposes. It allows API calls to have a client-application id (any string) associated with the userid in the activity log. A user interface user may search in the **Activity Log** (in the **User Id** field) by either the **userid** or **loguserid**. Additionally, each API entry in the the **Activity Log** screen shows the **userid** followed by the **loguserid** in parentheses.
password	The password of the person making the request (for authentication purposes).
reqparams	A set of semicolon-separated, name/value pairs that define the parameters for the particular request being made. Name and value are separated by the tilde (**~**) character.
timeout	Number of seconds to wait for a response. The default value is 30 seconds.
requestid	In the event of an incomplete response due to a timeout, a requestid is returned to the user. The user may resubmit the request with this requestid. See **Handling Timeouts** below.
routeid	An SMS/800 *route ID*, used to retrieve unsolicited messages. See **Reading Unsolicited Messages** below.

The following is a simple example of an 8MS API URL. This is a request to search SMS/800 for 5

spare toll-free numbers:

```
https://web.8mscorp.com/8ms/api?userid=jsmith& password=bbbbbb&
request=Search&reqparams=count~5
```

Generating Requests

Using HTTP GET rather than POST Is An Extremely Bad Idea!

Developers coding to the 8MS API should always use POST, not GET, for submitting requests. Additionally, all data sent in the POST should be included in the POST body, not as part of the URL.

There are 2 primary reasons for this:

1. *GET is insecure.*

   Any data specified as part of the URL is sent as clear text. This enables anyone to grab your login and password.

2. *GET has size limitations.*

   Depending on the language or tool being used to send requests, GET often has a maximum size limit for the number of characters specified in the URL.

When data for a POST is included in the POST body, your request will be secure and you will not run into size limitations.

Use POST.

The following is sample Java code fragment that illustrates how to use the **URL** and **URLConnection** classes to access the 8MS API. This request will reserve two 877 numbers, waiting up to 20 seconds for a response, with **XML** as the response format.

```
String line;
String urlString = "https://web8ms.corp.com/8ms/api";
String postData = "userid=bso&password=bbbbbb&" +
            "timeout=20&request=Reserve&" +
            reqparams=npa~877;count~2;cname~BSO;" +
            "cphone~7323020222";

try
{
                            // Make the connection.

    URL url = new URL(urlString);

    HttpURLConnection conn =
            (HttpURLConnection) url.openConnection();
    conn.setDoOutput(true);

                            // Post the data.

    OutputStreamWriter wr =
            new OutputStreamWriter(conn.getOutputStream());

    wr.write(postData);
    wr.flush();
    wr.close();
```

```
                                // Get the results.
    conn.connect();

    BufferedReader br =
            new BufferedReader(
                    new InputStreamReader(conn.getInputStream()));

    while ( (line = br.readLine()) != null)
            System.out.println(line);
}
catch (Exception ex)
{
    System.out.println(ex);
}
```

*Note that all data in the URL must be encoded as specified in **RFC 1738** published by the **Internet Engineering Task Force** (IETF). Newer versions of Java provide a java.net.URLEncoder class to properly encode a URL.*

Request Results

When a response to a request is received from SMS/800, the results will be returned to the caller in XML format. For example:

```
<API8MS>
<reqid>OA01417659</reqid>
<message>
 <routeid>RSR</routeid>
<termrpt>COMPLD</termrpt>
<errorcd>0</errorcd>
<msgparams>
ID=ART01000
RO=ART01
CNT=02
NUM=8774328699
NUM=8774328701
</msgparams>
</message>
</API8MS>
```

The following are the tags that appear in the XML response:

<API8MS>	The entire response.
<error>	If request generation fails, or the request does not complete within the given timeout, this tag will contain a message from the 8MS API.
<reqid>	If request succeeds, this tag will contain a request ID. This request ID can be used to subsequently determine when the request completes, and to obtain the results (more on this below).
<message>	A single message in the response (see below).
<routeid>	The type of the SMS/800 response message as defined by the MGI.
<termrpt>	The termination report from SMS/800; either **COMPLD** or **DENIED**.
<errorcd>	The SMS/800 error code; **0** for COMPLD messages, or an error code for DENIED messages.

<msgparams>	The contents of the SMS/800 response; space-separated name/value pairs. The tags are defined by the MGI.

Requests that result in more than one response from SMS/800 will produce an **XML** document with multiple **<message>** tags. For example, searching for 30 877 numbers might produce:

```
<API8MS>
<reqid>OA00453294</reqid>
<message>
<routeid>RSR</routeid>
<termrpt>COMPLD</termrpt>
<errorcd>0</errorcd>
<msgparams>
ID=ART01000
RO=ART01
CNT=10
NUM=8772340467
NUM=8772340525
NUM=8772340534
NUM=8772340535
NUM=8772344868
NUM=8772348692
NUM=8772348728
NUM=8772348824
NUM=8772348926
NUM=8772349726
</msgparams>
</message>
<message>
<routeid>RSR</routeid>
<termrpt>COMPLD</termrpt>
<errorcd>0</errorcd>
<msgparams>
ID=ART01000
RO=ART01
CNT=10
NUM=8772700406
NUM=8772700470
NUM=8772700544
NUM=8772700586
NUM=8772700593
NUM=8772700711
NUM=8772700733
NUM=8772700803
NUM=8772700825
NUM=8772700830
</msgparams>
</message>
<message>
<routeid>RSR</routeid>
<termrpt>COMPLD</termrpt>
<errorcd>0</errorcd>
<msgparams>
ID=ART01000
RO=ART01
CNT=10
NUM=8772693069
NUM=8772693070
NUM=8772693071
NUM=8772693085
NUM=8772693130
NUM=8772693136
NUM=8772693222
NUM=8772693230
NUM=8772693301
NUM=8772693302
</msgparams>
</message>
</API8MS>
```

Handling Timeouts

If a request does not complete within the given timeout period, the following will be returned:

```
<API8MS>
<error>Information Incomplete</error>
<reqid>OA01921131</reqid>
</API8MS>
```

The user can then use the returned request ID to check if all responses have been received For example, submitting a subsequent the following will determine if the given request has completed:

```
userid=bso&password=bbbbbb&requestid=OA1921131
```

If the request has since completed, the results will be returned. If the request is still in progress, the **Information Incomplete** message defined above will again be returned.

Request Generation Failure

If an error occurs in trying to create the request, the following XML document will result:

```
<API8MS>
<error>Request Generation Failed</error>
</API8MS>
```

Generation of requests may fail for a number of reasons, such as bad/missing request type, bad/missing reqparams, etc.

Reading Unsolicited Messages

The 8MS API interface provides a way to retrieve unsolicited messages from the 8MS database. Unsolicited messages are generated by SMS/800 to signal certain events not related to a particular request. Two examples are customer record activations and RespOrg changes.

The following query string will look for messages with a route ID of **URC** (an unsolicited RespOrg change message):

```
userid=bso&password=bbbbbb&routeid=URC
```

If any messages with the given route ID exist, they are returned, and deleted from the 8MS database. Each message is represented as described above, except that unsolicited messages have neither a termination report or error code. The following is a sample XML document that might result from a query for **URC** messages:

```
<API8MS>
<message>
<routeid>URC</routeid>
<msgparams>
RO=ART01
NUM=8005418128
```

```
OLDRO=ALN01
NEWRO=ART01
STAT=WORKING
</msgparams>
</message>
<message>
<routeid>URC</routeid>
<msgparams>
RO=ART01
NUM=8776287250
OLDRO=ALN01
NEWRO=ART01
STAT=WORKING
</msgparams>
</message>
<message>
<routeid>URC</routeid>
<msgparams>
RO=ART01
NUM=8663674709
OLDRO=CWC02
NEWRO=ART01
STAT=WORKING
</msgparams>
</message>
</API8MS>
```

Syntax and Validation

The current 8MS API does very little error checking. The URL passed must contain a valid user ID and password, as well as a known API request type. Failing either of these cases will result in an error message as listed below.

Validation of parameters specific to each request type are not currently part of the API. Values will be passed on to SMS/800, where error detection will take place. Thus, bad input to a particular API request will not result in an error from the 8MS API, but rather a reponse message from SMS/800 containing error reports.

The following are some general guidelines for data sent through the 8MS API:

- Toll free numbers should always be provided as 10 numeric digits (i.e., they cannot contain formatting characters such as dashes or parens).
- Except where noted, the date and time format used by the API is the same as the SMS/800 effective date/time format. An example of this format might be: **10/15/04 10:15A**.
- Generally speaking, the case of the values for API request parameters is irrelevant; SMS/800 will translate everything to upper case.

Error Messages

The following are the error messages that may be produced by the 8MS API:

Unable to get database handle: The database connection failed. Should not happen.

Database Error: A database error occurred. The database connection failed. Should not happen.

Incomplete Request: Required information in the request was missing (e.g., user name and password, request type).

Invalid Request: The request type is not known.

User Access Denied: The given user name/password combination is not valid.

Information Incomplete: The request timed out.

No Active Transactions for Request ID: Results for the given request were already retrieved.

No Unsolicited Messages for Route ID: No unsolicited messages exist for the given route ID.

Request Generation Failed: 8MS failed to generate the request.

Resource Management

This section discusses resource management when coding to the 8MS API.

Sharing Resources with Your Company's Users

The 8MS user interface should be viewed as an API client operating at the same level as any API client that you develop. Whenever a user submits a request, the user interface generates the request via the 8MS API. The user interface then waits, by default, for 30 seconds for the request to be processed. In the case of the user interface, the vast majority of the time, users are actively waiting to see what happens as they need to know as soon as possible if their request succeeded, so the 30 second delay is reasonable and necessary.

The 8MS API supports polling to see if a request was successful, so it is not necessary to wait for a response. When an 8MS API client submits a request, that client must support polling because there is always a chance that the request could time out. (See **Coding for Somos**, below, for more information.) Resources such as database connection are tied up from the time the request is submitted until the response is sent back. Just like the user interface, the API defaults to waiting 30 seconds for a request to complete. This means that all these resources are tied up for as long as 30 seconds. Since polling code must be written for the possibility of a time out, it is in your company's best interest to design your API client to support polling for all cases.

To keep any single company from tying up too many resources, 8MS does not allow a company to have more than four simultaneous API requests. Designing your API client to start up hundreds or thousands of threads and allowing those threads to connection to 8MS with requests of a single number will not speed up your processing time, they will slow it down. Large numbers of connections that 8MS must process only to determine that they have exceeded the per-company limit is just busy work and will result in the following error.

```
<error>Unable to get company connection<error>
<error>Access Denied: Unable to get company conn<error>
```

Since the maximum-of-four is a per-company restriction, and since the user interface also submits requests via the API, the API user shares these resources with user interface users. If the API client always has four requests being submitted, user interface users are unable to submit their requests, effectively preventing your own users from getting their work done. To avoid locking out your users, you should uses the **Batch Message Requests** rather than single number API calls.

API clients should only submit one request at a time.

Batch Benefits

The primary benefit of batch calls is that the user can let the 8MS batch manager process large requests efficiently and effectively, ensuring that resources are tied up for the barest minimal amount of time. However, to reduce locking of resources you must set the API *timeout* parameter is set to 0. Your API client can then poll for completed responses. Polling will only ever return completed responses, so the API client can quickly determine what requests are complete. If results exist, then they will be returned immediately. As discussed above, in the section **Handling Timeouts**, when a timeout occurs, you will see a response similar to the following:

```
<API8MS>
<error>Information Incomplete<error>
<reqid>OA01921131<reqid>
<API8MS>
```

The Information Incomplete indicates that the request did not complete and you should poll for results.

> *Set API timeout to 0 and use the polling mechanism to check back for completed batches.*

To take advantage of 8MS batch management and to minimize resource usage, API clients should seek to put large amounts of toll-free numbers into a single batch request. 8MS can easily support requests with 50,000 numbers, or more, in a single batch.

8MS also limits the number of batches that any one company can submit. It is strongly recommended that API clients limit the current number of batches to around 20. By creating large batches, it is less likely your company will need to exceed 20 batches at any one instant.

> *Submit single large batches rather than many small (1-10 numbers) batches.*

Coding for Somos Downtimes

Somos typically has one downtime per month and that downtime can range from 1 or 2 hours up to 12 or more. During that downtime, Somos will not respond to any messages, so 8MS effectively enters a holding state in which it tries to connect and send messages, fails, waits, and then tries again. No messages you sent to 8MS will be lost during that time. However your API design must take into account that, at some times, it can be many hours before responses are returned. This is one more reason why you should set the *timeout* parameter to 0 and use the polling mechanism. To avoid flooding 8MS with polling requests that will continually return an *Information Incomplete* response, allow some time to pass between polling requests.

> *Poll for results roughly every 60 seconds and specify a timeout of 5 seconds on polling API calls.*

There is one more item you should take into consideration in regards to the Somos system. Most 8MS API developers begin their work in an 8MS test platform, which connects to a Somos Sandbox (test) platform. Somos production is orders of magnitude faster than their sandbox. Do not base your expected performance on how the Sandbox behaves.

Summary

If you follow the above recommendations, you will find that user interface users will have a better

experience and your API client will get overall better performance. To summarize, you can optimize your use of API Calls by the following:

1. *Group as many TFNs as possible in a single batch request.*
2. *Make the Batch API call with a timeout of 0 so resources are locked up for the least amount of time.*
3. *Only submit one request at a time; avoid submitting parallel requests.*
4. *Poll for results every 60 seconds and specify a timeout of 5 seconds on polling API calls.*

Polling Example

The remainder of this section provides an example of submitting a request designed for polling.

1. Submit a POST request.

```
URL
https://test.8msweb.com/8ms/api

POST Data
userid=apiuser&password=apipwd& timeout=0&
request=BatchCopyRecord&
reqparams=source~8554617560;
dnlist~8554617561,8554617562,8554617563; destDateTime~04/01/21
08:00A; sourceDateTime~03/03/21 08:00A
```

2. Receive a timeout (Information Incomplete) response.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE API8MS>
<API8MS>
<error>Information Incomplete</error>
<reqid>OA89ZWZ06H</reqid>
<API8MS>
```

3. Submit a poll request using the returned *reqid* in the above *Information Incomplete* response.

```
https://test.8msweb.com/8ms/api
userid=apiuser&password=apipwd&timeout=0&requestid=
OA89ZWZ06H
```

4. When a response other than *Information Incomplete* is returned, your request has completed and you may process the response.

   Note that an incomplete requestid may be polled an unlimited number of times, but when a completed requestid has been polled, 8MS removes the response details. It is the API

client's responsibility to retain this data at this point.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE API8MS>
<API8MS>
<reqid>
OA89ZWZ04G
<reqid>
<message>
<routeid>
TXU
<routeid>
<termrpt>
COMPLD
<termrpt>
<errorcd>
0
<errorcd>
<msgparams>

detailed response removed for clarity

<msgparams>
</message>
</API8MS>
```

Transition from Somos MGI to Somos Registry

In mid-2019, Somos replaced their SMS/800 system with a completely rewritten system known as the SMS/800 TFN Registry. As part of that new system, a new TFN Registry API was introduced which is available alongside the legacy MGI API that all clients of SMS/800 have been using since the 1990s. In October of 2019, Somos announced that the MGI would be decommissioned in 2021.

As a client of SMS/800 for over 25 years, 8MS was built on the MGI for all toll-free provisioning. For our API clients, rather than returning MGI formatted responses and forcing our customers to learn the rather unusual message formatting of the MGI, we chose to provide a more readable,consistent interface. The framework of the 8MS API responses is XML. For values being returned from SMS/800, all values are stored within the <msgparams> tag and these values are returned as key/value pairs. Each key/value is separated by an equals (=) sign and each pair is separated by a newline. A simple example of this can be seen in the **Request Results** section of the **API Overview** page of the API Guide.

Beginning with 8MS release 17.1, we have begun porting the 8MS API code, replacing MGI calls with Registry API calls. Early on in the design of this release, we established a set of ground rules for the conversion. Primary among those rules was to provide backward compatiblity for our API customers. We have striven to do exactly that, but found during development that it was not possible to be 100% backward compatible. The remainder of this page of the API Guide documents any known items that prevent 100% backward compatibility.

MsgParams Missing from Registry API Call Responses

In some cases, where the MGI API returned a key/value pair, the Registry API does not return this value. Each key known to be missing is listed below.

RED (Requested Effective Date), RET (Requested Effective Time)

The RED and RET fields are not returned where they are expected, in the following 8MS API calls:

- RecordQuery
- RecordRetrieve
- NumberRetrieveAll
- SmsTemplateQuery
- SmsTemplateRetrieveAll

LNS (Number of Terminating Lines)

The LNS field is not returned where it is expected, in the following 8MS API calls:

- SmsTemplateRetrieve
- SmsTemplateRetrieveAll

Extra MsgParams in Registry API Calls

In some cases, the Registry API returns additional key/value pairs. These are keys that were added by Somos in the Registry API that do not exist in the MGI API. These keys may be

distinguished in the XML response by the case of the key name. All known MGI keys are listed in all upper case; Registry keys that do not exist in MGI are listed in mixed case. Each extra key is listed below. In all cases, the calling client may safely ignore these key/value pairs.

recVersionId, lockStatus, reqeffDtTm, numStatus, perms, custRecCompPart, actionCode, lastUpDt, priority, lastUsr

These fields may appear in the response of many 8MS API calls.

stat

for each SMS Template record in the 8MS API call *SmsTemplateList*, the response will contain this field.

Inconsistencies in Registry API Calls

In many cases, the Registry API call responses returned in 8MS API calls will have the following issues.

MsgParam Order

The keys listed in the <msgparams> tag in a response from a Registry API call may not be in the same order they were in a response from an MGI API call.

If you currently parse the msgparams with an expectation of the order, you will experience problems.

10 Record Limit on Queries

When querying toll-free number and SMS Template records, the MGI had a limit of 10 records that would be returned in a single response. The Registry API does not have this limit, so all records will be returned in a single call. This impacts the following 8MS API calls:

- RecordQuery
- NumberQueryAll
- SmsTemplateQuery

SMS Template List returns errorcd=10

The 8MS *SmsTemplateList* API call, when using the Registry API, returns a warning with an *errorcd* of *10* and a VERR of "No Template records exist for the requested keys." if a non-existant Template Name (8MS API param *templatename*) is provided. The MGI did not return this warning.

8MS Message Requests

Message requests are those that cause messages to be sent to SMS/800 for a single toll-free number or SMS Template. The following single-number requests are supported by the 8MS API:

CopyRecord
Disconnect
GeoInfo
MultiChangeResporg
NumberChange
NumberHistory
NumberQuery
NumberQueryAll
NumberRetrieve
NumberRetrieveAll
NumberSpare
PointerRecordChange
RecordChange
RecordDelete
RecordQuery
RecordRetrieve
RecordTransfer
ReferralQuery
ReptAsi
Reserve
ReserveActivate
ReserveSP
Reserve8fa
ScpQuery
ScpResend
Search
SearchSP
Search8fa
SmsTemplateChange
SmsTemplateCopy
SmsTemplateDelete
SmsTemplateDisconnect
SmsTemplateHistory
SmsTemplateList
SmsTemplateQuery
SmsTemplateRetrieve
SmsTemplateRetrieveAll
SmsTemplateTransfer
SmsTemplateScpQuery
SmsTemplateScpResend
Spare8fa

Name

CopyRecord Copy and send a customer record

Description

CopyRecord copies a customer record from a given number, 8MS Routing Set or SMS Template and sends a record change for a second number and date/time. This provides, for example, the ability to apply an 8MS Routing Set record to any dialed number. The following are the allowed fields in *reqparams*:

* **source** The number, 8MS Routing Set or SMS Template containing the customer record to be copied. The value is interpreted according to these rules:

    1. If a **sourceDateTime** is specified, the value of **source** is assumed to be a toll free number unless the source begins with a *, in which case it is assumed to be an SMS Template. The pair of values identifies the exact record from which the copy is to be made.
    2. If no **sourceDateTime** is specified, the value is assumed to name an 8MS Routing Set from which the copy is to be made.
    3. If no **sourceDateTime** is specified and no matching 8MS Routing Set is found, the value is assumed to be a toll free number unless the source begins with a *, in which case it is assumed to be an SMS Template. The latest **ACTIVE** or **SENDING** record for that number or SMS Template is used as the source for the copy.

    If an 8MS Routing Set has been given a toll free number as its name, and the intended action is to use the toll free number as the source of a copy, **sourceDateTime** must be used to specify the exact record to be copied from. Otherwise, the 8MS Routing Set will be used as the source.

sourceDateTime The effective date/time of the source record, to be used when **source** is a toll-free number. Note that this field is *not* used for an SMS Template source. In the case where the source is an SMS Template, the current Active, Sending, most recent Pending or most recent Old record is used.

The format of this value is:

```
mm/dd/yy hh:mmX/Y
```

where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C.

Be aware that Somos only allows schedule date/times where the minute on the quarter hour :00, :15, :30, or :45.

Any other value for minute will cause the record to fail validation.

*	**destNumber**	The number for which the change is sent. This field cannot contain any formatting characters; it may only contain alphanumerics.
*	**destDateTime**	The effective date/time of the destination record. The format of this value is as described for **sourceDateTime** above, and may also contain the value **NOW** to request an immediate change.
	highpriority	This parameter provides access to the SMS/800 High Priority Update feature. Setting it to **1** will tell SMS to place this customer record at the front of the SCP queue. Only a small number of these updates can be done each day. After that, if this parameter is set to **1** the message will fail. You may check the Activity Log to see how many such updates remain for the current day.

MGI Messages

CopyRecord sends a **REQ-CRC** message with an action code of either **N** or **C**. The response message is a **RSP-CRC**.

If the specified dialed number is not known in the 8MS database, the number status information will first be retrieved from SMS/800. Note that the retrieve will only be done on dedicated and private sites; it will not be done on any of the service bureau sites. This requires sending a **REQ-NSR** message; the response is **RSP-NSR**.

Example

The following request parameters will retrieve the routing from the 8MS Routing Set called **CLASS1**, and send it for **8005551212**, **08/15/04 02:00P:**

```
source~CLASS1;destNumber~8005551212;destDateTime~08/15/04 02:00P
```

Name

Disconnect Schedule a disconnect of an active number

Description

Disconnect schedules a disconnect of the accompanying active number. The following are the allowed fields in *reqparams*:

* **dialedNumber** The number scheduled to be disconnected.

* **dateTime** The effective date/time of the record. The format of this value is:

  `mm/dd/yy hh:mmX/Y`

  where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C.

  > *Be aware that Somos only allows schedule date/times where the minute on the quarter hour :00, :15, :30, or :45.*
  >
  > *Any other value for minute will cause the record to fail validation.*

  This field may also contain the value **NOW** to request an immediate disconnect.

interceptDate The End Intercept Date, sent in the SMS/800 **eint** field. Format is *MM/DD/YY*.

This field may also contain the value **NOW** to allow the number to become transitional immediately.

referralOption The Referral Option, sent in the SMS/800 **refer** field, and used for disconnect records. Allowable values are **0** and **1**.

highpriority This parameter provides access to the SMS/800 High Priority Update feature. Setting it to **1** will tell SMS to place this customer record at the front of the SCP queue. Only a small number of these updates can be done each day. After that, if this parameter is set to **1** the message will fail. You may check the Activity Log to see how many such updates remain for the current day.

MGI Messages

Disconnect sends a **REQ-CRC** message with an action code of **D**. The response message is a **RSP-CRC**.

Example

The following request parameters will send a disconnect of **8005551212** scheduled for **4/20/04 08:00A/C** with an intercept date of **4/30/04**:

```
dialedNumber~8005551212;dateTime~04/20/04 08:00A/C;
interceptDate~04/30/04
```

Name

GeoInfo Get geographic information

Description

GeoInfo is used to retrieve geographic information about a particular state, lata, npa, or npanxx. All appropriate state, lata, npa and npanxx values are returned. The following are the allowed fields in *reqparams*:

* **findType** The type of geographic information to be searched. Valid values are **state**, **lata**, **npa**, or **npanxx**.
* **findValue** A valid, appropriate value for the specified **findType**. For example, if **findType** is **state** then **findValue** should be a valid two character state code.

MGI Messages

None.

Example

The following request will ask for all geographic data for **New Jersey**:

```
findType=state;findValue=NJ
```

The following request will ask for all geographic data for NPA **732**:

```
findType=npa;findValue=732
```

Name

MultiChangeResporg Change the RespOrg of a toll-free number

Description

MultiChangeResporg is used to change the RespOrg of a number. (The SMS/800 MGI message used by this 8MS API call also allows multiple numbers to be sent in one message; see **BatchMultiChangeResporg** for more information).

> *Note that 8MS had another API call, **ChangeResporg**, which performed the same action but had the side effect of creating a new customer record. The **ChangeResporg** API call has been deprecated and should no longer be used.*

The following are the allowed fields in *reqparams*:

* **dn** The toll-free number to be changed.
* **resporg** The new RespOrg code.

MGI Messages

MultiChangeResporg sends a **REQ-MRO** message. The response message is a **RSP-MRO**.

Example

The following request parameters will attempt to change the RespOrg of **8005551212** to **ART01**:

```
dn~8005551212;resporg~ART01
```

Name

NumberChange Change information associated with a number

Description

NumberChange is used to change the administrative information associated with a number. The following are the allowed fields in *reqparams*:

*	**dn**	The number to be changed.
	ru	The *reserved until* date.
	cname	The contact person name.
	cphone	The contact person phone number.
	notes	Descriptive notes.
	reserve	If set to *1* for a toll-free number with a status of *Transitional* the number may be re-reserved. If set to *0* or omitted the number is not re-reserved.

MGI Messages

NumberChange sends a **REQ-NSC** message with an action code of **C**. The response message is a **RSP-NSC**.

Example

The following request parameters will update the contact name and phone number for **8005551212**:

```
dn~8005551212;cname~J. Smith;cphone~7323020222
```

Name

NumberHistory Retrieve the change history of a number.

Description

SMS/800 maintains a database of the change history of all toll free numbers. Any time a change is made, an entry is made noting the date/time of the change, the login ID used to make the change, a RespOrg change that may have resulted, and/or a change to the status of the number. The information in the SMS history database is updated nightly. Thus, a change may take up to a day to be reflected in the history database.

NumberHistory provides a means to access this information via the 8MS API. The following are the allowed fields in *reqparams*:

* **dn** The number to be queried.

MGI Messages

None.

Results

The result of a **NumberHistory** request (i.e., the change history of the given number) is represented as an XML document. This XML document will be contained with the **<reqparams>** tag of the normal 8MS API return structure. For example, the following might be returned from a history request:

```
<API8MS>
<reqid>OA03743771</reqid>
<message>
<routeid></routeid>
<termrpt>COMPLD</termrpt>
<errorcd>0</errorcd>
<reqparams>
<history dn="8666006000">
 <change>
   <change-date>05/29/2003 10:50:09</change-date>
   <status>WORKING</status>
   <status-change-date>07/29/2000 18:23:31</status-change-date>
   <resporg>ART10</resporg>
   <prev-resporg>ART01</prev-resporg>
   <logon-id>ART01000</logon-id>
   <download-date>05/30/2003 00:01:00</download-date>
 </change>
 <change>
   <change-date>07/29/2000 18:23:31</change-date>
   <status>WORKING</status>
   <status-change-date>07/29/2000 18:23:31</status-change-date>
   <resporg>ART01</resporg>
   <prev-resporg>     </prev-resporg>
   <logon-id>ART01000</logon-id>
   <download-date>03/28/2002 12:56:29</download-date>
 </change>
 <change>
   <change-date>07/29/2000 18:23:27</change-date>
   <status>ASSIGNE</status>
   <status-change-date>07/29/2000 18:23:27</status-change-date>
   <resporg>ART01</resporg>
   <prev-resporg>     </prev-resporg>
   <logon-id>ART01000</logon-id>
   <download-date>03/12/2002 21:14:26</download-date>
 </change>
```

```
 <change>
   <change-date>07/29/2000 12:01:25</change-date>
   <status>RESERVE</status>
   <status-change-date>07/29/2000 12:01:25</status-change-date>
   <resporg>ART01</resporg>
   <prev-resporg>     </prev-resporg>
   <logon-id>ART01000</logon-id>
   <download-date>03/12/2002 21:14:04</download-date>
 </change>
</history>

</reqparams>
</message>
</API8MS>
```

Each **<change>** within the **<history>** consists of the following:

change-date	the date/time of the change
status	the number status at that point in time
status-change-date	the date/time the number attained the given status
resporg	the controlling resporg of the number at that time
prev-resporg	the previous resporg (if the associated change included a resporg change)
logon-id	the SMS800 logon ID used to make the change
download-date	the date/time that the change record was retrieved from the main SMS/800 DB

Example

The following request parameter will request the history for the number **8005551212**:

```
dn~8005551212
```

Name

NumberQuery Query number information

Description

NumberQuery retrieves the administrative information associated with a number. The following are the allowed fields in *reqparams*:

* **dn** The number to be queried.

MGI Messages

NumberQuery sends a **REQ-NSR** message with an action code of **Q**. The response message is a **RSP-NSR**.

Example

The following request parameters will send a query for the number **8005551212**:

```
dn~8005551212
```

Name

NumberQueryAll Query number and customer record information

Description

NumberQueryAll queries the administrative information associated with a number, as well as the status of customer records for the number. The following are the allowed fields in *reqparams*:

* **dn** The number to be queried.

MGI Messages

NumberQueryAll sends a **REQ-NSR** message with an action code of **Q**. The response message is a **RSP-NSR**. A **REQ-CRQ** is then sent to retrieve customer record status information. If more than 10 customer records exist for the number, additional **REQ-CRQ** messages are sent. For each **REQ-CRQ** message sent a **RSP-CRQ** message is received.

Example

The following request parameters will query all information associated with the number **8005551212**:

```
dn~8005551212
```

Name

NumberRetrieve Retrieve number information

Description

NumberRetrieve retrieves the administrative information associated with a number, and stores it in the 8MS database. The following are the allowed fields in *reqparams*:

* **dn** The number to be retrieved.

MGI Messages

NumberRetrieve sends a **REQ-NSR** message with an action code of **Q**. The response message is a **RSP-NSR**.

Example

The following request parameters will retrieve the information associated with the number **8005551212**:

```
dn~8005551212
```

Name

NumberRetrieveAll Retrieve number and customer record information

Description

NumberRetrieveAll retrieves the administrative information associated with a number, as well as all customer records, and stores them in the 8MS database. The following are the allowed fields in *reqparams*:

*	**dn**	The number to be retrieved.
	forceall	Normally only customer records whose status appear on the list defined by the system parameter **RetrieveStatusList** will be retrieved. The **forceall** *reqparam* entry may be set to **1** to force all records to be retrieved (i.e. ignore the **RetrieveStatusList** parameter).
	involved	Normally, if 8MS believes the number being retrieved is not owned by the RespOrg making the request, it will not be stored. This means that records not owned, but for which the RespOrg is an involved carrier, will not be stored. The **involved** *reqparam* entry will force un-owned records to be stored.

MGI Messages

NumberRetrieveAll is implemented by sending **REQ-NSR**, followed by one or more **REQ-CRQ** messages to find the effective date/times of all customer records (returned in **RSP-CRQ**). A **REQ-CRV** message is then sent for each effective date/time. The response to **REQ-CRV** is **RSP-CRV**.

Example

The following request parameters will retrieve all information associated with the number **8005551212**:

```
dn~8005551212
```

Name

NumberSpare Spare a number

Description

NumberSpare is used to spare a previously reserved number. The following are the allowed fields in *reqparams*:

* **dn** The number to be spared.

MGI Messages

NumberSpare sends a **REQ-NSC** with an action code of **S**. The response message is **RSP-NSC**.

Example

The following request parameters will attempt to spare the number **8005551212**:

```
dn~8005551212
```

Name

PointerRecordChange Update a pointer (PAD) record

Description

PointerRecordChange updates the pointer (PAD) record associated with a given number. The following are the allowed fields in *reqparams*:

* **dialedNumber** The number containing the customer record, sent in the SMS/800 **num** field.

* **dateTime** The effective date/time of the record. The format of this value is:

```
mm/dd/yy hh:mmX/Y
```

where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C.

> *Be aware that Somos only allows schedule date/times where the minute on the quarter hour :00, :15, :30, or :45.*
>
> *Any other value for minute will cause the record to fail validation.*

This field may also contain the value **NOW** to request an immediate change.

* **templatename** The SMS Template to which this record should point.

orderNumber The Service Order Number, sent in the SMS/800 **so** field. Must be from 4-13 bytes, where the first character is alphabetic, the second through twelfth characters are alphanumeric, and the thirteenth character is alphabetic.

sfNumber The Supplemental Form Number, sent in the SMS/800 **sf** field. Must be 1-6 alphanumeric characters. Required if Service Order Number is not specified.

notes Commentary to be sent in the SMS/800 **note** field. Can be up to 151 characters.

lns The number of terminating lines sent in the SMS/800 **lns** field. Format is 1 to 4 decimal numerals.

duedate The Due Date, sent in the SMS/800 **dd** field. Format is *MM/DD/YY*. If not specified, SMS/800 will generate a value.

holdValue The Hold Due Date indicator, sent in the SMS/800 **hdd** field. Allowable values are **0** and **1**.

rao The Revenue Accounting Office, sent in the SMS/800 **rao** field. Must be 3 bytes, decimal numerals. The following rules apply to this field:

- If the Bill To Number (**billToNumber**) is **9999999999**, then **rao**

must be **999**. If it is not specified, SMS/800 will generate a default value of **999**.
- If complex routing is specified, then **rao** is required, and cannot be **999**.

billToNumber The Bill To, or Alternate Billing Number, sent in the SMS/800 **abn** field. Must be 10 alphanumeric characters. If **rao** is **999**, then **billToNumber** must be **9999999999**.

customer The On-Line Access Customer, sent in the SMS/800 **cus** field. Must be 5 alphanumeric characters.

agent The On-Line Agent for Customer, sent in the SMS/800 **agent** field. Must be 5 alphanumeric characters.

telco The company that sold SMS access, sent in the SMS/800 **telco** field. Must be 4 alphanumeric characters.

icecBilling The IC/EC Billing Indicator, sent in the SMS/800 **cbi** field. Must be 3 alphabetic characters for an Interexchange Carrier, 4 alphabetic characters for an Exchange Carrier.

contactName The name of the contact person, sent in the SMS/800 **ncon** field. Maximum of 30 characters.

contactPhone The Contact Phone Number, sent in the SMS/800 **ctel** field. Must be 10 decimal numerals.

interceptDate The End Intercept Date, sent in the SMS/800 **eint** field. Format is *MM/DD/YY*. An End Intercept Date is used for disconnect records. Specifying a value for **interceptDate** will cause a disconnect rather than a change to be sent.

referralOption The Referral Option, sent in the SMS/800 **refer** field, and used for disconnect records. Allowable values are **0** and **1**.

highpriority This parameter provides access to the SMS/800 High Priority Update feature. Setting it to **1** will tell SMS to place this customer record at the front of the SCP queue. Only a small number of these updates can be done each day. After that, if this parameter is set to **1** the message will fail. You may check the Activity Log to see how many such updates remain for the current day.

MGI Messages

PointerRecordChange sends a **REQ-CRC** message with an action code of **N**, **C** or **D**. The response message is a **RSP-CRC**.

Example

The following request parameters will send an immediate change to **8005551212**. The SMS Template will be set to ***AR-CSF02** and the order number to **ORDER1**:

```
dialedNumber~8005551212;dateTime~NOW;templatename~*AR-CSF02;
```

orderNumber~ORDER1

Name

RecordChange Update a customer record

Description

RecordChange updates the customer record associated with a given number. The following are the allowed fields in *reqparams*:

*	**dialedNumber**	The number containing the customer record, sent in the SMS/800 **num** field.
*	**dateTime**	The effective date/time of the record. The format of this value is: `mm/dd/yy hh:mmX/Y` where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C. *Be aware that Somos only allows schedule date/times where the minute on the quarter hour :00, :15, :30, or :45.* *Any other value for minute will cause the record to fail validation.* This field may also contain the value **NOW** to request an immediate change.
	resporg	The RespOrg, sent in the SMS/800 **newro** field. This field allows a RespOrg change to be sent.
	interLATACarrier	The numeric CIC code of the primary InterLATA carrier, sent in the SMS/800 **iec** field.
	intraLATACarrier	The numeric CIC code of the primary IntraLATA carrier, sent in the SMS/800 **iac** field.
	potsTermination	A POTS number to which SMS/800 will direct calls, sent in the SMS/800 **tel** field. Also see **interLATAPOTSValue** below.
	interLATAPOTSValue	Allows InterLATA POTS to be specifed. If a value is specified in the potsTermination field, and the value of **interLATAPOTSValue** is **0**, both the POTS and Toll Free (**dialedNumber**) values will be sent as **tel**s. If the value is **1**, only the given POTS number will be sent.
	orderNumber	The Service Order Number, sent in the SMS/800 **so** field. Must be from 4-13 bytes, where the first character is alphabetic, the second through twelfth characters are alphanumeric, and the thirteenth character is alphabetic.

sfNumber The Supplemental Form Number, sent in the SMS/800 **sf** field. Must be 1-6 alphanumeric characters. Required if Service Order Number is not specified.

planDN The dialed number serving as the source of the complex routing plan. Also see **complexRecord** below.

complexRecord The name of the entry in the RPLAN or TRPLAN table that contains the complex routing for this record. If **planDN** was specified, then this value and **complexRecord** are used as an index into RPLAN. If **planTemplatename** was specified, then this value and **complexRecord** are used as an index into TRPLAN. If neither **planDN** nor **planTemplatename** are specified then **complexRecord** is assumed to be a local SMS Template plan and that value alone is used as an index into TRPLAN.

notes Commentary to be sent in the SMS/800 **note** field. Can be up to 151 characters.

lns The number of terminating lines sent in the SMS/800 **lns** field. The format of this value is:

```
num^numlines,num^numlines,...
```

where num is a POTS or toll-free number and numlines is the actual number of lines. The number of lines must be 1 to 4 digits. Since a CPR allows multiple POTS terminations to be specified along with the toll-free number, this value can be a comma-separated field that allows each individual POTs term plus the toll-free number term to have unique values.

In the case where **lns** is set and no CPR is to be used, the **complexRecord** field must still be specified, but with no value, as shown in the following example:

```
complexRecord~;lns~8005551212^888;
```

duedate The Due Date, sent in the SMS/800 **dd** field. Format is *MM/DD/YY*. If not specified, SMS/800 will generate a value.

holdValue The Hold Due Date indicator, sent in the SMS/800 **hdd** field. Allowable values are **0** and **1**.

rao The Revenue Accounting Office, sent in the SMS/800 **rao** field. Must be 3 bytes, decimal numerals. The following rules apply to this field:

- If the Bill To Number (**billToNumber**) is **9999999999**, then **rao** must be **999**. If it is not specified, SMS/800 will generate a default value of **999**.
- If complex routing is specified, then **rao** is required, and cannot be **999**.

billToNumber	The Bill To, or Alternate Billing Number, sent in the SMS/800 **abn** field. Must be 10 alphanumeric characters. If **rao** is **999**, then **billToNumber** must be **9999999999**.
customer	The On-Line Access Customer, sent in the SMS/800 **cus** field. Must be 5 alphanumeric characters.
agent	The On-Line Agent for Customer, sent in the SMS/800 **agent** field. Must be 5 alphanumeric characters.
telco	The company that sold SMS access, sent in the SMS/800 **telco** field. Must be 4 alphanumeric characters.
icecBilling	The IC/EC Billing Indicator, sent in the SMS/800 **cbi** field. Must be 3 alphabetic characters for an Interexchange Carrier, 4 alphabetic characters for an Exchange Carrier.
contactName	The name of the contact person, sent in the SMS/800 **ncon** field. Maximum of 30 characters.
contactPhone	The Contact Phone Number, sent in the SMS/800 **ctel** field. Must be 10 decimal numerals.
interceptDate	The End Intercept Date, sent in the SMS/800 **eint** field. Format is *MM/DD/YY*. An End Intercept Date is used for disconnect records. Specifying a value for **interceptDate** will cause a disconnect rather than a change to be sent.
referralOption	The Referral Option, sent in the SMS/800 **refer** field, and used for disconnect records. Allowable values are **0** and **1**.
network	Network Area of Service, sent in the SMS/800 **anet** field. The value is a comma-separated list of valid SMS/800 network codes.
state	State Area of Service, sent in the SMS/800 **asta** field. The value is a comma-separated list of valid SMS/800 state codes.
areacode	Area Code Area of Service, sent in the SMS/800 **aac** field. The value is a comma-separated list of valid area codes.
lata	LATA Area of Service, sent in the SMS/800 **alat** field. The value is a comma-separated list of valid LATAs.
labels	Label Area of Service, sent in the SMS/800 **albl** field. The value is a comma-separated list of valid SMS/800 AOS label names. The SMS/800 MGI provides no means to define AOS labels; they must previously exist.
directoryAssistanceType	The Directory Assistance Type, sent in the SMS/800 **dat** field. The valid values are **N** (Normal), **G** (Government), and **F** (Frequently Called).
updateValue	The Directory Assistance Update indicator, sent in the SMS/800 **dau** field. The valid values are **0** and **1**.
listingType	The Listing Type, sent in the SMS/800 **li** field. The valid values

	are **BL** (BLocked), **LI** (Published), and **NP** (Non-published).
address	The Listing Address, sent in the SMS/800 **la** field. Can be up to 75 characters.
names	The Listing Names, sent in the SMS/800 **ln** field. Up to nine listing names can be specified, each up to 75 characters, separated by the **^** character.
highpriority	This parameter provides access to the SMS/800 High Priority Update feature. Setting it to **1** will tell SMS to place this customer record at the front of the SCP queue. Only a small number of these updates can be done each day. After that, if this parameter is set to **1** the message will fail. You may check the Activity Log to see how many such updates remain for the current day.

MGI Messages

RecordChange sends a **REQ-CRC** message with an action code of **N**, **C** or **D**. The response message is a **RSP-CRC**.

Example

The following request parameters will send an immediate change to **8005551212**. The InterLATA carrier will be set to **0432**, the order number to **ORDER1**, the area of service to **US**, and the listing name to **CSF**:

```
dialedNumber~8005551212;dateTime~NOW;interLATACarrier~0432;
orderNumber~ORDER1;names~CSF;network~US
```

Name

RecordDelete Delete a customer record

Description

RecordDelete deletes a customer record associated with a given number. The following are the allowed fields in *reqparams*:

* **dn** The number containing the record to be deleted.
* **datetime** The effective date/time of the record to be deleted. The format of this value is:

  mm/dd/yy hh:mmX/Y

  where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C.

MGI Messages

RecordDelete sends a **REQ-CRC** message with an action code of **X**. The response message is a **RSP-CRC**.

Example

The following request parameters will delete the customer record for **07/01/04 10:00am** from **8005551212**:

dn~8005551212;datetime~07/01/04 10:00A/C

Name

RecordQuery Query the status of a customer record

Description

RecordQuery requests the status of the customer record associated with a given number. The following are the allowed fields in *reqparams*:

*	**dn**	The number to be queried.
	ed	The effective date of the record to be queried.
	et	The effective time of the record to be queried.

If **ed** and **et** are omitted, all versions of the customer record are requested. If **ed** is specified and **et** is omitted, all versions with dates equal to or greater than the given **ed** are returned. If **ed** and **et** are specified, all versions with dates and times equal to or greater than the given values are returned.

MGI Messages

RecordQuery sends the **REQ-CRQ** message. The SMS/800 MGI will only return, at most, 10 records for a single **REQ-CRQ** request. In cases where more than 10 records match the request, 8MS automatically takes care of requesting the additional records, generating additional **REQ-CRQ** requests as appropriate. The response to **REQ-CRQ** is **RSP-CRQ**. In cases where one or more additional **RSP-CRQ** messages are expected, the **MORE** field in the **RSP-CRQ** message will be set to **Y**.

Example

The following request parameters will query all customer records for **8005551212**:

```
dn~8005551212
```

Name

RecordRetrieve Retrieve a customer record

Description

RecordRetrieve retrieves a version of a customer record and stores it in the 8MS database. The following are the allowed fields in *reqparams*:

*	**dn**	The number containing the customer record to be retrieved.
	ed	The effective date of the record to be retrieved.
	et	The effective time of the record to be retrieved.

The **ed** and **et** fields are optional. However, if one is specified then both must be specified. If neither one is specified, the most recent record with a status of **ACTIVE**, **SENDING** or **DISCONNECT** will be returned.

MGI Messages

RecordRetrieve sends a **REQ-CRV** message. The response message is a **RSP-CRV**.

Example

The following request parameters will retrieve the **10/01/04 10:00 am** version of the customer record associated with **8005551212**:

```
dn~8005551212;ed~10/01/04;et~10:00A/
```

Name

RecordTransfer Transfer a customer record

Description

RecordTransfer transfers a customer record associated with a given number. The following are the allowed fields in *reqparams*:

* **dn** The number containing the customer record to be transferred.

* **fromDate** The source effective date/time for the transfer. The format of this value is:

  ```
  mm/dd/yy hh:mmX/Y
  ```

  where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C.

  Be aware that Somos only allows schedule date/times where the minute on the quarter hour :00, :15, :30, or :45.

  Any other value for minute will cause the record to fail validation.

* **toDate** The new effective date/time for the record. The format is as described above. This entry may also contain the value **NOW** to request a transfer to the current date/time.

MGI Messages

RecordTransfer sends a **REQ-CRC** message with an action code of **T**. The response message is a **RSP-CRC**.

Example

The following request parameters will transfer the **10/01/04 10:00am** record in **8005551212** to **10/15/04 02:00pm**.

```
dn~8005551212;fromDate~10/01/04 10:00A/C;toDate~10/15/04 02:00P/C
```

Name

ReferralQuery Query a trouble referral number

Description

ReferralQuery queries the trouble referral number associated with a RespOrg or a number. The following are the allowed fields in *reqparams*:

dn The number to be queried.

resporg The RespOrg to be queried.

Either **dn** or **resporg** must be specified, not both.

MGI Messages

ReferralQuery sends a **REQ-TRN** message. The response message is a **RSP-TRN**.

Example

The following request parameters will query the trouble referral number associated with **8005551212**:

```
dn~8005551212
```

Name

ReptAsi Report version to SMS/800

Description

ReptAsi sends the OS version to SMS/800. The version to send is obtained from the system parameter **PARAM_TA_VERSION**. There are no relevant *reqparams* entries.

MGI Messages

ReptAsi sends a **REPT-ASI** message. There is no response message from SMS/800.

Name

Reserve Reserve numbers

Description

Reserve is used to reserve numbers in the SMS/800 database. The following are the allowed fields in *reqparams*:

	numbers	A comma-separated list of up to 10 particular numbers to be reserved, or a single number containing either ***** or **&** (i.e., a pattern or *wildcard* search).
*	**resporg**	The resporg to be associated with any numbers reserved.
	npa	The NPA of the toll-free number(s).
	nxx	The starting NXX of the toll-free number(s).
	line	The starting line of the toll-free number(s).
	npalist	A comma-separated list of NPA's: 800, 888, 877, 866, 855, 844 or 833.
	numberword	The number being searched for. The format is 1-7 characters. Valid characters include: • alpha-numeric, • ***** (match any character), • **&** (repeat a digit - e.g. *&0&1* finds *7071*, *8081*) • **%** (a second repeating digit - e.g. *&%&%* finds *7070*, *8080*, *1212*, *3434*) • **^** (starts with), and • **$** (ends with).
	trailingdigits	The number of trailing digits: 0, 1 or 2. *If **trailingdigits** is 0, 1 or 2 the search will return results that match up to the trailingdigits value.* *If no **trailingdigits** parameter is specified, the search is first executed with a **trailingdigits** value of **0**. If no results are returned then a second search is executed with a **trailingdigits** value of **2**.*
	count	The count of numbers to be reserved. *Note that if **count** is not specified, it is defaulted to 1, regardless of the actual amount of numbers specified in the **numbers** field.*
	contiguous	Whether or not the reserved numbers must be contiguous; value must be **0** or **1**.
	minimum	For a contiguous request, the minimum count of numbers that is

		acceptable. For example, if a request for 100 contiguous numbers cannot be satisfied, with a **minimum** of 50, 50 contiguous numbers will be returned if available.
*	**cname**	The contact person name.
*	**cphone**	The contact person phone number.
	notes	Descriptive notes.
	numlist	If set to **1**, number lists will be created containing the numbers successfully reserved, and those that were not reserved. If set to **0** or omitted, no such lists are made.
	nlname	The name of the number list to be created. This is only used if numlist is set to 1.
	custname	This field is used for individual company auditing and tracking purposes, allowing a company to track the customer for which this number was reserved. This field is stored on 8MS only; it is not sent to SMS/800.
	btn	This field is used for individual company auditing and tracking purposes, allowing a company to track the billing telephone number for which this number was reserved. This field is stored on 8MS only; it is not sent to SMS/800.
	onum_sreq	This field is used for individual company auditing and tracking purposes, allowing a company to track the order number or service request for which this number was reserved. This field is stored on 8MS only; it is not sent to SMS/800.

Depending on what fields are used for the query, a Search request may first be generated to 800ForAll. The following fields will generate a Search to 800ForAll: ***npalist****,* ***numberword****,* ***trailingdigits****.*

MGI Messages

Reserve sends a **REQ-NSR** message with an action code of **R**. The response message is a **RSP-NSR**.

The SMS/800 MGI will allow, at most, 10 numbers to be reserved in a single **REQ-NSR** request. If the specified **count** is greater than ten, 8MS will automatically generate additional **REQ-NSR** messages.

Example

The following will attempt to reserve ten **800** numbers starting at NXX **777**:

```
npa~800;nxx~777;count~10;resporg~ART01;cname~BSO;cphone~7323566999
```

The following will attempt to reserve two specific numbers:

```
numbers~8007765555,877CALLCSF;resporg~ART01;cname~BSO;cphone~7323566999
```

The following is an example of using SMS/800 wildcards in the requested number; these

parameters specify a search for any 800 number where the last four digits are the same:

*numbers~800***&&&&;resporg~ART01;cname~BSO;cphone~7323566999*

Name

ReserveActivate Reserve numbers

Description

ReserveActivate is used to reserve numbers in the SMS/800 database and then immediately activate those numbers. The following are the allowed fields in *reqparams*:

	numbers	A comma-separated list of up to 10 particular numbers to be reserved, or a single number containing either ***** or **&** (i.e., a pattern or *wildcard* search).
*	**resporg**	The resporg to be associated with any numbers reserved.
	npa	The NPA of the toll-free number(s).
	nxx	The starting NXX of the toll-free number(s).
	line	The starting line of the toll-free number(s).
	count	The count of numbers to be reserved.
	contiguous	Whether or not the reserved numbers must be contiguous; value must be **0** or **1**.
	minimum	For a contiguous request, the minimum count of numbers that is acceptable. For example, if a request for 100 contiguous numbers cannot be satisfied, with a **minimum** of 50, 50 contiguous numbers will be returned if available.
*	**cname**	The contact person name.
*	**cphone**	The contact person phone number.
	notes	Descriptive notes.
	numlist	If set to **1**, number lists will be created containing the numbers successfully reserved, and those that were not reserved. If set to **0** or omitted, no such lists are made.
	nlname	The name of the number list to be created. This is only used if numlist is set to 1.
*	**source**	The number or 8MS Routing Set containing the customer record to be copied. The value specified will first be tried as a the name of an 8MS Routing Set. If an 8MS Routing Set with the given name is found, it is used as the source. Otherwise, it is assumed that the source is a toll free number.
*	**sourceDateTime**	The effective date/time of the source record, to be used when **source** is a toll-free number. The format of this value is: `mm/dd/yy hh:mmX/Y` where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C.

Be aware that Somos only allows schedule date/times where the minute on the quarter hour :00, :15, :30, or :45.

Any other value for minute will cause the record to fail validation.

If sourceDateTime is omitted, the latest **ACTIVE** or **SENDING** record in the source dialed number will be used. If there is no such record, CopyRecord fails.

*	**destDateTime**	The effective date/time of the destination record. The format of this value is as described for **sourceDateTime** above, and may also contain the value **NOW** to request an immediate change.
	comment	A comment describing the batch activate operation.
	highpriority	This parameter provides access to the SMS/800 High Priority Update feature. Setting it to **1** will tell SMS to place this customer record at the front of the SCP queue. Only a small number of these updates can be done each day. After that, if this parameter is set to **1** the message will fail. You may check the Activity Log to see how many such updates remain for the current day.

MGI Messages

ReserveActivate sends a **REQ-NSR** message with an action code of **R**. The response message is a **RSP-NSR**.

The SMS/800 MGI will allow, at most, 10 numbers to be reserved in a single **REQ-NSR** request. If the specfied **count** is greater than ten, 8MS will automatically generate additional **REQ-NSR** messages.

For each number successfully reserved, **ReserveActivate** sends a **REQ-CRC** with an action code of **N**. The response message is a **RSP-CRC**

Example

The following will attempt to reserve 10 **800** numbers starting at NXX **777** and activate the numbers in the 8MS Routing Set called **CLASS1**, on **08/15/05 02:00P**:

```
npa=800;nxx=777;count=10;resporg=ART01;cname=BSO;
cphone=7323566999;source~CLASS1;destDateTime~08/15/05 02:00P
```

Name

ReserveSP Reserve numbers from either SMS/800 or 800ForAll

Description

ReserveSP is used to reserve numbers in the SMS/800 or 800ForAll database. The following are the allowed fields in *reqparams*:

npalist A comma-separated list of NPA's: 800, 888, 877, 866, 855, 844 or 833.

numbers A comma-separated list of up to 10 SMS/800 numbers to be reserved.

numbersSP A comma-separated list of up to 10 available 800ForAll numbers, each with it's associated Resp Org, to be reserved. The format of this list is

```
number^resporg,number^resporg,...
```

> *If **numbersSP** has been specified, for any matching numbers found in the special pool, a **MultiChangeResporg** is performed to move these numbers to the specified **resporg**.*

* **resporg** The resporg to be associated with any numbers reserved.

* **cname** The contact person name.

* **cphone** The contact person phone number.

notes Descriptive notes.

> *One, and only one, of **numbers** or **numbersSP** must be specified.*

MGI Messages

If the **numbers** parameter has been specified then a **REQ-NSR** message is sent with an action code of **R**. The response message is a **RSP-NSR**.

If the parameter **numbersSP** has been specified then no MGI message is generated.

Example

The following example will reserve two numbers from SMS/800:

```
numbers~8005551212,8885551212;resporg~YHN19;cname~JQC;cphone~7323566999
```

The following example will reserve two numbers from 800ForAll:

```
numbersSP~8005551212^YHN19,8885551212^YHN19;resporg~YHN19;cname~JQC;cphone~7323566999
```

Name

Reserve8fa Reserve numbers from either SMS/800 or 800ForAll

Description

Reserve8fa is used to reserve numbers in the SMS/800 or 800ForAll database. The following are the allowed fields in *reqparams*:

	npalist	A comma-separated list of NPA's: 800, 888, 877, 866, 855, 844 or 833.
	numbers	A comma-separated list of up to 10 SMS/800 numbers to be reserved.
	numbers8fa	A comma-separated list of up to 10 available 800ForAll numbers, each with it's associated Resp Org, to be reserved. The format of this list is `number^resporg,number^resporg,...`
*	**resporg**	The resporg to be associated with any numbers reserved.
*	**cname**	The contact person name.
*	**cphone**	The contact person phone number.
	notes	Descriptive notes.

*One, and only one, of **numbers** or **numbers8fa** must be specified.*

MGI Messages

If the **numbers** parameter has been specified then a **REQ-NSR** message is sent with an action code of **R**. The response message is a **RSP-NSR**.

If the parameter **numbers8fa** has been specified then no MGI message is generated.

Example

The following example will reserve two numbers from SMS/800:

```
numbers~8005551212,8885551212;resporg~YHN19;cname~JQC;cphone~7323566999
```

The following example will reserve two numbers from 800ForAll:

```
numbers8fa~8005551212^YHN19,8885551212^YHN19;resporg~YHN19;cname~JQC;cphone~7323566999
```

Name

Search Search for spare numbers

Description

Search is used to search for available numbers in either the SMS/800 or 800ForAll database. The following are the allowed fields in *reqparams*:

dn A particular toll-free number to be queried, or a number containing either ***** or **&** (i.e., a pattern or *wildcard* search).

resporg The resporg to be associated with any numbers found in the search.

> *When a user searches for toll-free numbers at Somos, Somos puts a temporary lock on any numbers found. Since the lock is associated with that resporg, only that resporg may then reserve those numbers. Locks on numbers have a short life span and then time out.*

npa The NPA of the toll-free number(s).

nxx The starting NXX of the toll-free number(s).

line The starting line of the toll-free number(s).

npalist A comma-separated list of NPA's: 800, 888, 877, 866, 855, 844 or 833.

numberword The number being searched for. The format is 1-7 characters. Valid characters include:

- alpha-numeric,
- ***** (match any character),
- **&** (repeat a digit - e.g. *&0&1* finds *7071*, *8081*)
- **%** (a second repeating digit - e.g. *&%&%* finds *7070*, *8080*, *1212*, *3434*)
- **^** (starts with), and
- **$** (ends with).

trailingdigits The number of trailing digits: 0, 1 or 2.

> *If **trailingdigits** is 0, 1 or 2 the search will return results that match up to the trailingdigits value.*
>
> *If no **trailingdigits** parameter is specified, the search is first executed with a **trailingdigits** value of **0**. If no results are returned then a second search is executed with a **trailingdigits** value of **2**.*

count The count of numbers to be searched.

contiguous	Whether or not the reserved numbers must be contiguous; value must be **0** or **1**.
numlist	If set to **1**, number lists will be created containing the numbers found to be available and those that were not. If set to **0** or omitted, no such lists are made.
nlname	The name of the number list to be created. This is only used if numlist is set to 1.
showvanity	If set to 1, if a vanity name was indicated in the **numberword** field, then the vanity name as well as the number is returned. The number and vanity name are colon-separated in the response.

*The following fields will generate a Search to SMS/800: **dn**, **npa**, **nxx**, **line**.*

*The following fields will generate a Search to 800ForAll: **npalist**, **numberword**, **trailingdigits**, **showvanity**.*

MGI Messages

For an SMS/800 query, **Search** sends a **REQ-NSR** message with an action code of **S**. The response message is a **RSP-NSR**.

The SMS/800 MGI will allow, at most, 10 numbers to be searched in a single **REQ-NSR** request. If the specified **count** is greater than ten, 8MS will automatically generate additional **REQ-NSR** messages.

Example

The following request parameters search for seven spare **877** numbers starting at NXX **555**:

```
npa~877;nxx~555;count~7
```

Name

SearchSP Search for spare numbers from 800ForAll

Description

SearchSP is used to search for available numbers in the 800ForAll database. The following are the allowed fields in *reqparams*:

npalist A comma-separated list of NPA's: 800, 888, 877, 866, 855, 844 or 833.

numberword The number being searched for. The format is 1-7 characters. Valid characters include:

- alpha-numeric,
- ***** (match any character),
- **&** (repeat a digit - e.g. *&0&1* finds *7071*, *8081*)
- **%** (a second repeating digit - e.g. *&%&%* finds *7070*, *8080*, *1212*, *3434*)
- **^** (starts with), and
- **$** (ends with).

trailingdigits The number of trailing digits: 0, 1 or 2.

> *If **trailingdigits** is 0, 1 or 2 the search will return results that match up to the trailingdigits value.*
>
> *If no **trailingdigits** parameter is specified, the search is first executed with a **trailingdigits** value of **0**. If no results are returned then a second search is executed with a **trailingdigits** value of **2**.*

count The count of numbers to be searched.

showvanity If set to 1, if a vanity name was indicated in the **numberword** field, then the vanity name as well as the number is returned. The number and vanity name are colon-separated in the response.

> *For a consistent formatting of the returned data, it is highly recommended that this parameter always be set to **1**.*

In the returned results, the vanity number will contain any overdial trailing digits.

*At least one of **npalist** or **numberword** must be specified. If neither parameter is specified, a **Search** API request to SMS/800 will be executed.*

MGI Messages

None.

Results

A SearchSP response is an "MGI-like" document embedded within the normal API response. For each number found, a response will contain the 10-digit number, the vanity number and, optionally, a Resp Org. If the Resp Org is shown, this number is coming from the special pool of 800ForAll numbers. The two or three fields in a single line of data will be separated by a colon. Each number found is separated from the next by a newline.

A typical response will appear as shown below.

```
<msgparams>
NUM=first-10-digit-number:first-vanity-number
NUM=second-10-digit-number:second-vanity-number:RespOrg
NUM=third-10-digit-number:third-vanity-number
</msgparams>
```

Example

The following request parameters search for any numbers containing the word *flowers*:

```
numberword~flowers;trailingdigits~0;showvanity~1
```

Name

Search8fa Search for spare numbers from 800ForAll

Description

Search8fa is used to search for available numbers in the 800ForAll database. The following are the allowed fields in *reqparams*:

npalist A comma-separated list of NPA's: 800, 888, 877, 866, 855, 844 or 833.

numberword The number being searched for. The format is 1-7 characters. Valid characters include:

- alpha-numeric,
- ***** (match any character),
- **&** (repeat a digit - e.g. *&0&1* finds *7071*, *8081*)
- **%** (a second repeating digit - e.g. *&%&%* finds *7070*, *8080*, *1212*, *3434*)
- **^** (starts with), and
- **$** (ends with).

trailingdigits The number of trailing digits: 0, 1 or 2.

> *If **trailingdigits** is 0, 1 or 2 the search will return results that match up to the trailingdigits value.*
>
> *If no **trailingdigits** parameter is specified, the search is first executed with a **trailingdigits** value of **0**. If no results are returned then a second search is executed with a **trailingdigits** value of **2**.*

count The count of numbers to be searched.

showvanity If set to 1, if a vanity name was indicated in the **numberword** field, then the vanity name as well as the number is returned. The number and vanity name are colon-separated in the response.

> *For a consistent formatting of the returned data, it is highly recommended that this parameter always be set to **1**.*

In the returned results, the vanity number will contain any overdial trailing digits.

*At least one of **npalist** or **numberword** must be specified. If neither parameter is specified, a **Search** API request to SMS/800 will be executed.*

MGI Messages

None.

Results

A Search8fa response is an "MGI-like" document embedded within the normal API response. For each number found, a response will contain the 10-digit number, the vanity number and, optionally, a Resp Org. If the Resp Org is shown, this number is coming from the internal pool of 800ForAll numbers. The two or three fields in a single line of data will be separated by a colon. Each number found is separated from the next by a newline.

A typical response will appear as shown below.

```
<msgparams>
NUM=first-10-digit-number:first-vanity-number
NUM=second-10-digit-number:second-vanity-number:RespOrg
NUM=third-10-digit-number:third-vanity-number
</msgparams>
```

Example

The following request parameters search for any numbers containing the word *flowers*:

```
numberword~flowers;trailingdigits~0;showvanity~1
```

Name

ScpQuery Query the status of a customer record at one or more SCPs

Description

ScpQuery requests the status of a customer record at one or more SCPs. The following are the allowed fields in *reqparams*:

*	**dn**	The number to be queried.
	ed	The effective date of the record to be queried.
	et	The effective time of the record to be queried.

MGI Messages

ScpQuery sends a **REQ-SCP** message with an action code of **Q**. The response message is a **RSP-SCP**.

Example

The following request parameters will query the status of a customer record.

```
dn~8005551212;ed~01/01/11;et~08:00A;
```

Name

ScpResend Resend a customer record to one or more SCPs

Description

ScpResend resends a customer record to one or more SCPs. The following are the allowed fields in *reqparams*:

*	**dn**	The number to be sent.
	ed	The effective date of the record to be sent.
	et	The effective time of the record to be sent.
*	**scplist**	The list of SCPs to be queried. If this parameter is not specified, all SCPs are updated.
	critical	setting this parameter to **1** indicates this resend should be performed even if an SCP is in an *overload* condition.

MGI Messages

ScpResend sends a **REQ-SCP** message with an action code of **R**. The response message is a **RSP-SCP**.

Example

The following request parameters will resend the record associated with the specified number, date and time to the SCPs *BS01* and *BS02*.

```
dn~8005551212;ed~01/01/11;et~08:00A;scplist~BS01,BS02
```

Name

SmsTemplateChange Update an SMS Template record

Description

SmsTemplateChange updates the record associated with a given SMS Template. The following are the allowed fields in *reqparams*:

* **templatename** The SMS Template containing the record, sent in the SMS/800 **tmpltnm** field. Note that a template name must begin with a * followed by the two alphabetic characters representing your SMS/800 Entity.

* **dateTime** The effective date/time of the record. The format of this value is:

```
mm/dd/yy hh:mmX/Y
```

where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C.

> *Be aware that Somos only allows schedule date/times where the minute on the quarter hour :00, :15, :30, or :45.*
>
> *Any other value for minute will cause the record to fail validation.*

This field may also contain the value **NOW** to request an immediate change.

* **resporg** The RespOrg, sent in the SMS/800 **tro** field. This field also identifies the SMS/800 Entity to which this SMS Template will belong.

interLATACarrier The numeric CIC code of the primary InterLATA carrier, sent in the SMS/800 **iec** field.

intraLATACarrier The numeric CIC code of the primary IntraLATA carrier, sent in the SMS/800 **iac** field.

planDN The dialed number serving as the source of the complex routing plan. Also see **complexRecord** below.

planTemplatename The SMS Template serving as the source of the complex routing plan. Also see **complexRecord** below.

complexRecord The name of the entry in the RPLAN or TRPLAN table that contains the complex routing for this record. If **planDN** was specified, then this value and **complexRecord** are used as an index into RPLAN. If **planTemplatename** was specified, then this value and **complexRecord** are used as an index into TRPLAN. If neither **planDN** nor **planTemplatename** are specified then **complexRecord** is assumed to be a local SMS Template plan and that value alone is used as an index into TRPLAN.

notes	Commentary to be sent in the SMS/800 **note** field. Can be up to 151 characters.
description	SMS Template description to be sent in the SMS/800 **descrip** field. Can be up to 40 characters.
lns	The number of terminating lines sent in the SMS/800 **lns** field. Format is 1 to 4 decimal numerals.
contactName	The name of the contact person, sent in the SMS/800 **ncon** field. Maximum of 30 characters.
contactPhone	The Contact Phone Number, sent in the SMS/800 **ctel** field. Must be 10 decimal numerals.
network	Network Area of Service, sent in the SMS/800 **anet** field. The value is a comma-separated list of valid SMS/800 network codes.
state	State Area of Service, sent in the SMS/800 **asta** field. The value is a comma-separated list of valid SMS/800 state codes.
areacode	Area Code Area of Service, sent in the SMS/800 **aac** field. The value is a comma-separated list of valid area codes.
lata	LATA Area of Service, sent in the SMS/800 **alat** field. The value is a comma-separated list of valid LATAs.
labels	Label Area of Service, sent in the SMS/800 **albl** field. The value is a comma-separated list of valid SMS/800 AOS label names. The SMS/800 MGI provides no means to define AOS labels; they must previously exist.
highpriority	This parameter provides access to the SMS/800 High Priority Update feature. Setting it to **1** will tell SMS to place this customer record at the front of the SCP queue. Only a small number of these updates can be done each day. After that, if this parameter is set to **1** the message will fail. You may check the Activity Log to see how many such updates remain for the current day.

MGI Messages

SmsTemplateChange sends a **REQ-TRC** message with an action code of **N** or **C**. The response message is a **RSP-TRC**.

Example

The following request parameters will send an immediate change to ***AR-CSF01**. The InterLATA carrier will be set to **0432** and the area of service to **US**:

```
templatename~*AR-
CSF01;dateTime~NOW;interLATACarrier~0432;resporg~ART01;network~US
```

Name

SmsTemplateCopy Copy and send an SMS Template record

Description

SmsTemplateCopy copies a record from a given number, 8MS Routing Set or SMS Template and sends an SMS Template record change. This provides the ability to convert a toll-free number CAD record or an 8MS Routing Set record to an SMS Template as well as the ability to copy one SMS Template (directly, by specifying a source SMS Template or indirectly by specifying a source toll-free number PAD record) to another SMS Template. The following are the allowed fields in *reqparams*:

* **source** The number, 8MS Routing Set or SMS Template containing the record to be copied. The value is interpreted according to these rules:

1. If a **sourceDateTime** is specified, the value of **source** is assumed to be a toll free number unless the source begins with a *, in which case it is assumed to be an SMS Template. The pair of values identifies the exact record from which the copy is to be made.
2. If no **sourceDateTime** is specified, the value is assumed to name an 8MS Routing Set from which the copy is to be made.
3. If no **sourceDateTime** is specified and no matching 8MS Routing Set is found, the value is assumed to be a toll free number unless the source begins with a *, in which case it is assumed to be an SMS Template. The latest **ACTIVE** or **SENDING** record for that number or SMS Template is used as the source for the copy.

If an 8MS Routing Set has been given a toll free number as its name, and the intended action is to use the toll free number as the source of a copy, **sourceDateTime** must be used to specify the exact record to be copied from. Otherwise, the 8MS Routing Set will be used as the source.

If a toll-free number has been specified as the source, and the record is a CAD record, the CAD data is copied to the destination SMS Template. If the record is a PAD record, the SMS Template referenced by this PAD record is used as the source and is copied to the destination SMS Template.

sourceDateTime The effective date/time of the source record, to be used when **source** is a toll-free number. Note that this field is *not* used for an SMS Template source. In the case where the source is an SMS Template, the current Active, Sending, most recent Pending or most recent Old record is used.

The format of this value is:

```
mm/dd/yy hh:mmX/Y
```

where X is the meridiem (A or P), and Y is the SMS/800 timezone

indicator. An example of a valid date/time would be: 07/01/98 10:15A/C.

> *Be aware that Somos only allows schedule date/times where the minute on the quarter hour :00, :15, :30, or :45.*
>
> *Any other value for minute will cause the record to fail validation.*

* **destTemplatename** The SMS Template for which the change is sent.

* **destDateTime** The effective date/time of the destination SMS Template record. The format of this value is as described for **sourceDateTime** above, and may also contain the value **NOW** to request an immediate change.

highpriority This parameter provides access to the SMS/800 High Priority Update feature. Setting it to **1** will tell SMS to place this SMS Template record at the front of the SCP queue. Only a small number of these updates can be done each day. After that, if this parameter is set to **1** the message will fail. You may check the Activity Log to see how many such updates remain for the current day.

MGI Messages

SmsTemplateCopy sends a **REQ-TRC** message with an action code of **C**. The response message is a **RSP-TRC**.

If the specified source SMS Template is not known in the 8MS database, the SMS Template information will first be retrieved from SMS/800. This requires sending a **REQ-CRV** message; the response is a **RSP-CRV**.

Example

The following request parameters will copy the routing from the SMS Template ***AR-CSF-01**, and send it for ***AR-CSF-02**, **11/15/12 02:00P:**

```
source~*AR-CSF-01;destTemplatename~*AR-CSF-02;destDateTime~11/15/12
02:00P
```

Name

SmsTemplateDelete Delete an SMS Template record

Description

SmsTemplateDelete deletes a record associated with a given SMS Template. The following are the allowed fields in *reqparams*:

* **templatename** The SMS Template containing the record to be deleted.
* **datetime** The effective date/time of the record to be deleted. The format of this value is:

  mm/dd/yy hh:mmX/Y

  where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C.

MGI Messages

SmsTemplateDelete sends a **REQ-TRC** message with an action code of **X**. The response message is a **RSP-TRC**.

Example

The following request parameters will delete the customer record for **07/01/12 10:00am** from ***AR-CSF01**:

*templatename~*AR-CSF01;datetime~07/01/12 10:00A/C*

Name

SmsTemplateDisconnect Schedule a disconnect of an active SMS Template

Description

SmsTemplateDisconnect schedules a disconnect of the accompanying active SMS Template. The following are the allowed fields in *reqparams*:

* **templatename** The SMS Template scheduled to be disconnected.

* **dateTime** The effective date/time of the record. The format of this value is:

  ```
  mm/dd/yy hh:mmX/Y
  ```

  where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C.

  > *Be aware that Somos only allows schedule date/times where the minute on the quarter hour :00, :15, :30, or :45.*
  >
  > *Any other value for minute will cause the record to fail validation.*

  This field may also contain the value **NOW** to request an immediate disconnect.

highpriority This parameter provides access to the SMS/800 High Priority Update feature. Setting it to **1** will tell SMS to place this record at the front of the SCP queue. Only a small number of these updates can be done each day. After that, if this parameter is set to **1** the message will fail. You may check the Activity Log to see how many such updates remain for the current day.

MGI Messages

SmsTemplateDisconnect sends a **REQ-TRC** message with an action code of **D**. The response message is a **RSP-TRC**.

Example

The following request parameters will send a disconnect of ***AR-CSF01** scheduled for **4/20/12 08:00A/C**:

```
templatename~*AR-CSF01;dateTime~04/20/12 08:00A/C
```

Name

SmsTemplateHistory Retrieve the change history of an SMS Template.

Description

SMS/800 maintains a database of the change history of all SMS Templates. Any time a change is made, an entry is made noting the date/time of the change and the login ID used to make the change. The information in the SMS history database is updated nightly. Thus, a change may take up to a day to be reflected in the history database.

SmsTemplateHistory provides a means to access this information via the 8MS API. The following are the allowed fields in *reqparams*:

* **TemplateName** The SMS Template to be queried.

MGI Messages

None.

Results

The result of a **SmsTemplateHistory** request (i.e., the change history of the given SMS Template) is represented as an XML document. This XML document will be contained with the **<reqparams>** tag of the normal 8MS API return structure. For example, the following might be returned from a history request:

```
<API8MS>
<reqid>OA03743771</reqid>
<message>
<routeid></routeid>
<termrpt>COMPLD</termrpt>
<errorcd>0</errorcd>
<reqparams>
<history templatename="*AR-CSF-01">
 <change>
   <activity-date>05/15/2013 13:00:01</activity-date>
   <status>OLD</status>
   <effective-date>05/06/2013 12:45:00</effective-date>
   <control-resporg>ART10</control-resporg>
   <extract-date>06/16/2013 00:01:00</extract-date>
 </change>
 <change>
   <activity-date>06/11/2013 18:23:31</activity-date>
   <status>ACTIVE</status>
   <effective-date>06/11/2013 12:00:00</effective-date>
   <control-resporg>ART10</control-resporg>
   <extract-date>06/16/2013 00:01:00</extract-date>
 </change>
</history>

</reqparams>
</message>
</API8MS>
```

Each **<change>** within the **<history>** consists of the following:

activity-date the date/time the SMS Template was activated

status	the SMS Template status at that point in time
effective-date	the date/time the SMS Template attained the given status
control-resporg	the controlling resporg of the number at that time
extract-date	the date/time that the record was retrieved from the main SMS/800 DB

Example

The following request parameter will request the history for the SMS Template ***AR-CSF-01**:

```
TemplateName~*AR-CSF-01
```

Name

SmsTemplateList Retrieve a list of SMS Template names

Description

SmsTemplateList retrieves a list of SMS Template names.

*The **SmsTemplateList** API call is not available for **EDR** customers.*

The following are the allowed fields in *reqparams*:

templatename	The starting SMS Template name to retrieve. All names that are alphabetically equal to or greater than this name will be returned.
tren	The Entity owning the SMS Templates to retrieve.

MGI Messages

SmsTemplateList is implemented by sending a **REQ-TRL** message. The response is a **RSP-TRL**.

Example

The following request parameters will retrieve all records associated with Entity **AR**:

```
tren~AR
```

Name

SmsTemplateQuery Query the status of an SMS Template record

Description

SmsTemplateQuery requests the status of the record associated with a SMS Template. The following are the allowed fields in *reqparams*:

*	**templatename**	The SMS Template to be queried.
	ed	The effective date of the record to be queried.
	et	The effective time of the record to be queried.

If **ed** and **et** are omitted, all versions of the SMS Template record are requested. If **ed** is specified and **et** is omitted, all versions with dates equal to or greater than the given **ed** are returned. If **ed** and **et** are specified, all versions with dates and times equal to or greater than the given values are returned.

MGI Messages

SmsTemplateQuery sends the **REQ-CRQ** message. The SMS/800 MGI will only return, at most, 10 records for a single **REQ-CRQ** request. In cases where more than 10 records match the request, 8MS automatically takes care of requesting the additional records, generating additional **REQ-CRQ** requests as appropriate. The response to **REQ-CRQ** is **RSP-CRQ**. In cases where one or more additional **RSP-CRQ** messages are expected, the **MORE** field in the **RSP-CRQ** message will be set to **Y**.

Example

The following request parameters will query all customer records for ***AR-CSF-01**:

```
templatename~*AR-CSF-01
```

Name

SmsTemplateRetrieve Retrieve an SMS Template record

Description

SmsTemplateRetrieve retrieves a version of an SMS Template record and stores it in the 8MS database. The following are the allowed fields in *reqparams*:

*	**templatename**	The SMS Template containing the record to be retrieved.
	ed	The effective date of the record to be retrieved.
	et	The effective time of the record to be retrieved.

The **ed** and **et** fields are optional. However, if one is specified then both must be specified. If neither one is specified, the most recent record with a status of **ACTIVE**, **SENDING** or **DISCONNECT** will be returned.

MGI Messages

SmsTemplateRetrieve sends a **REQ-CRV** message. The response message is a **RSP-CRV**.

Example

The following request parameters will retrieve the **10/01/04 10:00 am** version of the customer record associated with ***AR-CSF01**:

```
templatename~*AR-CSF01;ed~10/01/04;et~10:00A;
```

Name

SmsTemplateRetrieveAll Retrieve all records for a single SMS Template or for all SMS Templates in a single Entity

Description

SmsTemplateRetrieveAll retrieves all records for a single SMS Template or for all SMS Templates in a single Entity and stores them in the 8MS database. The following are the allowed fields in *reqparams*:

templatename The SMS Template containing the records to be retrieved.

tren The Entity containing all SMS Template records to be retrieved.

> ***tren** is not available for **EDR** customers.*

forceall Normally only records whose status appear on the list defined by the system parameter **RetrieveStatusList** will be retrieved. The **forceall** *reqparam* entry may be set to **1** to force all records to be retrieved (i.e. ignore the **RetrieveStatusList** parameter).

Note that while none of the parameters are required for this call, either **templatename** or **tren** must be specified.

MGI Messages

SmsTemplateRetrieveAll is implemented by sending one or more **REQ-CRQ** messages to find the effective date/times of all SMS Template records (returned in **RSP-CRQ**). A **REQ-CRV** message is then sent for each effective date/time. The response to **REQ-CRV** is **RSP-CRV**.

Example

The following request parameters will retrieve all records associated with ***AR-CSF01**:

```
templatename~*AR-CSF01
```

Name

SmsTemplateScpQuery Query the status of an SMS Template record at one or more SCPs

Description

SmsTemplateScpQuery requests the status of an SMS Template record at one or more SCPs. The following are the allowed fields in *reqparams*:

*	**templatename**	The SMS Template to be queried.
	ed	The effective date of the record to be queried.
	et	The effective time of the record to be queried.

MGI Messages

SmsTemplateScpQuery sends a **REQ-SCP** message with an action code of **Q**. The response message is a **RSP-SCP**.

Example

The following request parameters will query the status of an SMS Template record.

```
templatename~*AR-CSF-01;ed~01/01/11;et~08:00A;
```

Name

SmsTemplateScpResend Resend an SMS Template record to one or more SCPs

Description

SmsTemplateScpResend resends an SMS Template record to one or more SCPs. The following are the allowed fields in *reqparams*:

*	**templatename**	The SMS Template to be sent.
	ed	The effective date of the record to be sent.
	et	The effective time of the record to be sent.
*	**scplist**	The list of SCPs to be queried. If this parameter is not specified, all SCPs are updated.
	critical	setting this parameter to **1** indicates this resend should be performed even if an SCP is in an *overload* condition.

MGI Messages

SmsTemplateScpResend sends a **REQ-SCP** message with an action code of **R**. The response message is a **RSP-SCP**.

Example

The following request parameters will resend the record associated with the specified SMS Template, date and time to the SCPs *BS01* and *BS02*.

```
templatename~*AR-CSF-01;ed~01/01/11;et~08:00A;scplist~BS01,BS02
```

Name

SmsTemplateTransfer Transfer an SMS Template record

Description

SmsTemplateTransfer transfers a record associated with a given SMS Template. The following are the allowed fields in *reqparams*:

* **templatename** The SMS Template containing the record to be transferred.

* **fromDate** The source effective date/time for the transfer. The format of this value is:

  ```
  mm/dd/yy hh:mmX/Y
  ```

  where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C.

  Be aware that Somos only allows schedule date/times where the minute on the quarter hour :00, :15, :30, or :45.

  Any other value for minute will cause the record to fail validation.

* **toDate** The new effective date/time for the record. The format is as described above. This entry may also contain the value **NOW** to request a transfer to the current date/time.

MGI Messages

SmsTemplateTransfer sends a **REQ-TRC** message with an action code of **T**. The response message is a **RSP-TRC**.

Example

The following request parameters will transfer the **10/01/04 10:00am** record in ***AR-CSF01** to **10/15/04 02:00pm**.

```
templatename~*AR-CSF01;fromDate~10/01/12 10:00A/C;toDate~10/15/12
02:00P/C
```

Name

Spare8fa Spare a number from 800ForAll

Description

Spare8fa is used to spare a previously reserved number from the 800ForAll database. The following are the allowed fields in *reqparams*:

* **dn** The number to be spared.

MGI Messages

None.

Example

The following request parameters will attempt to spare the number **8005551212**:

```
dn~8005551212
```

8MS Batch Message Requests

Batch requests are those that cause messages to be sent to SMS/800 for a list of toll free numbers. The comma-separated list of numbers is specified using the **dnlist** parameter.

Batch API calls have several parameters in common. The common parameters are **rundate**, **comment**, and **flowmax**.

API batches may be scheduled. The run date/time is specified using the **rundate** parameter. The format of this value is:

```
mm/dd/yy hh:mmX
```

where X is the meridiem (A or P).

To run a batch immediately, don't use this parameter.

> *If **rundate** contains a future date/time or **timeout** is set to **0** batch requests will always return immediately with an **Information Incomplete** message. In all other cases the api request will wait until the batch completes or the **timeout** is reached.*

An API batch may have a comment associated with it. The comment is specified using the **comment** parameter. This comment will be seen in the **Batch Log** screen.

The number of messages per hour sent to SMS/800 may be limited in a batch. This is intended to avoid exceeding the per Resp Org CPU percent allocation defined by SMS/800. The maximum messages per hour is set in the parameter **flowmax**.

> *For companies that are on an **8MS Plan Type** of **Transactions**, please note the following. All Batch API calls count as one GUI transaction for each number plus a single API transaction for the call itself.*

The following batch requests are supported by the 8MS API:

BatchCopyRecord
BatchDisconnect
BatchMultiChangeResporg
BatchNumberChange
BatchNumberQuery
BatchNumberQueryAll
BatchNumberRetrieve
BatchNumberRetrieveAll
BatchNumberSpare
BatchPointerRecordChange
BatchRecordChange
BatchRecordDelete
BatchRecordQuery
BatchRecordRetrieve
BatchRecordTransfer
BatchReferralQuery
BatchReserve
BatchScpQuery
BatchScpResend

Name

BatchCopyRecord Batch customer record copy

Description

BatchCopyRecord copies a customer record from a given number, 8MS Routing Set or SMS Template and sends a record change for a group of other numbers. This provides, for example, the ability to apply an 8MS Routing Set record to a large set of toll-free numbers. The following are the allowed fields in *reqparams*:

* **source** The number, 8MS Routing Set or SMS Template containing the customer record to be copied. The value is interpreted according to these rules:

1. If a **sourceDateTime** is specified the value of **source** is assumed to be a toll free number unless the source begins with a *, in which case it is assumed to be an SMS Template. The pair of values identifies the exact record from which the copy is to be made.
2. If no **sourceDateTime** is specified, the value is assumed to name an 8MS Routing Set from which the copy is to be made.
3. If no **sourceDateTime** is specified and no matching 8MS Routing Set is found, the value is assumed to be a toll free number unless the source begins with a *, in which case it is assumed to be an SMS Template. In either case, the latest **ACTIVE** or **SENDING** record for that number or SMS Template is used as the source for the copy.

If an 8MS Routing Set has been given a toll free number as its name, and the intended action is to use the toll free number as the source of a copy, **sourceDateTime** must be used to specify the exact record to be copied from. Otherwise, the 8MS Routing Set will be used as the source.

sourceDateTime The effective date/time of the source record, to be used when **source** is a toll-free number or an SMS Template. The format of this value is:

```
mm/dd/yy hh:mmX/Y
```

where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C.

Be aware that Somos only allows schedule date/times where the minute on the quarter hour :00, :15, :30, or :45.

Any other value for minute will cause the record to fail validation.

* **dnlist** The list of numbers for which the change is sent.

* **destDateTime** The effective date/time of the destination record. The format of this

value is as described for **sourceDateTime** above, and may also contain the value **NOW** to request an immediate change.

comment A comment describing the batch operation.

flowmax The maximum number of messages per hour sent to SMS/800.

rundate The date/time when this batch will run. The format of this value is:

```
mm/dd/yy hh:mmX
```

where X is the meridiem (A or P).

To run a batch immediately, either don't use this parameter or set the value to **NOW**.

MGI Messages

BatchCopyRecord sends a **REQ-CRC** message with an action code of either **N** or **C**. The response message is a **RSP-CRC**.

If any of the specified dialed numbers are not known in the 8MS database, the number status information will first be retrieved from SMS/800. Note that the retrieve will only be done on dedicated and private sites; it will not be done on any of the service bureau sites. This requires sending a **REQ-NSR** message; the response is **RSP-NSR**.

Example

The following request parameters will retrieve the **08/01/04 10:00A** record from the 8MS Routing Set **CLASS1**, and send it for the given list of numbers with an effective date/time of **08/15/04 02:00P**.

```
source~CLASS1;dnlist~8005551212,8775551212,8885551212;
destDateTime~08/15/04 02:00P
```

See Also

CopyRecord

Name

BatchDisconnect Schedule a disconnect of a group of active numbers

Description

BatchDisconnect tells SMS/800 when to schedule a disconnect of the accompanying active numbers. The following are the allowed fields in *reqparams*:

* **dnlist** The list of numbers number to be disconnected.

* **dateTime** The effective date/time of the record. The format of this value is:

  `mm/dd/yy hh:mmX/Y`

  where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C.

  > *Be aware that Somos only allows schedule date/times where the minute on the quarter hour :00, :15, :30, or :45.*
  >
  > *Any other value for minute will cause the record to fail validation.*

  This field may also contain the value **NOW** to request an immediate disconnect.

interceptDate The End Intercept Date, sent in the SMS/800 **eint** field. Format is *MM/DD/YY*.

This field may also contain the value **NOW** to allow the number to become transitional immediately.

referralOption The Referral Option, sent in the SMS/800 **refer** field, and used for disconnect records. Allowable values are **0** and **1**.

comment A comment describing the batch operation.

flowmax The maximum number of messages per hour sent to SMS/800.

rundate The date/time when this batch will run. The format of this value is:

`mm/dd/yy hh:mmX`

where X is the meridiem (A or P).

To run a batch immediately, either don't use this parameter or set the value to **NOW**.

MGI Messages

BatchDisconnect sends a **REQ-CRC** message with an action code of **D**. The response message is a **RSP-CRC**.

Example

The following request parameters will send a disconnect to **8005551212** and **8005551213** scheduled for **4/20/04 08:00A/C** with an intercept date of **4/30/04** and a comment of "**disconnect these numbers".**

```
dnList~8005551212,8005551213;dateTime~04/20/04 08:00A/C;
interceptDate~04/30/04;comment~disconnect these numbers;
```

See Also

Disconnect

Name

BatchMultiChangeResporg Change the RespOrg of a group of toll-free numbers

Description

BatchMultiChangeResporg is used to change the RespOrg of one or more toll-free numbers.

> *Note that 8MS had another API call, **BatchChangeResporg**, which performed the same action but had the side effect of creating a new customer record. The **BatchChangeResporg** API call has been deprecated and should no longer be used.*

The following are the allowed fields in *reqparams*:

*	**dnlist**	The list of number to be changed.
*	**resporg**	The new RespOrg code.
	comment	A comment describing the batch operation.
	flowmax	The maximum number of messages per hour sent to SMS/800.
	rundate	The date/time when this batch will run. The format of this value is: `mm/dd/yy hh:mmX` where X is the meridiem (A or P). To run a batch immediately, either don't use this parameter or set the value to **NOW**.

MGI Messages

BatchMultiChangeResporg sends a **REQ-MRO** message. The response message is a **RSP-MRO**.

Example

The following request parameters will attempt to change the RespOrg of **8005551212** and **8775551212** to **ART01**:

```
dnlist~8005551212,8775551212;resporg~ART01
```

See Also

MultiChangeResporg

Name

BatchNumberQuery Batch query number information

Description

BatchNumberQuery retrieves the administrative information associated with a group of numbers. The following are the allowed fields in *reqparams*:

*	**dnlist**	The list of numbers to be queried.
	comment	A comment describing the batch operation.
	flowmax	The maximum number of messages per hour sent to SMS/800.
	rundate	The date/time when this batch will run. The format of this value is: *mm/dd/yy hh:mmX* where X is the meridiem (A or P). To run a batch immediately, either don't use this parameter or set the value to **NOW**.

MGI Messages

BatchNumberQuery sends a **REQ-NSR** message with an action code of **Q**. The response message is a **RSP-NSR**.

Example

The following request parameters will send queries for **8005551212**, **8775551212**, and **8885551212**:

```
dnlist~8005551212,8775551212,8885551212;comment~Query three numbers
```

See Also

NumberQuery

Name

BatchNumberChange Change information associated with a number

Description

BatchNumberChange is used to change the administrative information associated with a group of numbers. The following are the allowed fields in *reqparams*:

*	**dnlist**	The list of numbers to be changed.
	ru	The *reserved until* date.
	cname	The contact person name.
	cphone	The contact person phone number.
	notes	Descriptive notes.
	reserve	If set to *1* for a toll-free number with a status of *Transitional* the number may be re-reserved. If set to *0* or omitted the number is not re-reserved.
	comment	A comment describing the batch operation.
	flowmax	The maximum number of messages per hour sent to SMS/800.
	rundate	The date/time when this batch will run. The format of this value is: *mm/dd/yy hh:mmX* where X is the meridiem (A or P). To run a batch immediately, either don't use this parameter or set the value to **NOW**.

MGI Messages

BatchNumberChange sends a **REQ-NSC** message with an action code of **C**. The response message is a **RSP-NSC**.

Example

The following request parameters will update the contact name and phone number for the numbers **8005551212**, **8775439999**, and **8887776666**

```
dnlist~8005551212,8775439999,8887776666;cname~J.
Smith;cphone~7323020222
```

See Also

ChangeNumber

Name

BatchNumberQueryAll Batch query number and customer record information

Description

BatchNumberQueryAll queries the administrative information associated with a group of numbers, as well as the status of customer records for each number. The following are the allowed fields in *reqparams*:

*	**dnlist**	The list of numbers to be queried.
	comment	A comment describing the batch operation.
	flowmax	The maximum number of messages per hour sent to SMS/800.
	rundate	The date/time when this batch will run. The format of this value is: *mm/dd/yy hh:mmX* where X is the meridiem (A or P). To run a batch immediately, either don't use this parameter or set the value to **NOW**.

MGI Messages

BatchNumberQueryAll sends a **REQ-NSR** message with an action code of **Q**. The response message is a **RSP-NSR**. A **REQ-CRQ** is then sent to retrieve customer record status information. If more than 10 customer records exist for the number, additional **REQ-CRQ** messages are sent. For each **REQ-CRQ** message sent a **RSP-CRQ** message is received.

Example

The following request parameters will send queries for **8005551212**, **8775551212**, and **8885551212**:

```
dnlist~8005551212,8775551212,8885551212;comment~Query three numbers
```

See Also

NumberQueryAll

Name

BatchNumberRetrieve Batch retrieve number information

Description

BatchNumberRetrieve retrieves the administrative information associated with a group of numbers, and stores it in the 8MS database. The following are the allowed fields in *reqparams*:

*	**dnlist**	The list of numbers to be retrieved.
	comment	A comment describing the batch operation.
	flowmax	The maximum number of messages per hour sent to SMS/800.
	rundate	The date/time when this batch will run. The format of this value is: *mm/dd/yy hh:mmX* where X is the meridiem (A or P). To run a batch immediately, either don't use this parameter or set the value to **NOW**.

MGI Messages

BatchNumberRetrieve sends a **REQ-NSR** message with an action code of **Q**. The response message is a **RSP-NSR**.

Example

The following will retrieve information for **8005551212**, **8775551212**, and **8885551212**:

```
dnlist~8005551212,8775551212,8885551212;comment~Retrieve three numbers
```

See Also

NumberRetrieve

Name

BatchNumberRetrieveAll Batch retrieve number and customer record information

Description

BatchNumberRetrieveAll retrieves the administrative information associated with a group of numbers, as well as all customer records, and stores them in the 8MS database. The following are the allowed fields in *reqparams*:

*	**dnlist**	The list of numbers to be retrieved.
	forceall	Normally only customer records whose status appear on the list defined by the system parameter **RetrieveStatusList** will be retrieved. The **forceall** *reqparam* entry may be set to **1** to force all records to be retrieved (i.e. ignore the **RetrieveStatusList** parameter).
	involved	Normally, if 8MS believes the number being retrieved is not owned by the RespOrg making the request, it will not be stored. This means that records not owned, but for which the RespOrg is an involved carrier, will not be stored. The **involved** *reqparam* entry will force un-owned records to be stored.
	comment	A comment describing the batch operation.
	flowmax	The maximum number of messages per hour sent to SMS/800.
	rundate	The date/time when this batch will run. The format of this value is: `mm/dd/yy hh:mmX` where X is the meridiem (A or P). To run a batch immediately, either don't use this parameter or set the value to **NOW**.

MGI Messages

BatchNumberRetrieveAll is implemented by sending **REQ-NSR**, followed by one or more **REQ-CRQ** messages to find the effective date/times of all customer records (returned in **RSP-CRQ**). A **REQ-CRV** message is then sent for each effective date/time. The response to **REQ-CRV** is **RSP-CRV**.

Example

The following request parameters will retrieve all data associated with **8005551212**, **8775551212**, and **8885551212**:

```
dnlist~8005551212,8775551212,8885551212;comment~Retrieve three numbers
```

See Also

NumberRetrieveAll

Name

BatchNumberSpare Spare a group of numbers

Description

BatchNumberSpare is used to spare a group of previously reserved numbers. The following are the allowed fields in *reqparams*:

*	**dnlist**	The list of numbers to be spared.
	comment	A comment describing the batch operation.
	flowmax	The maximum number of messages per hour sent to SMS/800.
	rundate	The date/time when this batch will run. The format of this value is: *mm/dd/yy hh:mmX* where X is the meridiem (A or P). To run a batch immediately, either don't use this parameter or set the value to **NOW**.

MGI Messages

BatchNumberSpare sends a **REQ-NSC** with an action code of **S**. The response message is **RSP-NSC**.

Example

The following request parameters will attempt to spare **8005551212**, **8775551212**, and **8885551212**:

```
dnlist~8005551212,8775551212,8885551212;comment~Spare three numbers
```

See Also

NumberSpare

Name

BatchPointerRecordChange Update a pointer (PAD) record for a list of numbers

Description

PointerRecordChange updates the pointer (PAD) record associated with a group of numbers. The following are the allowed fields in *reqparams*:

*	**dnlist**	The list of numbers to be changed.
*	**dateTime**	The effective date/time of the record. The format of this value is: `mm/dd/yy hh:mmX/Y` where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C. *Be aware that Somos only allows schedule date/times where the minute on the quarter hour :00, :15, :30, or :45.* *Any other value for minute will cause the record to fail validation.* This field may also contain the value **NOW** to request an immediate change.
*	**templatename**	The SMS Template to which this record should point.
	orderNumber	The Service Order Number, sent in the SMS/800 **so** field. Must be from 4-13 bytes, where the first character is alphabetic, the second through twelfth characters are alphanumeric, and the thirteenth character is alphabetic.
	sfNumber	The Supplemental Form Number, sent in the SMS/800 **sf** field. Must be 1-6 alphanumeric characters. Required if Service Order Number is not specified.
	notes	Commentary to be sent in the SMS/800 **note** field. Can be up to 151 characters.
	lns	The number of terminating lines sent in the SMS/800 **lns** field. Format is 1 to 4 decimal numerals.
	duedate	The Due Date, sent in the SMS/800 **dd** field. Format is *MM/DD/YY*. If not specified, SMS/800 will generate a value.
	holdValue	The Hold Due Date indicator, sent in the SMS/800 **hdd** field. Allowable values are **0** and **1**.
	rao	The Revenue Accounting Office, sent in the SMS/800 **rao** field. Must be 3 bytes, decimal numerals. The following rules apply to this field: • If the Bill To Number (**billToNumber**) is **9999999999**, then **rao** must be **999**. If it is not specified, SMS/800 will generate a default

value of **999**.
- If complex routing is specified, then **rao** is required, and cannot be **999**.

billToNumber The Bill To, or Alternate Billing Number, sent in the SMS/800 **abn** field. Must be 10 alphanumeric characters. If **rao** is **999**, then **billToNumber** must be **9999999999**.

customer The On-Line Access Customer, sent in the SMS/800 **cus** field. Must be 5 alphanumeric characters.

agent The On-Line Agent for Customer, sent in the SMS/800 **agent** field. Must be 5 alphanumeric characters.

telco The company that sold SMS access, sent in the SMS/800 **telco** field. Must be 4 alphanumeric characters.

icecBilling The IC/EC Billing Indicator, sent in the SMS/800 **cbi** field. Must be 3 alphabetic characters for an Interexchange Carrier, 4 alphabetic characters for an Exchange Carrier.

contactName The name of the contact person, sent in the SMS/800 **ncon** field. Maximum of 30 characters.

contactPhone The Contact Phone Number, sent in the SMS/800 **ctel** field. Must be 10 decimal numerals.

interceptDate The End Intercept Date, sent in the SMS/800 **eint** field. Format is *MM/DD/YY*. An End Intercept Date is used for disconnect records. Specifying a value for **interceptDate** will cause a disconnect rather than a change to be sent.

referralOption The Referral Option, sent in the SMS/800 **refer** field, and used for disconnect records. Allowable values are **0** and **1**.

highpriority This parameter provides access to the SMS/800 High Priority Update feature. Setting it to **1** will tell SMS to place this customer record at the front of the SCP queue. Only a small number of these updates can be done each day. After that, if this parameter is set to **1** the message will fail. You may check the Activity Log to see how many such updates remain for the current day.

comment A comment describing the batch operation.

flowmax The maximum number of messages per hour sent to SMS/800.

rundate The date/time when this batch will run. The format of this value is:

mm/dd/yy hh:mmX

where X is the meridiem (A or P).

To run a batch immediately, either don't use this parameter or set the value to **NOW**.

MGI Messages

PointerRecordChange sends a **REQ-CRC** message with an action code of **N**, **C** or **D**. The response message is a **RSP-CRC**.

Example

The following request parameters will send an immediate change to **8005551212**. The SMS Template will be set to ***AR-CSF02** and the order number to **ORDER1**:

```
dnlist~8005551212,8885551212;dateTime~NOW;templatename~*AR-CSF02;
orderNumber~ORDER1
```

Name

BatchRecordChange Update a customer record for a list of numbers

Description

BatchRecordChange updates the customer record associated with a group of numbers. The following are the allowed fields in *reqparams*:

* **dnlist** — The list of numbers to be changed.

* **dateTime** — The effective date/time of the record. The format of this value is:

  `mm/dd/yy hh:mmX/Y`

  where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C.

  > *Be aware that Somos only allows schedule date/times where the minute on the quarter hour :00, :15, :30, or :45.*
  >
  > *Any other value for minute will cause the record to fail validation.*

  This field may also contain the value **NOW** to request an immediate change.

resporg — The RespOrg, sent in the SMS/800 **newro** field. This field allows a RespOrg change to be sent.

interLATACarrier — The numeric CIC code of the primary InterLATA carrier, sent in the SMS/800 **iec** field.

intraLATACarrier — The numeric CIC code of the primary IntraLATA carrier, sent in the SMS/800 **iac** field.

potsTermination — A POTS number to which SMS/800 will direct calls, sent in the SMS/800 **tel** field. Also see **interLATAPOTSValue** below.

interLATAPOTSValue — Allows InterLATA POTS to be specifed. If a value is specified in the potsTermination field, and the value of **interLATAPOTSValue** is **0**, both the POTS and Toll Free (**dialedNumber**) values will be sent as **tel**s. If the value is **1**, only the given POTS number will be sent.

orderNumber — The Service Order Number, sent in the SMS/800 **so** field. Must be from 4-13 bytes, where the first character is alphabetic, the second through twelfth characters are alphanumeric, and the thirteenth character is alphabetic.

sfNumber — The Supplemental Form Number, sent in the SMS/800 **sf** field.

	Must be 1-6 alphanumeric characters. Required if Service Order Number is not specified.
planDN	The dialed number serving as the source of the complex routing plan. Also see **complexRecord** below.
complexRecord	The name of the entry in the RPLAN table that contains the complex routing for this record. The values from **planDN** and **complexRecord** are used as an index into RPLAN.
notes	Commentary to be sent in the SMS/800 **note** field. Can be up to 151 characters.
lns	The number of terminating lines sent in the SMS/800 **lns** field. Format is 1 to 4 decimal numerals.
duedate	The Due Date, sent in the SMS/800 **dd** field. Format is *MM/DD/YY*. If not specified, SMS/800 will generate a value.
holdValue	The Hold Due Date indicator, sent in the SMS/800 **hdd** field. Allowable values are **0** and **1**.
rao	The Revenue Accounting Office, sent in the SMS/800 **rao** field. Must be 3 bytes, decimal numerals. The following rules apply to this field: • If the Bill To Number (**billToNumber**) is **9999999999**, then **rao** must be **999**. If it is not specified, SMS/800 will generate a default value of **999**. • If complex routing is specified, then **rao** is required, and cannot be **999**.
billToNumber	The Bill To, or Alternate Billing Number, sent in the SMS/800 **abn** field. Must be 10 alphanumeric characters. If **rao** is **999**, then **billToNumber** must be **9999999999**.
customer	The On-Line Access Customer, sent in the SMS/800 **cus** field. Must be 5 alphanumeric characters.
agent	The On-Line Agent for Customer, sent in the SMS/800 **agent** field. Must be 5 alphanumeric characters.
telco	The company that sold SMS access, sent in the SMS/800 **telco** field. Must be 4 alphanumeric characters.
icecBilling	The IC/EC Billing Indicator, sent in the SMS/800 **cbi** field. Must be 3 alphabetic characters for an Interexchange Carrier, 4 alphabetic characters for an Exchange Carrier.
contactName	The name of the contact person, sent in the SMS/800 **ncon** field. Maximum of 30 characters.
contactPhone	The Contact Phone Number, sent in the SMS/800 **ctel** field. Must be 10 decimal numerals.
interceptDate	The End Intercept Date, sent in the SMS/800 **eint** field. Format

	is *MM/DD/YY*. An End Intercept Date is used for disconnect records. Specifying a value for **interceptDate** will cause a disconnect rather than a change to be sent.
referralOption	The Referral Option, sent in the SMS/800 **refer** field, and used for disconnect records. Allowable values are **0** and **1**.
network	Network Area of Service, sent in the SMS/800 **anet** field. The value is a comma-separated list of valid SMS/800 network codes.
state	State Area of Service, sent in the SMS/800 **asta** field. The value is a comma-separated list of valid SMS/800 state codes.
areacode	Area Code Area of Service, sent in the SMS/800 **aac** field. The value is a comma-separated list of valid area codes.
lata	LATA Area of Service, sent in the SMS/800 **alat** field. The value is a comma-separated list of valid LATAs.
labels	Label Area of Service, sent in the SMS/800 **albl** field. The value is a comma-separated list of valid SMS/800 AOS label names. The SMS/800 MGI provides no means to define AOS labels; they must previously exist.
directoryAssistanceType	The Directory Assistance Type, sent in the SMS/800 **dat** field. The valid values are **N** (Normal), **G** (Government), and **F** (Frequently Called).
updateValue	The Directory Assistance Update indicator, sent in the SMS/800 **dau** field. The valid values are **0** and **1**.
listingType	The Listing Type, sent in the SMS/800 **li** field. The valid values are **BL** (BLocked), **LI** (Published), and **NP** (Non-published).
address	The Listing Address, sent in the SMS/800 **la** field. Can be up to 75 characters.
names	The Listing Names, sent in the SMS/800 **ln** field. Up to nine listing names can be specified, each up to 75 characters, separated by the **~** character.
highpriority	This parameter provides access to the SMS/800 High Priority Update feature. Setting it to **1** will tell SMS to place this customer record at the front of the SCP queue. Only a small number of these updates can be done each day. After that, if this parameter is set to **1** the message will fail. You may check the Activity Log to see how many such updates remain for the current day.
comment	A comment describing the batch operation.
flowmax	The maximum number of messages per hour sent to SMS/800.
rundate	The date/time when this batch will run. The format of this value is:

mm/dd/yy hh:mmX

where X is the meridiem (A or P).

To run a batch immediately, either don't use this parameter or set the value to **NOW**.

MGI Messages

BatchRecordChange sends a **REQ-CRC** message with an action code of **N**, **C** or **D**. The response message is a **RSP-CRC**.

Example

The following request parameters will send an immediate change for **8005551212**, **8775551212**, and **8885551212**, updating the InterLATA carrier:

```
dnlist~8005551212,8775551212,8885551212;interLATACarrier~0432;
comment=Change 3 carriers;
```

See Also

RecordChange

Name

BatchRecordDelete Delete a group of customer records

Description

BatchRecordDelete deletes customer records associated with a group of numbers. The following are the allowed fields in *reqparams*:

*	**dnlist**	The list of numbers containing the records to be deleted.
*	**datetime**	The effective date/time of the records to be deleted. The format of this value is: `mm/dd/yy hh:mmX/Y` where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C.
	comment	A comment describing the batch operation.
	flowmax	The maximum number of messages per hour sent to SMS/800.
	rundate	The date/time when this batch will run. The format of this value is: `mm/dd/yy hh:mmX` where X is the meridiem (A or P). To run a batch immediately, either don't use this parameter or set the value to **NOW**.

MGI Messages

BatchRecordDelete sends a **REQ-CRC** message with an action code of **X**. The response message is a **RSP-CRC**.

Example

The following request parameters will delete the customer records for **07/01/04 10:00am** from **8005551212** and **8005551213** with the comment "**delete these numbers**".

```
dnlist~8005551212,8005551213;datetime~07/01/04 10:00A/C;comment~delete
these numbers;
```

See Also

RecordDelete

Name

BatchRecordQuery Query the status of customer records for a group of numbers

Description

BatchRecordQuery requests the status of all customer records associated with a group of numbers. The following are the allowed fields in *reqparams*:

*	**dnlist**	The list of numbers to be queried.
	comment	A comment describing the batch operation.
	flowmax	The maximum number of messages per hour sent to SMS/800.
	rundate	The date/time when this batch will run. The format of this value is: `mm/dd/yy hh:mmX` where X is the meridiem (A or P). To run a batch immediately, either don't use this parameter or set the value to **NOW**.

MGI Messages

BatchRecordQuery sends the **REQ-CRQ** message. The SMS/800 MGI will only return, at most, 10 records for a single **REQ-CRQ** request. In cases where more than 10 records match the request, 8MS automatically takes care of requesting the additional records, generating additional **REQ-CRQ** requests as appropriate. The response to **REQ-CRQ** is **RSP-CRQ**. In cases where one or more additional **RSP-CRQ** messages are expected, the **MORE** field in the **RSP-CRQ** message will be set to **Y**.

Example

The following request parameters will send queries for **8005551212**, **8775551212**, and **8885551212**:

```
dnlist~8005551212,8775551212,8885551212;comment~Query three numbers
```

See Also

RecordQuery

Name

BatchRecordRetrieve Retrieve a customer record for a list of numbers

Description

BatchRecordRetrieve retrieves a version of customer records for a group of numbers and stores them in the 8MS database. The most recent **ACTIVE**, **SENDING**, or **DISCONNECT** record in each of the given numbers will be retrieved. The following are the allowed fields in *reqparams*:

*	**dnlist**	The list of numbers to be retrieved.
	comment	A comment describing the batch operation.
	flowmax	The maximum number of messages per hour sent to SMS/800.
	rundate	The date/time when this batch will run. The format of this value is: `mm/dd/yy hh:mmX` where X is the meridiem (A or P). To run a batch immediately, either don't use this parameter or set the value to **NOW**.

MGI Messages

BatchRecordRetrieve sends a **REQ-CRV** message. The response message is a **RSP-CRV**.

Example

The following request parameters will retrieve one record for **8005551212**, **8775551212**, and **8885551212**:

```
dnlist~8005551212,8775551212,8885551212;comment~Retrieve 3 records
```

See Also

RecordRetrieve

Name

BatchRecordTransfer Transfer a set of customer records

Description

BatchRecordTransfer transfers customer records associated with a group of numbers. The following are the allowed fields in *reqparams*:

*	**dnlist**	The number containing the customer record to be transferred.
*	**fromDate**	The source effective date/time for the transfer. The format of this value is: `mm/dd/yy hh:mmX/Y` where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/98 10:15A/C. *Be aware that Somos only allows schedule date/times where the minute on the quarter hour :00, :15, :30, or :45.* *Any other value for minute will cause the record to fail validation.*
*	**toDate**	The new effective date/time for the record. The format is as described above. This entry may also contain the value **NOW** to request a transfer to the current date/time.
	comment	A comment describing the batch operation.
	flowmax	The maximum number of messages per hour sent to SMS/800.
	rundate	The date/time when this batch will run. The format of this value is: `mm/dd/yy hh:mmX` where X is the meridiem (A or P). To run a batch immediately, either don't use this parameter or set the value to **NOW**.

MGI Messages

BatchRecordTransfer sends a **REQ-CRC** message with an action code of **T**. The response message is a **RSP-CRC**.

Example

The following request parameters will transfer the **10/01/04 10:00am** records in **8005551212** and **8005551213** to **10/15/04 02:00pm**.

```
dnlist~8005551212,8005551213;fromDate~10/01/04
10:00A/C;toDate~10/15/04 02:00P/C
```

See Also

RecordTransfer

Name

BatchReferralQuery Query a trouble referral number for a list of numbers

Description

BatchReferralQuery queries the trouble referral number for each entry in a group of numbers. The following are the allowed fields in *reqparams*:

*	**dnlist**	The list of numbers to be queried.
	comment	A comment describing the batch operation.
	flowmax	The maximum number of messages per hour sent to SMS/800.
	rundate	The date/time when this batch will run. The format of this value is: *mm/dd/yy hh:mmX* where X is the meridiem (A or P). To run a batch immediately, either don't use this parameter or set the value to **NOW**.

MGI Messages

BatchReferralQuery sends a **REQ-TRN** message. The response message is a **RSP-TRN**.

Example

The following request parameters will query the trouble referral numbers for **8005551212**, **8775551212**, and **8885551212**:

```
dnlist~8005551212,8775551212,8885551212;comment~Query three numbers
```

See Also

ReferralQuery

Name

BatchReserve Reserve a list of numbers

Description

BatchReserve is used to reserve numbers in the SMS/800 database. The following are the allowed fields in *reqparams*:

*	**dnlist**	The list of numbers to be reserved.
*	**resporg**	The resporg to be associated with any numbers reserved.
*	**cname**	The contact person name.
*	**cphone**	The contact person phone number.
	notes	Descriptive notes.
	numlist	If set to **1**, number lists will be created containing the numbers successfully reserved, and those that were not reserved. If set to **0** or omitted, no such lists are made.
	nlname	The name of the number list to be created. This is only used if numlist is set to 1.
	comment	A comment describing the batch operation.
	flowmax	The maximum number of messages per hour sent to SMS/800.
	rundate	The date/time when this batch will run. The format of this value is: *mm/dd/yy hh:mmX* where X is the meridiem (A or P). To run a batch immediately, either don't use this parameter or set the value to **NOW**.
	custname	This field is used for individual company auditing and tracking purposes, allowing a company to track the customer for which this number was reserved. This field is stored on 8MS only; it is not sent to SMS/800.
	btn	This field is used for individual company auditing and tracking purposes, allowing a company to track the billing telephone number for which this number was reserved. This field is stored on 8MS only; it is not sent to SMS/800.
	onum_sreq	This field is used for individual company auditing and tracking purposes, allowing a company to track the order number or service request for which this number was reserved. This field is stored on 8MS only; it is not sent to SMS/800.

MGI Messages

BatchReserve sends a **REQ-NSR** message with an action code of **R**. The response message is a **RSP-NSR**.

The SMS/800 MGI will allow, at most, 10 numbers to be reserved in a single **REQ-NSR** request. If more than ten numbers are requested, 8MS will automatically generate additional **REQ-NSR** messages.

Example

The following request parameters will attempt to reserve the numbers **8002060095**, **80020664648**, **8002224262**, **8002230606**:

```
dnlist~8002060095,80020664648,8002224262,8002230606;
resporg~ART01;cname~John Smith;cphone~7323566999
```

See Also

Reserve

Name

BatchScpQuery Query the status of a customer record at one or more SCPs for a group of numbers

Description

BatchScpQuery requests the status of the customer record associated with a group of numbers at one or more SCPs. The following are the allowed fields in *reqparams*:

*	**dnlist**	The list of numbers to be queried.
	ed	The effective date of the record to be queried.
	et	The effective time of the record to be queried.

MGI Messages

BatchScpQuery sends a **REQ-SCP** message with an action code of **Q**. The response message is a **RSP-SCP**.

Example

The following request parameters will query the status of a customer record for a set of toll-free numbers.

```
dnlist~8005551212,8885551212;ed~01/01/11;et~08:00A;
```

Name

BatchScpResend Resend a customer record to one or more SCPs for a group of numbers

Description

BatchScpResend resends a the customer record associated with a group of numbers to one or more SCPs. The following are the allowed fields in *reqparams*:

*	**dnlist**	The list of numbers to be sent.
	ed	The effective date of the record to be sent.
	et	The effective time of the record to be sent.
*	**scplist**	The list of SCPs to be queried. If this parameter is not specified, all SCPs are updated.
	critical	This parameter provides access to the SMS/800 Critical Update feature, which is similar in intent to the SMS/800 High Priority Update feature. Setting it to **1** will tell SMS to place this customer record at the front of the SCP queue. Only a small number of these updates can be done each day. After that, if this parameter is set to **1** the message will fail.

MGI Messages

BatchScpResend sends a **REQ-SCP** message with an action code of **R**. The response message is a **RSP-SCP**.

Example

The following request parameters will resend the record associated with the specified number, date and time to the SCPs *BS01* and *BS02*.

```
dnlist~8005551212,8885551212;ed~01/01/11;et~08:00A; scplist~BS01,BS02
```

Miscellaneous Requests

Miscellaneous requests are those 8MS API messages that are not necessarily associated with specific SMS/800 MGI messages. Some of these messages use the SMS/800 history database, while others are independent of SMS/800 altogether.

The following miscellaneous requests are supported by the 8MS API:

CountSpares
Cprgen
GetSpares
SmsContacts
SpellNumber
SearchLocalCarrierExpress
SearchLocalNum
SearchLocalRec
UsageCounts

Name

CountSpares Count the amount of upcoming spare toll-free numbers.

Description

Spare Number Forecast is a feature that uses the SMS/800 History database to determine what numbers will be coming spare in the near future.

CountSpares counts how many toll-free numbers will be coming spare in the near future. The search may be restricted to specific NPAs or toll-free numbers. SMS/800 wildcard patterns (* and &) may be used in the toll-free number field. Start and end dates may be used to restrict the search even more. The following are the allowed fields in *reqparams*:

npalist A comma-separated list of NPAs to be searched.

dn A single toll-free number or a number pattern. A number pattern contains alphanumerics plus the standard SMS/800 wildcard characters, * and &, plus two additional pattern match characters, ^ and $.

This parameter takes up to seven characters. Note that the NPA is *not* included in this parameter. If less than seven characters are provided the search function will assume the pattern may appear anywhere in the number. Putting a ^ at the start of the pattern indicates that the pattern *must* appear at the start of the number. Putting a $ at the end of the pattern indicates that the pattern *must* appear at the end of the number.

startdate The earliest expected spare date. The format of this value is:

mm/dd/yy

enddate The latest expected spare date. The format of this value is:

mm/dd/yy

trailingdigits The number of digits (0-2) that may be added on to the number to allow longer spelled words. Words may then be up to 9 letters long excluding the NPA. This parameter is optional.

resporg A full or partially specified resporg. This limits the search to numbers whose current resporg matches this resporg. This parameter is optional.

MGI Messages

None.

Results

A CountSpares response is an XML document embedded within the normal API response. A typical XML response will have the following fields.

```
<msgparams>
<countspares>
<npadata>
<npa>first NPA</npa>
<count>amount found for first NPA</count>
</npadata>
<npadata>
<npa>second NPA</npa>
<count>amount found for second NPA</count>
</npadata>
</countspares>
</msgparams>
```

Example

The following request parameters will request a count of all **877** numbers coming spare between **May 1** and **May 7, 2007**.

```
npalist~877;startdate~05/01/07;enddate~05/07/07;
```

Name

Cprgen Request generation of a CPR

Description

Cprgen 8MS offers an add-on feature called **CPRgen**. It provides a means for automatically generating SMS/800 complex routing (CPR) from user supplied input. For a description of this input, as well as details on the tool itself, see the CPRgen section in the *8MS User's Guide*. At a high level, the user will upload a CPR description, requesting that a CPR of a given name be generated for a given number. If CPR generation is successful, there is also an option to automatically activate the new routing in SMS/800.

The **Cprgen** request provides a means to request CPR generation via the 8MS API. The following are the allowed fields in *reqparams*:

*	**cprdata**	The blob of text that defines the routing to be generated. Details of this format are described in the **CPRGen** page of the User Guide.
	notes	Optional text describing the request.

MGI Messages

None.

Results

The result of a **Cprgen** request is represented in the **<msgparams>** tag of the normal 8MS API return structure. The parameters indicate the numbers of CPRs successfully generated, the number of failed generation requests, and, if activation was requested, how many activation requests were sent. If the number of activation requests is non-zero, then the SMS/800 responses to the activation requests are also returned.

For example, the following might be returned from a **Cprgen** request:

```
<API8MS>
<reqid>OA03743771</reqid>
<message>
<routeid>lt;/routeid>
<termrpt>COMPLD</termrpt>
<errorcd>0</errorcd>
<msgparams>
succeeded=1
failed=0
activate=1
</msgparams>
</message>
<message>
<routeid>RXU</routeid>
<termrpt>DENIED</termrpt>
<errorcd>1</errorcd>
<msgparams>
ID=ART21000
RO=ART01
NUM=8667736580
ED=NOW
ET=
CNT=04
ERR=1006
VERR=IEC:0123
```

```
ERR=1006
VERR=IEC:0124
ERR=1005
VERR=IAC:0123
ERR=1005
VERR=IAC:0124
</msgparams>
</message>
</API8MS>
```

Example

The following parameters will request generation of a CPR.

```
reqparams=cprdata~

-START NUM-

-START HDR-
Type:6DIGIT
Target#:8662998932
Cprname:BSO Test
Activate:Y
Effdate:11/11/02 10:00A
-END HDR-

-START CPR-
C:222:609332,609333,609334,732456,732649,732655,732889,908233,
      908434,908883,908924,908994
C:444:201338,201336,201998,605552,701898,701899,904332
C:555:212
C:110/9085550123:732445,732988
B:OBA:908312,908322
O:555
-END CPR-

-END NUM-
```

Name

GetSpares Get a list of upcoming spare toll-free numbers.

Description

Spare Number Forecast is a feature that uses the SMS/800 History database to determine what numbers will be coming spare in the near future.

GetSpares retrieves a list of toll-free numbers that will be coming spare in the near future. The search may be restricted to specific NPAs or toll-free numbers. SMS/800 wildcard patterns (* and &) may be used in the toll-free number field. Start and end dates may be used to restrict the search even more. The following are the allowed fields in *reqparams*:

npalist A comma separated list of NPAs to be searched.

dn A single toll-free number or a number pattern. A number pattern contains alphanumerics plus the standard SMS/800 wildcard characters, * and &, plus two additional pattern match characters, ^ and $.

This parameter takes up to seven characters. Note that the NPA is *not* included in this parameter. If less than seven characters are provided the search function will assume the pattern may appear anywhere in the number. Putting a ^ at the start of the pattern indicates that the pattern *must* appear at the start of the number. Putting a $ at the end of the pattern indicates that the pattern *must* appear at the end of the number.

startdate The earliest expected spare date. The format of this value is:

mm/dd/yy

enddate The latest expected spare date. The format of this value is:

mm/dd/yy

order The ordering of the list of numbers. Valid values are *telnumber*, *resporg*, *sparedate*.

trailingdigits The number of digits (0-2) that may be added on to the number to allow longer spelled words. Words may then be up to 9 letters long excluding the NPA. This parameter is optional.

resporg A full or partially specified resporg. This limits the search to numbers whose current resporg matches this resporg. This parameter is optional.

retrievelimit The maximum amount of numbers to retrieve. If the value specified exceeds the system limit, the system limit is used. This parameter is optional.

MGI Messages

None.

Results

A GetSpares response is an XML document embedded within the normal API response. A typical XML response will have the following fields.

```
<msgparams>
<getspares>
<count>amount found</count>
<numberdata>
<dn>first toll-free number</dn>
<resporg>resp org</resporg>
<sparedate>date to go spare</sparedate>
</numberdata>
<numberdata>
<dn>second toll-free number</dn>
<resporg>resp org</resporg>
<sparedate>date to go spare</sparedate>
</numberdata>
</getspares>
</msgparams>
```

Example

The following request parameters will retrieve a list of all **877** numbers coming spare between **May 1** and **May 7, 2007**.

```
npalist~877;startdate~05/01/07;enddate~05/07/07;
```

Name

SmsContacts Get a list of SMS/800 Resp Org Primary Contact and Change Contact information.

Description

SMS/800 provides a list of Resp Org contacts to allow one Resp Org owner to contact another. This contact list is normally only viewable at the SMS/800 web site (not the user interface). 8MS stores a snapshot of this contact data to allow easy access to the data from within 8MS. This snapshot is updated on a regularly scheduled basis.

SmsContacts retrieves one or more contacts from the 8MS snapshot of the SMS/800 Resp Org contacts list. The following are the allowed fields in *reqparams*:

name	A full or partially specified contact name. This limits the search to contacts whose name matches this name.
company	A full or partially specified company name. This limits the search to contacts whose company name matches this company name.
resporg	A full or partially specified resporg. This limits the search to contacts whose resporg matches this resporg.
type	The contact type. Valid values are *primary* or *change*.

MGI Messages

None.

Results

An SmsContacts response is an XML document embedded within the normal API response. A typical XML response will have the following fields.

```
<SmsContacts>
<Filters>
<Name></Name>
<Company></Company>
<Resporg>ART01</Resporg>
<Type></Type>
</Filters>
<NumRecs>1</NumRecs>
<LastLoadDate>09/24/09 08:59AM</LastLoadDate>
<Contacts>
<Contact>
<Name>John Smith</Name>
<Company>Smith Corporation</Company>
<Address>Smith Corporation;270 Davidson Avenue;Somerset, New Jersey
08873</Address>
<PhoneNum>732-302-0222</PhoneNum>
<TroubleNum>732-356-6999</TroubleNum>
<FaxNum>732-302-0799</FaxNum>
<EmailAddr>ssl@csfcorp.com</EmailAddr>
<RespOrg>ART01</RespOrg>
<ContactType>Primary</ContactType>
```

```
<Notes>Please send email to 8mscust@iconectiv.com for NPA
800</Notes>
<RocParticipant>Yes</RocParticipant>
</Contact>
</Contacts>
</SmsContacts>
```

Example

The following request parameters will retrieve the contact for Resp Org **ART01**.

```
resporg~art01
```

Name

SearchLocalCarrierExpress Retrieve information about a toll-free number's records for a specific carrier.

Description

SearchLocalCarrierExpress provides the ability to retrieve certain information in a toll-free number's carrier-specific records. This includes the date/time of a record plus the carrier-specific features defined for that record. The search is limited to numbers owned by the API user's company. In addition to information about a number's individual records, a count of records is also provided. The following are the allowed fields in *reqparams*:

*	**carrier**	The carrier for which data will be queried. Valid values are: • ATT • Lumen-5102 • Sinch • Verizon-0222
*	**dn**	The toll-free number to be queried.
	latestdate	Only return information related to the latest record. Setting **latestdate** to **y** will retrieve only the latest record. Setting this param to any other value or not specifying it indicates that all records should be returned.

The timeout parameter, available to all API calls, should not be set to less than 30 seconds for this call since this call does not support retrieving data after a timeout occurs.

MGI Messages

None.

Results

The result of a **SearchLocalCarrierExpress** request is represented as an XML document. This XML document will be contained within the *<msgparams>* tag of the normal 8MS API return structure. For example, the following might be returned from a request:

```
<msgparams>
<SearchLocalCarrierExpress>
<RecordsTotal>1</RecordsTotal>
<Verizon-0222>
<Record>
<Number>855-711-8583</Number>
<RequestDateTime>11/21/11 07:15A</RequestDateTime>
<ResponseDateTime>11/23/11 07:15A</ResponseDateTime>
<Resporg>ART01</Resporg>
<PlanId>1</PlanId>
<CorpId>1</CorpId>
<Record>
<Verizon-0222>
<SearchLocalCarrierExpress>
```

```
</msgparams>
```

Example

The following request parameter will request the record date/time and CIC list for the toll-free number in the *Active* status:

```
dn~8005551212;carrier~ATT
```

Name

SearchLocalNum Retrieve information about one or more toll-free numbers.

Description

SearchLocalNum provides the ability to retrieve status information about one or more numbers based on search parameters. The search is limited to numbers owned by the api user's company. In addition to information about individual numbers, a count of numbers by status is also provided. Any status with a count of zero numbers will not be shown. The following are the allowed fields in *reqparams*:

dn	A specific toll-free number. Setting this parameter limits the search to just this number.
resporg	The controlling RespOrg. Setting this parameter limits the search to numbers controlled by this RespOrg.
resporgisnot	When **resporg** is set, setting **resporgisnot** to 1 limits the search to numbers *not* controlled by the specified RespOrg (but still within the api user's company). Setting **resporgisnot** to 0 is equivalent to not providing it. **resporg** then behaves as described above.
statuslist	A comma separated list of 8MS number status values. Setting this parameter limits the search to numbers in the one or more specified status values. Values should match the status words (case insensitive) shown on the **Search Local Number** user interface screen. The values include: *Assigned*, *Disconnected*, *Reserved*, *Spare*, *Suspended*, *Transitional*, *Unavailable*, *Working*. A default status of *Working* is set if no *statuslist* is specified.
statusisnot	When **statuslist** is set, setting **statusisnot** to 1 limits the search to numbers *not* in the specific list of status values. Setting **statusisnot** to 0 is equivalent to not providing it. **statuslist** then behaves as described above.
maxnumbers	The maximum amount of numbers to return. Setting this parameter restricts the amount of numbers to return, even if more numbers match. The first set of matching numbers found will be returned. The resulting XML contains two fields related to this. *NumberTotal* shows the total amount of numbers that matched the query. If **maxnumbers** has been specified and is less than *NumberTotal* then *NumbersReturned* will equal **maxnumbers**. Otherwise *NumbersReturned* will equal *NumberTotal*.
showcounts	Setting this parameter causes the data returned to include **only** the count of numbers, by status. The actual numbers and related data are not returned.

The timeout parameter, available to all API calls, should not be set to less than 30 seconds for this call since this call does not support retrieving data after a timeout occurs.

Note that while none of the parameters are required for this call, at least one parameter must be specified. This does not include **maxnumbers**; if **maxnumbers** is specified another parameter must be specified.

MGI Messages

None.

Results

The result of a **SearchLocalNum** request is represented as an XML document. This XML document will be contained within the *<msgparams>* tag of the normal 8MS API return structure. For example, the following might be returned from a request:

```
<msgparams>
<SearchLocalNum>
<ShowCounts>
<Reserved>1</Reserved>
<Working>1</Working>
</ShowCounts>
<NumberTotal>2</NumberTotal>
<NumbersReturned>2</NumbersReturned>
<LocalNumbers>
<Numbers dn="800-555-1212">
<Status>Working>/Status>
<StatusDate>01/03/96</StatusDate>
<Resporg>ART01</Resporg>
<LastActive></LastActive>
<EndDate></EndDate>
</Numbers>
<Numbers dn="866-555-1212">
<Status>Reserved</Status>
<StatusDate>01/13/10</StatusDate>
<Resporg>ART01</Resporg>
<LastActive>02/10/05</LastActive>
<EndDate>02/27/10</EndDate>
</Numbers>
</LocalNumbers>
</SearchLocalNum>
</msgparams>
```

Each *<Numbers>* tag within the *<LocalNumbers>* tag consists of the following fields:

Numbers dn=*tfn*	The *dn* attribute of the *Numbers* tag contains the actual toll-free number.
Status	The status of the number.
StatusDate	The date and time the number attained the given status.
Resporg	The controlling RespOrg of this number.
LastActive	The date and time this number was last active.
EndDate	The date when this number will be returned to the *Spare* status. End dates may only be set for *Reserved* and *Transitional* numbers.

Example

The following request parameter will request all numbers in the *Reserved* status:

```
statuslist~reserved
```

Name

SearchLocalRec Retrieve information about a toll-free number's records.

Description

SearchLocalRec provides the ability to retrieve certain information in a toll-free number's records. This includes the date/time, status, Area of Service, SMS Template name and list of CICs specified for a record. The search is limited to numbers owned by the API user's company. In addition to information about a number's individual records, a count of records is also provided. The following are the allowed fields in *reqparams*:

*	**dn**	The toll-free number to be queried.
	statuslist	A comma separated list of 8MS number status values. Setting this parameter limits the search to records in the one or more specified status values. Values should match the status words (case insensitive) shown on the **Search Local Record** user interface screen. The values include: *Active*, *Disconnect*, *Hold*, *Invalid*, *Old*, *Pending*, *Saved*, *Sending*, *MustCheck*, *Failed*, *Rejected*. A default status of *Active* is set if no *statuslist* is specified.
	latestdate	Only return information related to the latest record. Setting **latestdate** to **y** will retrieve only the latest record. Setting this param to any other value or not specifying it indicates that all records should be returned.
	showcic	Setting **showcic** to **1** indicates that all CICs used in the routing of this number's record should be returned. A list of CICs is returned. Setting this param to any other value or not specifying it indicates that the list of CICs should not be returned.

The timeout parameter, available to all API calls, should not be set to less than 30 seconds for this call since this call does not support retrieving data after a timeout occurs.

MGI Messages

None.

Results

The result of a **SearchLocalRec** request is represented as an XML document. This XML document will be contained within the *<msgparams>* tag of the normal 8MS API return structure. For example, the following might be returned from a request:

```
<msgparams>
<SearchLocalRec>
<RecordsTotal>1</RecordsTotal>
<RecordsReturned>1</RecordsReturned>
<LocalRecords>
<Records>
<Number>855-711-8583</Number>
<DateTime>11/21/11 07:15A</DateTime>
```

```
<Status>Active</Status>
<AosLabel><AosLabel>
<AosAreaCode><AosAreaCode>
<AosLata><AosLata>
<AosNetwork>US<AosNetwork>
<AosState><AosState>
<TemplateName><TemplateName>
<CicList>
<Cic>0444</Cic>
<CicList>
<Records>
<LocalRecords>
<SearchLocalRec>
</msgparams>
```

Example

The following request parameter will request the record date/time and CIC list for the toll-free number in the *Active* status:

```
dn~8005551212;statuslist~active;showcic~1
```

Name

UsageCounts Get the current Usage Counts for your company.

Description

The **Usage Counts** screen in the user interface allows a user to determine the current and maximum amount of numbers stored on 8MS for any particular day or month as well as all transaction counts for Local Deletes, API calls, Test Calls and all Other transactions.

The **UsageCounts** API call allows the count information be be retrieved. The details that the **Usage Counts** screen returns are not available; only counts are available. If no parameters are provided, the current month's data is returned. Additionally, a *BillingInfo* section is included which shows the maximum values allowed for your company. If any parameters are specified, the *BillingInfo* section is not included. The following are the allowed fields in *reqparams*:

startdate	The starting date for viewing usage count data. If *startdate* is specified but *enddate* is not, enddate defaults to the current date.
enddate	The ending date for viewing usage count data. If *enddate* is specified but *startdate* is not, *enddate* is ignored and the current month's data is returned. Note, however, that the *BillingInfo* section is not included in this case.

MGI Messages

None.

Results

A UsageCounts response is an XML document embedded within the normal API response. A typical XML response will have the following fields.

```
<msgparams>
<UsageCounts>
<StartDate>01/01/09 12:00A<StartDate>
<EndDate>01/23/09 11:05A<EndDate>
<BillingInfo>
<NumbersAllocation>1000<NumbersAllocation>
<APITransactionsAllocation>5000<APITransactionsAllocation>
<LocalDeleteTransactionsAllocation>500<LocalDeleteTransactionsAllocation>
<TestCallTransactionsAllocation>0<TestCallTransactionsAllocation>
<GUITransactionsAllocation>5000<GUITransactionsAllocation>
<BillingInfo>
<NumbersCurrent>985<NumbersCurrent>
<NumbersMaximum>988<NumbersMaximum>
<TestCalls>0<TestCalls>
<APITransactions>0<APITransactions>
<LocalDeleteTransactions>23<LocalDeleteTransactions>
<GUITransactions>430<GUITransactions>
<UsageCounts>
</msgparams>
```

Example

The following request parameters will retrieve the usagecounts data between **June 1** and **July 15, 2008**.

```
startdate~06/01/08;enddate~07/15/08;
```

Carrier Express Message Requests

Carrier Express is a system that allows toll-free numbers to be activated and disconnected on carriers' networks. Carrier Express message requests send messages to Carrier Express which sends the requests on to a single carrier's network. The following single-number and batch Carrier Express requests are supported by the 8MS API:

CarrierExpressActivate	**BatchCarrierExpressActivate**
CarrierExpressChange	**BatchCarrierExpressChange**
CarrierExpressDisconnect	**BatchCarrierExpressDisconnect**

*For support of AT&T within Carrier Express, **Profile** definitions are created within Carrier Express, and accessible from within the 8MS GUI and API. A Profile contains a set of provisioning values. The Carrier Express API calls accept either a **profilename** or the set of values, allowing companies to simplify their API calls and to be certain that the provisioning values for a set of numbers are consistent, whether these numbers are provisioned as a batch or separately.*

A single Profile contains the following fields:

- *enterpriseid*
- *fqdn*
- *distgroup*
- *payphoneind*
- *billingtype*
- *edge*
- *intelivr*
- *callrecording*
- *callverify*
- *smmktbusiness*
- *hybrid*
- *skyvera*
- *tfservicetype*
- *mcn*
- *lso*
- *apn*
- *rao*
- *routingnumber*
- *groupsize*

*When specifying profilename in an API call, **none** of the above parameters may be included in that API call. To be clear, the API does **not** support overriding any values defined in a Profile by specifying those parameters in conjunction with a profilename.*

*All other parameters defined in an API call (i.e. those not in a Profile, such as **contactname** and **contactphone**) are not part of a Profile and must be specified in the API call.*

Name

CarrierExpressActivate Send an activate request for a single toll-free number to a single carrier via the Carrier Express system.

Description

CarrierExpressActivate sends an activate request to a single carrier for a single toll-free number. The following are the allowed fields in *reqparams*:

*	**carrier**	The carrier on which the specified number is to be activated. Valid values are: • ATT • Lumen-5102 • Sinch • Verizon-0222
*	**dn**	The toll-free number to be activated.

The following parameters are used only when **carrier** is set to *ATT*.

*	**action**	The type of service to be activated. Valid values are: • iptf360
*	**profilename**	Name of profile containing a group of shared values used in place of individual values.
*	**contactname**	The contact name for this toll-free number. Its maximum size is 30 characters.
*	**contactphone**	The contact phone number for this toll-free number.
	clientorderno	The purchase order number/client order number for this toll-free number. Its maximum size is 21 characters.
*	**payphoneind**	Indicates whether or not payphone calls are blocked. Valid values are upper-case *Y* or *N*.
*	**resporg**	The RespOrg of the toll-free number.

The following parameters are used only when **carrier** is set to *Lumen-5102*.

	customernumber	The Lumen customer number. This parameter is required for this carrier.
	billingaccount	The Lumen billing account. This parameter is required for this carrier.
	billingtelno	The billing telephone number which must be a valid POTS number. This parameter is required for this carrier.

resellerclientid The Reseller Client ID.

servicearea The network area of service. Valid values are:

- US
- XA
- XB
- XC

This parameter is required for this carrier.

Lumen allows 2 forms of routing, based on the type of termination for this record. The two termination types are known as shared and unique. All parameters above this note are common values associated with all Lumen records. All parameters below this note are associated with either shared or unique routing.

*A shared record only sets the **routename** parameter.*

*A unique record uses a combination of all remaining parameters: **priringtonumber**, **priswitch**, **pritrunk**, **pridnis**, **altringtonumber altswitch**, **alttrunk**, **altdnis**, **secaltswitch**, **secalttrunk**, **secaltdnis**.*

If shared and unique routing parameters are both specified in a single API call, 8MS will reject that call with a validation error indicating that conflicting parameters have been specified in a single call.

routename The Route Name for this toll-free number.

This field is associated with the *shared* termination type.

priringtonumber The primary Ring To Number (Translated WTN) which must be a valid POTS number. This can be the same as **billingtelno**.

This field is associated with the *unique* termination type.

altringtonumber An alternate Ring To Number (Translated WTN) which must be a valid POTS number.

This field is associated with the *unique* termination type.

priswitch The primary Switch used for routing calls for this toll-free number.

This field is associated with the *unique* termination type.

pritrunk The Trunk Group associated with the specified **priswitch**.

This field is associated with the *unique* termination type.

pridnis The DNIS digits passed on during the call, associated with the specified **priswitch**. If specified it must be 2-10 digits or the

string *"TFN"*.

This field is associated with the *unique* termination type.

altswitch The alternate Switch used for routing calls for this toll-free number.

This field is associated with the *unique* termination type.

alttrunk The Trunk Group associated with the specified **altswitch**.

This field is associated with the *unique* termination type.

altdnis The DNIS digits passed on during the call, associated with the specified **altswitch**. If specified it must be 2-10 digits or the string *"TFN"*.

This field is associated with the *unique* termination type.

secaltswitch The secondary alternate Switch used for routing calls for this toll-free number.

This field is associated with the *unique* termination type.

secalttrunk The Trunk Group associated with the specified **secaltswitch**.

This field is associated with the *unique* termination type.

secaltdnis The DNIS digits passed on during the call, associated with the specified **secaltswitch**. If specified it must be 2-10 digits or the string *"TFN"*.

This field is associated with the *unique* termination type.

The following parameters are used only when **carrier** is set to *Sinch*. These parameters are all required for this carrier.

* **productAbbr** The abbreviation identifying the product. Valid values are:

- 8XX

customerOrderReference The customer order reference identifier.

routingLabel The routing option to assign the Toll-Free Number. This is optional for *Dedicated* service and must not be set for *Switched* service. Alpha-numeric and dash are valid characters.

dnis The DNIS digits passed on during the call. This is optional for *Dedicated* service and must not be set for *Switched* service. Must be numeric.

billingTn The billing Toll-Free Number. If specified, it must be 10 digits.

termTn The terminating Toll-Free Number. If specified, it must be 10

digits.

This is optional for *Switched* service and must not be set for *Dedicated* service.

***dnis**, **termTn** and **routingLabel** are mutually exclusive. Only one of these parameters may be set.*

The following parameters are used only when **carrier** is set to *Verizon-0222*. These parameters are all required for this carrier.

* **corpid** The Verizon Corp ID in which to activate this number.
* **planid** The Verizon Plan ID to be used for routing this number.

Example

The following request parameters will activate a number in Verizon's CIC 0222 network.

```
carrier~Verizon-0222;dn~8005551212;corpid~ABC123;planid~1
```

Name

CarrierExpressChange Send a change request for a single toll-free number to a single carrier via the Carrier Express system.

Description

CarrierExpressChange sends a change request to a single carrier for a single toll-free number. The following are the allowed fields in *reqparams*:

*	**carrier**	The carrier on which the specified number is to be change. • ATT • Sinch
*	**dn**	The toll-free number to be changed.

The following parameters are used only when **carrier** is set to *ATT*.

*	**action**	The type of service to be changed. Valid values are: • iptf360
*	**profilename**	Name of profile containing a group of shared values used in place of individual values.
	distgroup	Distribution group
	payphoneind	Indicates whether or not payphone calls are blocked. Valid values are upper-case *Y* or *N*.
	resporg	The RespOrg of the toll-free number.
*	**changetype**	The type of change being made. Valid values include any combination of: *R*, *P*, and *D* where • *R* = RespOrg • *P* = Payphone Indicator • *D* = Distribution Group

***changetype** restrictions include:*

- *all values must be upper-case.*
- *values should not be separated by any characters, e.g. "R,D" or "R D" are invalid.*
- *values should match the change fields that are set: **resporg**, **payphoneind**, and **distgroup**.*
- *values should be combined in any order, e.g.*

*when setting **resporg** and **distgroup**, "RD" or "DR" are both valid.*

*	**contactname**	The contact name for this toll-free number. Its maximum size is 30 characters.
*	**contactphone**	The contact phone number for this toll-free number.
	clientorderno	The purchase order number/client order number for this toll-free number. Its maximum size is 21 characters.

The following parameters are used only when **carrier** is set to *Sinch*. These parameters are all required for this carrier.

	customerOrderReference	The alpha-numeric customer order reference identifier.
*	**changetype**	The type of change being made. Valid values are one of: *R*, *D*, *T* or *X* where • *R* = set Routing Label • *D* = set DNIS • *T* = set Term TN • *X* = remove DNIS ***changetype** restrictions include:* • *all values must be upper-case.* • *only one character is allowed to be specified.* • *values should match the change field that is set:* ***R**: **routingLabel** must be set* ***D**: **dnis** must be set* ***T**: **termTn** must be set* ***X**: Indicates dnis will be removed at Sinch. No fields (except **customerOrderReference**) may be set.*
	routingLabel	The routing option to assign the Toll-Free Number. Alpha-numeric and dash are valid characters.
	dnis	The DNIS digits passed on during the call. Must be numeric.
	termTn	The terminating Toll-Free Number. If specified, it must be 10 digits.

Example

The following request parameters will change a number in AT&T's network.

carrier~ATT;dn~8005551212;mcn~F1234567;action~iptf360;contactname~Joe Q. Contact;contactphone~7323020222;

Name

CarrierExpressDisconnect Send a disconnect request for a single toll-free number to a single carrier via the Carrier Express system.

Description

CarrierExpressDisconnect sends an disconnect request to a single carrier for a single toll-free number. The following are the allowed fields in *reqparams*:

*	**carrier**	The carrier on which the specified number is to be disconnected. Valid values are: • ATT • Lumen-5102 • Sinch • Verizon-0222
*	**dn**	The toll-free number to be disconnected.

The following parameters are used only when **carrier** is set to *ATT*.

*	**action**	The type of service to be disconnected. Valid values are: • iptf360
*	**profilename**	Name of profile containing a group of shared values used in place of individual values.
*	**contactname**	The contact name for this toll-free number. Its maximum size is 30 characters.
*	**contactphone**	The contact phone number for this toll-free number.
	clientorderno	The purchase order number/client order number for this toll-free number. Its maximum size is 21 characters.

The following parameters are used only when **carrier** is set to *Sinch*. These parameters are all required for this carrier.

	customerOrderReference	The alpha-numeric customer order reference identifier.

The following parameters are used only when **carrier** is set to *Lumen-5102*. These parameters are all required for this carrier.

*	**customernumber**	The Lumen customer number.
	resellerclientid	The Reseller Client ID.
*	**terminationtype**	The termination type for this toll-free number. Valid values are *0* and *1*. A value of *0* means this is a *Shared* termination while a value of *1* means this is a *Unique* termination.

The following parameters are used only when **carrier** is set to *Verizon-0222*. These parameters are all required for this carrier.

*	**corpid**	The Verizon Corp ID in which to disconnect this number.

Example

The following request parameters will disconnect a number from Verizon's CIC 0222 network.

```
carrier~Verizon-0222;dn~8005551212;corpid~ABC123
```

Name

BatchCarrierExpressActivate Send an activate request for a group of toll-free numbers to a single carrier via the Carrier Express system.

Description

BatchCarrierExpressActivate sends an activate request to a single carrier for a group of toll-free numbers. The following are the allowed fields in *reqparams*:

*	**carrier**	The carrier on which the specified number is to be activated. Valid values are: • ATT • Lumen-5102 • Sinch • Verizon-0222
*	**dnlist**	The list of numbers to be activated.

The following parameters are used only when **carrier** is set to *ATT*.

*	**action**	The type of service to be activated. Valid values are: • iptf360
*	**profilename**	Name of profile containing a group of shared values used in place of individual values.
*	**contactname**	The contact name for this toll-free number. Its maximum size is 30 characters.
*	**contactphone**	The contact phone number for this toll-free number.
	clientorderno	The purchase order number/client order number for this toll-free number. Its maximum size is 21 characters.
*	**payphoneind**	Indicates whether or not payphone calls are blocked. Valid values are upper-case *Y* or *N*.
*	**resporg**	The RespOrg of the toll-free number.

The following parameters are used only when **carrier** is set to *Lumen-5102*.

	customernumber	The Lumen customer number. This parameter is required for this carrier.
	billingaccount	The Lumen billing account. This parameter is required for this carrier.
	billingtelno	The billing telephone number which must be a valid POTS number. This parameter is required for this carrier.

resellerclientid The Reseller Client ID.

servicearea The network area of service. Valid values are:

- US
- XA
- XB
- XC

This parameter is required for this carrier.

Lumen allows 2 forms of routing, based on the type of termination for this record. The two termination types are known as shared and unique. All parameters above this note are common values associated with all Lumen records. All parameters below this note are associated with either shared or unique routing.

*A shared record only sets the **routename** parameter.*

*A unique record uses a combination of all remaining parameters: **priringtonumber**, **priswitch**, **pritrunk**, **pridnis**, **altringtonumber altswitch**, **alttrunk**, **altdnis**, **secaltswitch**, **secalttrunk**, **secaltdnis**.*

If shared and unique routing parameters are both specified in a single API call, 8MS will reject that call with a validation error indicating that conflicting parameters have been specified in a single call.

routename The Route Name for this toll-free number.

This field is associated with the *shared* termination type.

priringtonumber The primary Ring To Number (Translated WTN) which must be a valid POTS number. This can be the same as **billingtelno**.

This field is associated with the *unique* termination type.

altringtonumber An alternate Ring To Number (Translated WTN) which must be a valid POTS number.

This field is associated with the *unique* termination type.

priswitch The primary Switch used for routing calls for this toll-free number.

This field is associated with the *unique* termination type.

pritrunk The Trunk Group associated with the specified **priswitch**.

This field is associated with the *unique* termination type.

pridnis The DNIS digits passed on during the call, associated with the specified **priswitch**. If specified it must be 2-10 digits or the

		string *"TFN"*. This field is associated with the *unique* termination type.
	altswitch	The alternate Switch used for routing calls for this toll-free number. This field is associated with the *unique* termination type.
	alttrunk	The Trunk Group associated with the specified **altswitch**. This field is associated with the *unique* termination type.
	altdnis	The DNIS digits passed on during the call, associated with the specified **altswitch**. If specified it must be 2-10 digits or the string *"TFN"*. This field is associated with the *unique* termination type.
	secaltswitch	The secondary alternate Switch used for routing calls for this toll-free number. This field is associated with the *unique* termination type.
	secalttrunk	The Trunk Group associated with the specified **secaltswitch**. This field is associated with the *unique* termination type.
	secaltdnis	The DNIS digits passed on during the call, associated with the specified **secaltswitch**. If specified it must be 2-10 digits or the string *"TFN"*. This field is associated with the *unique* termination type.
*	**productAbbr**	The abbreviation identifying the product. Valid values are: • 8XX
	customerOrderReference	The customer order reference identifier.
	routingLabel	The routing option to assign the Toll-Free Number. This is optional for *Dedicated* service and must not be set for *Switched* service. Alpha-numeric and dash are valid characters.
	dnis	The DNIS digits passed on during the call. This is optional for *Dedicated* service and must not be set for *Switched* service. Must be numeric.
	billingTn	The billing Toll-Free Number. If specified, it must be 10 digits.
	termTn	The terminating Toll-Free Number. If specified, it must be 10 digits. This is optional for *Switched* service and must not be set for *Dedicated* service.

***dnis**, **termTn** and **routingLabel** are mutually exclusive. Only one of these parameters may be set.*

The following parameters are used only when **carrier** is set to *Verizon-0222*. These parameters are all required for this carrier.

* **corpid** The Verizon Corp ID in which to activate this number.
* **planid** The Verizon Plan ID to be used for routing this number.

Example

The following request parameters will activate a group of numbers in Verizon's CIC 0222 network.

```
carrier~Verizon-
0222;dnlist~8005551212,8885551212;corpid~ABC123;planid~1
```

Name

BatchCarrierExpressChange Send a change request for a group of toll-free numbers to a single carrier via the Carrier Express system.

Description

BatchCarrierExpressChange sends a change request to a single carrier for a group of toll-free numbers. The following are the allowed fields in *reqparams*:

* **carrier** — The carrier on which the specified number is to be change.
  - ATT
  - Sinch
* **dnlist** — The list of numbers to be changed.

The following parameters are used only when **carrier** is set to *ATT*.

* **action** — The type of service to be changed. Valid values are:
  - iptf360
* **profilename** — Name of profile containing a group of shared values used in place of individual values.

  distgroup — Distribution group

  payphoneind — Indicates whether or not payphone calls are blocked.

  Valid values are upper-case *Y* or *N*.

  resporg — The RespOrg of the toll-free number.
* **changetype** — The type of change being made.

  Valid values include any combination of: *R*, *P*, and *D*

  where
  - *R* = RespOrg
  - *P* = Payphone Indicator
  - *D* = Distribution Group

  ***changetype** restrictions include:*
  - *all values must be upper-case.*
  - *values should not be separated by any characters, e.g. "R,D" or "R D" are invalid.*
  - *values should match the change fields that are set: **resporg**, **payphoneind**, and **distgroup**.*
  - *values may be combined in any order, e.g. when setting **resporg** and **distgroup**, "RD" or "DR" are both valid.*
* **contactname** — The contact name for this toll-free number. Its maximum size is 30 characters.

*	**contactphone**	The contact phone number for this toll-free number.
	clientorderno	The purchase order number/client order number for this toll-free number. Its maximum size is 21 characters.

The following parameters are used only when **carrier** is set to *Sinch*. These parameters are all required for this carrier.

	customerOrderReference	The alpha-numeric customer order reference identifier.
*	**changetype**	The type of change being made. Valid values are one of: *R*, *D*, *T* or *X* where • *R* = set Routing Label • *D* = set DNIS • *T* = set Term TN • *X* = remove DNIS ***changetype** restrictions include:* • *all values must be upper-case.* • *only one character is allowed to be specified.* • *values should match the change field that is set:* ***R**: **routingLabel** must be set* ***D**: **dnis** must be set* ***T**: **termTn** must be set* ***X**: Indicates dnis will be removed at Sinch. No fields (except **customerOrderReference**) may be set.*
	routingLabel	The routing option to assign the Toll-Free Number. Alpha-numeric and dash are valid characters.
	dnis	The DNIS digits passed on during the call. Must be numeric.
	termTn	The terminating Toll-Free Number. If specified, it must be 10 digits.

Example

The following request parameters will change a list of numbers in AT&T's network.

```
carrier~ATT;dnlist~8005551212,8885551212;mcn~F1234567;action~iptf360;contactname~Joe Q. Contact;contactphone~7323020222;
```

Name

BatchCarrierExpressDisconnect Send a disconnect request for a group of toll-free numbers to a single carrier via the Carrier Express system.

Description

BatchCarrierExpressDisconnect sends an disconnect request to a single carrier for a group of toll-free numbers. The following are the allowed fields in *reqparams*:

*	**carrier**	The carrier on which the specified number is to be disconnected. Valid values are: • ATT • Lumen-5102 • Sinch • Verizon-0222
*	**dnlist**	The list of numbers to be activated.

The following parameters are used only when **carrier** is set to *ATT*.

*	**action**	The type of service to be disconnected. Valid values are: • iptf360
*	**profilename**	Name of profile containing a group of shared values used in place of individual values.
*	**contactname**	The contact name for this toll-free number. Its maximum size is 30 characters.
*	**contactphone**	The contact phone number for this toll-free number.
	clientorderno	The purchase order number/client order number for this toll-free number. Its maximum size is 21 characters.

The following parameters are used only when **carrier** is set to *Sinch*. These parameters are all required for this carrier.

	customerOrderReference	The alpha-numeric customer order reference identifier.

The following parameters are used only when **carrier** is set to *Lumen-5102*. These parameters are all required for this carrier.

*	**customernumber**	The Lumen customer number.
	resellerclientid	The Reseller Client ID.
*	**terminationtype**	The termination type for this toll-free number. Valid values are *0* and *1*. A value of *0* means this is a *Shared* termination while a value of *1* means this is a *Unique* termination.

The following parameters are used only when **carrier** is set to *Verizon-0222*. These parameters are all required for this carrier.

*	**corpid**	The Verizon Corp ID in which to disconnect this set of numbers.

Example

The following request parameters will disconnect a group of numbers from Verizon's CIC 0222 network.

```
carrier~Verizon-0222;dnlist~8005551212,8885551212;corpid~ABC123
```

ROC (Resp Org Change) Requests

A Resp Org change allows movement of toll-free numbers from one Resp Org to another. The Resp Org Change feature, generally known as ROC, is an SMS/800 feature that provides the ability to manage requests and approvals of Resp Org changes. The following ROC requests are supported by the 8MS API:

ROCDownloadDocument

ROCCreateLOA

ROCManageRequest

ROCNumberRequest

ROCRetrieveRequests

ROCRetrieveRequestDetail

ROCUploadDocument

Name

ROCDownloadDocument Download a document associated with a request.

Description

ROCDownloadDocument downloads a single LOA or other document associated with a request. The following are the allowed fields in *reqparams*:

rocusername	The username registered at SMS/800 for ROC requests.
rocpasswd	The password associated with the username registered at SMS/800 for ROC requests.
entity	The SMS/800 entity for ROC requests.
loaid	The ID identifying this LOA document.
documentid	The ID identifying this non-LOA document.

*Either the single parameter **entity** or the pair of parameters **rocusername** and **rocpasswd** must be specified.*

*Either **loaid** or **documentid** must be specified.*

ROC Messages

ROCDownloadDocument sends a **GetLOAFile** message to retrieve an LOA or it sends a **GetDocument** message to retrieve any other document.

Results

A RocDocumentDownload response is an XML document embedded within the normal API response. If the document is an LOA document then a GetLOAFileResponse XML document will be embedded; if the document is a non-LOA document then a GetDocumentResponse document will be embedded. Note that the response XML for these two types are different from one another.

*The **ErrorList** section will only be included if an error has occurred.*

A typical XML response for an LOA document will have the following fields.

```
<GetLOAFileResponse>
<StatusCode>1</StatusCode>
<LOAFile>
<FileName>LOA 01.pdf</FileName>
<EncodedContent>content omitted</EncodedContent>
<MimeType>text/pdf</MimeType>
</LOAFile>
<ErrorList>
<Error>
<Code>ErrorCode1</Code>
<Description>ErrorDescription1</Description>
<AdditionalInfo>AdditionalInfo1</AdditionalInfo>
<Error>
```

```
<ErrorList>
</GetLOAFileResponse>
```

A typical XML response for a non-LOA document will have the following fields.

```
<GetDocumentResponse>
<StatusCode>1</StatusCode>
<DocumentFile>
<DocumentID>41</DocumentID>
<FileTitle>Sample.pdf</FileTitle>
<EncodedContent>content omitted</EncodedContent>
<FileNotes>Additional information as requested.<FileNotes>
</DocumentFile>
<ErrorList>
<Error>
<Code>ErrorCode1>/Code>
<Description>ErrorDescription1</Description>
<AdditionalInfo>AdditionalInfo</AdditionalInfo>
<Error>
<ErrorList>
</GetDocumentResponse>
```

Most data validation is left to Somos to perform. In some cases, 8MS may provide some up-front validation. In that event, the following XML will be returned.

```
<err>
<code>E0000000</code>
<context>param-in-error</context>
<description>text describing error.</description>
<err>
```

Possible errors returned for this API call include

Code	Context	Description
E0000091	loaid	Parameter 'loaid' is invalid
E0000092	documentid	Parameter 'documentid' is invalid
E0000093	entity/rocusername	A valid 'entity' or 'rocusername' must be specified.

Example

The following request parameters will retrieve a non-LOA document with a docid of 41:

```
entity~YH;documentid~41
```

Name

ROCCreateLOA Create a Standard LOA Cover document to attach to a request.

Description

ROCCreateLOA allows the requesting Resp Orgs to create a Standard LOA Cover document, which can be attached to a ROC request. The following are the allowed fields in *reqparams*:

	rocusername	The username registered at SMS/800 for ROC requests.
	rocpasswd	The password associated with the username registered at SMS/800 for ROC requests.
	entity	The SMS/800 entity for ROC requests.
*	**dnlist**	A comma-separated list of numbers to be included in this LOA.
*	**contactcustomer**	The name of the company submitting this LOA.
*	**contactname**	The name of the person submitting this LOA.
*	**contacttitle**	The title of the person submitting this LOA.
*	**address1**	The first line of the contact's address.
	address2	The second line of the contact's address.
*	**city**	The contact's city.
*	**state**	The contact's state.
*	**zipcode**	The contact's zip code.
*	**contactphone**	The contact's phone number.
	contactextension	The contact's phone extension.
	contactfax	The contact's fax number.

*Either the single parameter **entity** or the pair of parameters **rocusername** and **rocpasswd** must be specified.*

ROC Messages

ROCCreateLOA sends a **GenerateStandardLOA** message.

Results

A RocRetrieveRequests response is an XML document embedded within the normal API response. A typical XML response will have the following fields.

*The **ErrorList** section will only be included if an error has occurred.*

```
<GenerateStandardLOAResponse>
```

```
<StatusCode>1</StatusCode>
<StandardLOAFile>
<FileName>LOABarcode.pdf</FileName>
<EncodedContent>Dkkjdkafwew9ekjsakdKJDKFJ==</EncodedContent>
<MimeType>application/pdf</MimeType>
</StandardLOAFile>
</GenerateStandardLOAResponse>
```

Most data validation is left to Somos to perform. In some cases, 8MS may provide some up-front validation. In that event, the following XML will be returned.

```
<err>
<code>E0000000</code>
<context>param-in-error</context>
<description>text describing error.</description>
<err>
```

Possible errors returned for this API call include

Code	Context	Description
E0000093	entity/rocusername	A valid 'entity' or 'rocusername' must be specified.

Example

The following request parameters will generate an LOA for CSF.

```
entity~AR;dnlist~8005551212;contactcustomer~CSF;contactname~John
Smith; contacttitle~VP of Sales;address1~285 Davidson
Ave;address2~Suite 103;
city~Somerset;state~NJ;zipcode~08873;contactphone~7323020222;
contactextension~1234;contactfax~73233020799
```

Name

ROCManageRequest Approve or reject numbers in a request, remove numbers from a Pending request or completely cancel a request.

Description

ROCManageRequest allows the controlling Resp Org to approve or reject numbers in a request, allows the submitting Resp Org to remove one or more numbers from a Pending request, or allows an entire request to be canceled. The following are the allowed fields in *reqparams*:

rocusername The username registered at SMS/800 for ROC requests.

rocpasswd The password associated with the username registered at SMS/800 for ROC requests.

entity The SMS/800 entity for ROC requests.

dnlist A comma-separated list of numbers to be managed in the specified request.

> ***dnlist** should not be specified when **action** is set to 'cancel', as this is the only action that affects the entire request. For all other values of **action**, **dnlist** is required.*

* **action** The action to perform on the set of toll-free numbers in this request. Valid values are:

Value	Description	Allowed By
approved	Immediately approve the request for the specified numbers	Controlling Resp Org
pendingApprove	Approve the request for the specified numbers when the due date of the request is reached	Controlling Resp Org
cancelPendingApprove	Cancel a pending approval of a request for the specified numbers. This can only be done if a pendingApprove was previously done.	Controlling Resp Org
rejected	Reject the request for the specified numbers	Controlling Resp Org
remove	Remove the specified numbers from this request	Submitting Resp Org
cancel	Cancel an entire request	Submitting Resp Org

* **txnid** The Transaction ID returned from a **ROCRetrieveRequests API call.**

rejectcode A comma-separated list of codes indicating why these numbers were rejected.

*This parameter is required if **action** is reject.*

Valid values are the codes shown in the table below.

Code	Description
01	Customer name mismatch/missing
02	Address mismatch/missing (verification done if address is different but all other information is the same)
03	Contact/Customer signature missing
04	Toll-Free Shared or Bundled
05	Customer signature date missing/or expired (must be less than 30 days)
06	Sent to wrong Resp Org
07	Toll-Free Number not listed on request
08	All data mismatch
09	LOA missing or linking Reseller/Subscriber LOA missing
11	Illegible LOA
12	More recent LOA (provide copy of LOA to Resp Org)
15	Unauthorized contact/Customer signature
18	Resp Org is no longer in control of the Toll-Free Number

The above codes and descriptions are taken from the SMS/800 document Centralized Resp Org Change (ROC) Management System Web Services Interface Specification, Issue 1, Version 2.14, dated March 10, 2016, Appendix C - Table 14.

rejectnote A note indicating why this request is being rejected.

*Either the single parameter **entity** or the pair of parameters **rocusername** and **rocpasswd** must be specified.*

ROC Messages

ROCManageRequest sends a **ProcessRespOrgChangeRequest** message to approve or reject numbers in a request, a **RemoveDialNumbers** to remove numbers from a request or a **CancelROCRequest** message to cancel a complete request.

Results

A RocManageRequest response is an XML document embedded within the normal API response. If the request was to approve or reject a request then a ProcessRespOrgChangeRequest XML document will be embedded; if the document was to remove numbers from a request then a RemoveDialNumbers document will be embedded. Note that the response XML for these two types differs only in the top-level tag, as shown below.

*The **ErrorList** section will only be included if an error has occurred.*

A typical XML response for approving or rejecting numbers in a request will have the following fields.

```
<ProcessRespOrgChangeRequestResponse>
<StatusCode>1</StatusCode>
<ErrorList>
<Error>
<Code>ErrorCode1>/Code>
<Description>ErrorDescription1</Description>
<AdditionalInfo>AdditionalInfo1</AdditionalInfo>
<Error>
<ErrorList>
</ProcessRespOrgChangeRequestResponse>
```

A typical XML response for removing numbers from a request will have the following fields.

```
<RemoveDialNumbersResponse>
<StatusCode>1</StatusCode>
<ErrorList>
<Error>
<Code>ErrorCode1</Code>
<Description>ErrorDescription1</Description>
<AdditionalInfo>AdditionalInfo1</AdditionalInfo>
<Error>
<ErrorList>
</RemoveDialNumbersResponse>
```

A typical XML response for canceling a request will have the following fields.

```
<CancelRespOrgChangeRequestResponse>
<StatusCode>1</StatusCode>
<ErrorList>
<Error>
<Code>ErrorCode1<Code>
<Description>ErrorDescription1</Description>
<AdditionalInfo>AdditionalInfo1</AdditionalInfo>
</Error>
</ErrorList>
</CancelRespOrgChangeRequestResponse>
```

Most data validation is left to Somos to perform. In some cases, 8MS may provide some up-front validation. In that event, the following XML will be returned.

```
<err>
<code>E0000000</code>
<context>param-in-error</context>
<description>text describing error.</description>
<err>
```

Possible errors returned for this API call include

Code	Context	Description
E0000089	action	Parameter 'action' was not specified
E0000090	action	Parameter 'action' is invalid. Must be one of approved, pendingApprove, cancelPendingApprove, rejected, remove or cancel

E0000093	entity/rocusername	A valid 'entity' or 'rocusername' must be specified.
E0000095	rejectcode	rejectcode(s) are required when action is rejected.

Example

The following request parameters will remove a number from a ROC request:

```
entity~YH;action~remove;dnlist~8005551212; txnid~~b5fa1b8a-a54c-4c2a-8014-3e5f2dd30d6e;
```

Name

ROCNumberRequest Generate a new authorization request.

Description

ROCNumberRequest allows a requesting Resp Org to submit an authorization request to a controllling Resp Org, requesting that one or more numbers be ported to the requesting Resp Org. The following are the allowed fields in *reqparams*:

rocusername The username registered at SMS/800 for ROC requests.

rocpasswd The password associated with the username registered at SMS/800 for ROC requests.

entity The SMS/800 entity for ROC requests.

dnlist A comma-separated list of numbers to be Resp Org changed.

resporg The Resp Org to which this set of numbers should be changed.

notes Any additional notes to assist in processing this request.

loafilename The name of the LOA document to be uploaded.

> *The file name can contain A-Z, a-z, 0-9, underscore (_), hyphen (-) only. The maximum length of the file name is 100 characters.*

loaencodedcontent The LOA document data, encoded in Base64.

> *This parameter is required if **loafilename** is specified.*

loamimetype The mime type of the file data specified in **loaencodeddata**.

> *This value defaults to application/pdf if **loafilename** ends in .pdf or image/tiff if **loafilename** ends in .tif or .tiff.*

requestdate The date on which this request should be sent to the Controlling Resp Org. This allows the Submitting Resp Org to create a request but delay sending it to the Controlling Resp Org. The format of this value is:

```
mm/dd/yy
```

An example of a valid date/time would be 02/03/23.

> *The vast majority of the time, users do not set a requestdate. Somos then immediately sends the ROC request to the controlling Resp Org.*
>
> *Providing a requestdate means Somos will create the ROC request but will not send it to the controlling Resp Org until*

the day of the requestdate. This means the controlling Resp Org won't be able to see the request, or respond to it, until the date you specified.

We strongly recommend that you do not set the requestdate parameter unless you fully understand what this means.

*Either the single parameter **entity** or the pair of parameters **rocusername** and **rocpasswd** must be specified.*

ROC Messages

ROCNumberRequest sends a **SubmitRespOrgChange** message.

Results

A RocNumberRequest response is an XML document embedded within the normal API response. A typical XML response will have the following fields.

*The **ErrorList** section will only be included if an error has occurred.*

```
<SubmitRespOrgChangeResponse>
<StatusCode>1</StatusCode>
<TxnID>3F5E3197-9A24-446F-884D-10F95D83F8CB</TxnID>
<LOAId>100</LOAID>
<LOAFileName>LoaFile.pdf</LOAFileName>
<ErrorList>
<Error>
<Code>ErrorCode1>/Code>
<Description>ErrorDescription1</Description>
<AdditionalInfo>AdditionalInfo1</AdditionalInfo>
<Error>
<ErrorList>
</SubmitRespOrgChangeResponse>
```

Most data validation is left to Somos to perform. In some cases, 8MS may provide some up-front validation. In that event, the following XML will be returned.

```
<err>
<code>E0000000</code>
<context>param-in-error</context>
<description>text describing error.</description>
<err>
```

Possible errors returned for this API call include

Code	Context	Description
E0000088	requestdate	Parameter 'requestdate' is invalid
E0000093	entity/rocusername	A valid 'entity' or 'rocusername' must be specified.

Example

The following request parameters will submit a ROC request for a single number:

```
entity~AR;dnlist~8448153985;resporg~art01;loafilename~8448153985-bill.pdf;loaencodedcontent~content-omitted
```

*The pdf encoded content has been omitted in the above example because this value is typically very long and unreadable since it is encoded. For the actual API call, the loaencodedcontent should contain the actual pdf encoded value; it should **not** include the string "content-omitted".*

Name

ROCRetrieveRequests Search all requests for requests matching a set of search criteria.

Description

ROCRetrieveRequests allows the controlling and requesting Resp Orgs to search all requests for requests matching a set of search criteria. Controlling Resp Orgs may only receive a list of numbers that have been requested of them; numbers not belonging to them may not be queried. The following are the allowed fields in *reqparams*:

rocusername The username registered at SMS/800 for ROC requests.

rocpasswd The password associated with the username registered at SMS/800 for ROC requests.

entity The SMS/800 entity for ROC requests.

resporg To limit the returned results to requests involving this Resp Org.

dn To limit the returned results to requests involving this single number.

* **type** To limit the returned results to requests with the specified request type. Valid values are:

- *incoming*
- *outgoing*

status To limit the returned results to requests with the specified request status. Valid values are:

- *all*
- *pending*
- *processing*
- *ported*
- *declined*
- *expired*
- *escalated*
- *invalid*
- *failed*
- *overdue*
- *duedateapproval*

***status** defaults to all if no value is provided.*

progress To limit the returned results to open or closed requests. Valid values are:

- *all*
- *open*
- *closed*

***progress** defaults to all if no value is provided.*

rejectcode A single code as defined in **ROCManageRequest**.

startdatetime To limit the returned results to requests with the submitted after and including this date/time. The format of this value is:

mm/dd/yy hh:mmX/Y

where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/16 10:15A/C.

enddatetime To limit the returned results to requests with the submitted up to and including this date/time. The format of this value is:

mm/dd/yy hh:mmX/Y

where X is the meridiem (A or P), and Y is the SMS/800 timezone indicator. An example of a valid date/time would be: 07/01/16 10:15A/C.

bytxnid To eliminate all toll-free number detail, resulting in a much smaller XML, returned faster, and faster to process, set this to the value *1*. To get the number detail, either set this to the value *0* or don't specify this parameter.

*Either the single parameter **entity** or the pair of parameters **rocusername** and **rocpasswd** must be specified.*

ROC Messages

ROCRetrieveRequests sends a **SearchRespOrgChangeRequests** message if **bytxnid** is *0* or not specified.
It sends a **SearchRespOrgChangeRequestsByTxnID** message if **bytxnid** is *1*.

Results

A RocRetrieveRequests response is an XML document embedded within the normal API response.

*The **ErrorList** section will only be included if an error has occurred.*

A typical XML response, when *bytxnid* is either not specified or set to *0*, will have the following fields.

```
<SearchRespOrgChangeRequestsResponse>
<StatusCode>1</StatusCode>
<DialNumberResultList>
<DialNumber>
<Dial>8443202161</Dial>
<TxnID>7671548c-9a43-4fb6-8b82-2cd35fed8f41</TxnID>
<SubmittedDateTime>2016-05-04T12:21:28</SubmittedDateTime>
<ProcessedDateTime>2016-05-05T06:14:09.253</ProcessedDateTime>
<DueDateTime>2016-05-06T12:21:00</DueDateTime>
<SubmittingRespOrg>art01</SubmittingRespOrg>
<ControllingRespOrg>ART04</ControllingRespOrg>
<Status>3</Status>
```

```
<LOAID>143</LOAID>
<LOAFileName>Test 143.pdf</LOAFileName>
<DocumentList>
<Document>
<DocumentID>144</DocumentID>
<FileTitle>Test 144.pdf</FileTitle>
<FileNotes>Test Comments</FileNotes>
</Document>
</DocumentList>
<RejectReasonList>
<RejectReason>
<ReasonCode>01</ReasonCode>
<ReasonDescription>Customer name mismatch/missing</ReasonDescription>
</RejectReason>
<RejectReason>
<ReasonCode>02</ReasonCode>
<ReasonDescription>Address mismatch/missing (verification done if address is
different but all other information is the same)</ReasonDescription>
</RejectReason>
</RejectReasonList>
<RejectNote>Test Reject Note</RejectNote>
</DialNumber>
<DialNumber>
<Dial>8443202162</Dial>
<TxnID>16ac084c-be8e-4b6f-b9d5-8ca62b70ab74</TxnID>
<SubmittedDateTime>2016-05-04T12:24:10</SubmittedDateTime>
<ProcessedDateTime>2016-05-05T06:14:09.257</ProcessedDateTime>
<DueDateTime>2016-05-06T12:24:00</DueDateTime>
<SubmittingRespOrg>art01</SubmittingRespOrg>
<ControllingRespOrg>ART04</ControllingRespOrg>
<Status>3</Status>
<LOAID>143</LOAID>
<LOAFileName>Test 143.pdf</LOAFileName>
<DocumentList>
<Document>
<DocumentID>144 <FileTitle>Test 144.pdf</FileTitle>
<FileNotes>Test Comments</FileNotes>
</Document>
</DialNumber>
</DialNumberResultList>
<ErrorList>
<Error>
<Code>ErrorCode1>/Code>
<Description>ErrorDescription1</Description>
<AdditionalInfo>AdditionalInfo1</AdditionalInfo>
<Error>
<ErrorList>
</SearchRespOrgChangeRequestsResponse>
```

A typical XML response, when *bytxnid* is set to *1*, will have the following fields.

```
<SearchRespOrgChangeRequestsByTxnIDResponse>
<StatusCode>1</StatusCode>
<TransactionResultList>
<Transaction>
<TxnID>8C1D4C99-8D45-49C2-899E-32B3BA3212B0</TxnID>
<TFNCount>25</TFNCount>
<ExpediteROC>Yes</ExpediteROC>
<SubmittedDateTime>2018-07-01T00:00:00</SubmittedDateTime>
<DueDateTime>2018-07-03T00:00:00</DueDateTime>
<IsRequestCheckedOut>Yes</IsRequestCheckedOut>
<RequestCheckedOutBy>xxxxxxx</RequestCheckedOutBy>
<SubmittingRespOrg>ZATXT</SubmittingRespOrg>
```

```
<LOAID>43</LOAID>
<LOAFileName>Test 43.pdf</LOAFileName>
<DocumentList>
<Document>
<DocumentID>1122</DocumentID>
<FileTitle>TEST 1122.PDF</FileTitle>
<FileNotes />
</Document>
<Document>
<DocumentID>3344</DocumentID>
<FileTitle>TEST 3344.PDF</FileTitle>
<FileNotes />
</Document>
</DocumentList>
<ControllingRespOrgList>
<ControllingRespOrg>
<RespOrg>ABC01</RespOrg>
<Count>7</Count>
</ControllingRespOrg>
<ControllingRespOrg>
<RespOrg>TST99</RespOrg>
<Count>15</Count>
</ControllingRespOrg>
</ControllingRespOrgList>
<UnavailRespOrgTFNCount>3</UnavailRespOrgTFNCount>
</Transaction>
</TransactionResultList>
<ErrorList>ErrorList>
</SearchRespOrgChangeRequestsByTxnIDResponse>
```

Most data validation is left to Somos to perform. In some cases, 8MS may provide some up-front validation. In that event, the following XML will be returned.

```
<err>
<code>E0000000</code>
<context>param-in-error</context>
<description>text describing error.</description>
<err>
```

Possible errors returned for this API call include

Code	Context	Description
E0000093	entity/rocusername	A valid 'entity' or 'rocusername' must be specified.
E0000094	type	Parameter 'type' was not specified.

Example

The following request parameters will query all outgoing ROC requests for entity AR, starting at midnight on May 1, 2016.

```
entity~AR;type~outgoing;startdatetime~05/01/16 12:00A
```

Name

ROCRetrieveRequestDetail Retrieve the details of a specific request.

Description

ROCRetrieveRequestDetail allows the controlling and requesting Resp Orgs to retrieve the details of a specific request. Controlling Resp Orgs may only receive a list of numbers that have been requested of them; numbers not belonging to them may not be queried. The following are the allowed fields in *reqparams*:

	rocusername	The username registered at SMS/800 for ROC requests.
	rocpasswd	The password associated with the username registered at SMS/800 for ROC requests.
	entity	The SMS/800 entity for ROC requests.
*	**txnid**	The Transaction ID returned from a **ROCRetrieveRequests API call.**

*Either the single parameter **entity** or the pair of parameters **rocusername** and **rocpasswd** must be specified.*

ROC Messages

ROCRetrieveRequestDetail sends a **GetRespOrgChangeRequest** message.

Results

A RocRetrieveRequestDetail response is an XML document embedded within the normal API response. A typical XML response will have the following fields.

*The **ErrorList** section will only be included if an error has occurred.*

```
<GetRespOrgChangeRequestResponse>
<StatusCode>1</StatusCode>
<SubmittingRespOrg>art01</SubmittingRespOrg>
<SubmissionDateTime>2016-05-05T07:36:42</SubmissionDateTime>
<DueDateTime>2016-05-09T07:36:00</DueDateTime>
<LOAID>329</LOAID>
<LOAFileName>LOA File 1.pdf</LOAFileName>
<DocumentList>
<Document>
<DocumentID>330</DocumentID>
<FileTitle>Doc2.pdf</FileTitle>
<FileNotes>Remarks</FileNotes>
</Document>
<Document>
<DocumentID>331</DocumentID>
<FileTitle>Doc4.pdf</FileTitle>
<FileNotes>Remarks</FileNotes>
</Document>
</DocumentList>
<Notes></Notes>
<SubmitterName>Steve-AR Levinn</SubmitterName>
<SubmitterPhone>732-302-0222</SubmitterPhone>
```

```
<SubmitterEmail>roc@iconectiv.com</SubmitterEmail>
<SubmitterCompanyName>csf Corporation</SubmitterCompanyName>
<NewRespOrgID>art01</NewRespOrgID>
<DialNumberList>
<DialNumber>
<Dial>8448153984</Dial>
<ProcessedDateTime>2016-05-06T05:58:42.303</ProcessedDateTime>
<Status>3</Status>
<StatusDescription>Declined</StatusDescription>
<RejectReasonList>
<RejectReason>
<ReasonCode>18</ReasonCode>
<ReasonDescription>Resp Org is no longer in control of the Toll Free
Number</ReasonDescription>
</RejectReason>
</RejectReasonList>
<ControllingRespOrgName>ART04</ControllingRespOrgName>
<RejectNote>Test Reject Notes</RejectNote>
</DialNumber>
</DialNumberList>
<ErrorList>
<Error>
<Code>ErrorCode1>/Code>
<Description>ErrorDescription1</Description>
<AdditionalInfo>AdditionalInfo1</AdditionalInfo>
<Error>
<ErrorList>
</GetRespOrgChangeRequestResponse>
```

Most data validation is left to Somos to perform. In some cases, 8MS may provide some up-front validation. In that event, the following XML will be returned.

```
<err>
<code>E0000000</code>
<context>param-in-error</context>
<description>text describing error.</description>
<err>
```

Possible errors returned for this API call include

Code	Context	Description
E0000093	entity/rocusername	A valid 'entity' or 'rocusername' must be specified.

Example

The following request parameters will retrieve detail for a specific ROC request:

```
entity~AR;txnid~a1dd40c5-52aa-4332-83b5-131ada513840;
```

Name

ROCUploadDocument Upload a LOA or other document necessary to support a request.

Description

ROCUploadDocument uploads a single document associated with a request. The following are the allowed fields in *reqparams*:

	rocusername	The username registered at SMS/800 for ROC requests.
	rocpasswd	The password associated with the username registered at SMS/800 for ROC requests.
	entity	The SMS/800 entity for ROC requests.
*	**txnid**	The Transaction ID returned from a **ROCRetrieveRequests API call.**
*	**docfilename**	The name of the document to be uploaded
*	**docencodedcontent**	The document data, encoded in Base64.
*	**doccomments**	Any comments to be associated with this document.

*Either the single parameter **entity** or the pair of parameters **rocusername** and **rocpasswd** must be specified.*

ROC Messages

ROCUploadDocument sends an **AddDocument** message.

Results

A ROCUploadDocument response is an XML document embedded within the normal API response. A typical XML response will have the following fields.

*The **ErrorList** section will only be included if an error has occurred.*

```
<AddDocumentResponse>
<StatusCode>1</StatusCode>
<DocumentID>345</DocumentID>
</AddDocumentResponse>
<ErrorList>
<Error>
<Code>ErrorCode1>/Code>
<Description>ErrorDescription1</Description>
<AdditionalInfo>AdditionalInfo</AdditionalInfo>
<Error>
<ErrorList>
```

Most data validation is left to Somos to perform. In some cases, 8MS may provide some up-front validation. In that event, the following XML will be returned.

```
<err>
<code>E0000000</code>
<context>param-in-error</context>
<description>text describing error.</description>
<err>
```

Possible errors returned for this API call include

Code	Context	Description
E0000093	entity/rocusername	A valid 'entity' or 'rocusername' must be specified.

Example

The following request parameters will add a new document to the associated ROC Request:

```
entity~AR;txnid~7a4eb197-d293-4569-886c-
6bf0a6745517;docfilename~newbill.tif; docencodedcontent~content-omitted
```