Gemini and NTS Exit Reform

API Usage Guidelines

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Acronym	Description
API	Application Programming Interface
DOM	Document Object Model
BA	Business Associate
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IX	Information Exchange
LSO	Local Security Officer
NG	National Grid
NTS	National Transmission System
SAX	Simple API for XML
URI	Uniform Resource Identifier
XML	Extensible Mark-up Language
XSD	XML Schema Definition

1. Introduction

The Gemini and National Transmission System (NTS) Exit Reform systems provide an alternative to screen access via Application Programming Interfaces (APIs), which can be used by National Grid Business Associates (BAs) to access specific Gemini and NTS Exit Reform functions. In order to access National Grid APIs, BAs must develop their own API clients.

APIs are split into two basic categories:

- Functional APIs: These implement the Gemini and NTS Exit Reform business processes that are accessible via APIs.
- Session Management APIs – that is login, logout and change password.

This document provides:

- Guidelines for developing API clients (see Section 2)
- Guidelines on how to use APIs
- Specifications for Session Management APIs

This document does not provide:

- Specifications for the Functional APIs. These are captured in separate documents – one for each API.
- Detailed explanations of the core technologies used to interact with the APIs (example HTTP, XML) – these are fully documented elsewhere (For example: publications of the World Wide Web Consortium). We will only explain aspects of these technologies that are especially pertinent to API client interaction with Gemini and NTS Exit Reform APIs.

Note: The interaction between an API client and an API does not use Citrix technology. This is reserved for screen-based communication with Gemini / NTS Exit Reform.

1.1. API Technology Overview

The following figure illustrates the API client communication with Gemini/NTS Exit Reform.

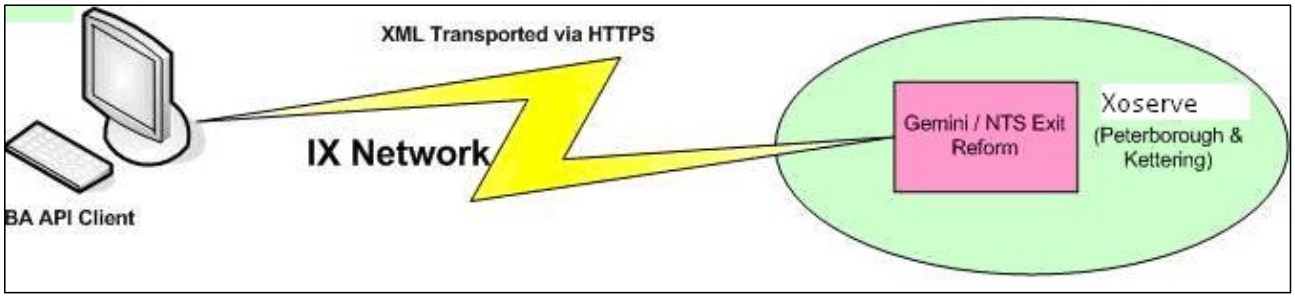


Figure 1: API Client Communication with Gemini/NTS Exit Reform

API clients specify a URL to access an API. These URLs are different from those of Gemini and NTS Exit Reform screens. API clients must issue a Hypertext Transfer Protocol Secure (HTTPS) request that contains API input parameters in Extensible Mark-up Language (XML) format. After processing the request, Gemini / NTS Exit Reform will return an HTTPS response with the output data, also in XML format, in the body.

Each API interaction is therefore a request/response pair. Communication between BAs and Gemini / NTS Exit Reform for API access is via the IX network.

The core technologies used to interact with Gemini / NTS Exit Reform APIs are Hypertext Transfer Protocol (HTTP[1]) and XML. HTTP is used as the communications protocol. XML is used to represent the data that must pass between the API client and the Gemini / NTS Exit Reform APIs to invoke the relevant business functionality.

Note 1: where the HTTP interaction between API client and Gemini / NTS Exit Reform APIs is described in this guide, this is in fact the standard interaction between a web client (normally a web browser) and a web server, supported by the HTTP protocol. In this respect, API client interaction with Gemini / NTS Exit Reform APIs differs only in that it is a two-way, XML-based message exchange alternative to the serving of HTML to a web browser.

Note 2: No XML is exchanged with the Session Management APIs, since they do not implement business functionality.

[1] Or, more precisely HTTPS, since data transfer between API clients and Gemini NTS Exit APIs is secured. However, terms HTTP and HTTPS are used interchangeably without necessarily making the distinction.

2. API Client Development Guidelines

The following diagram illustrates a typical sequence of interactions between an API client and Gemini / NTS Exit Reform during the lifetime of a session:

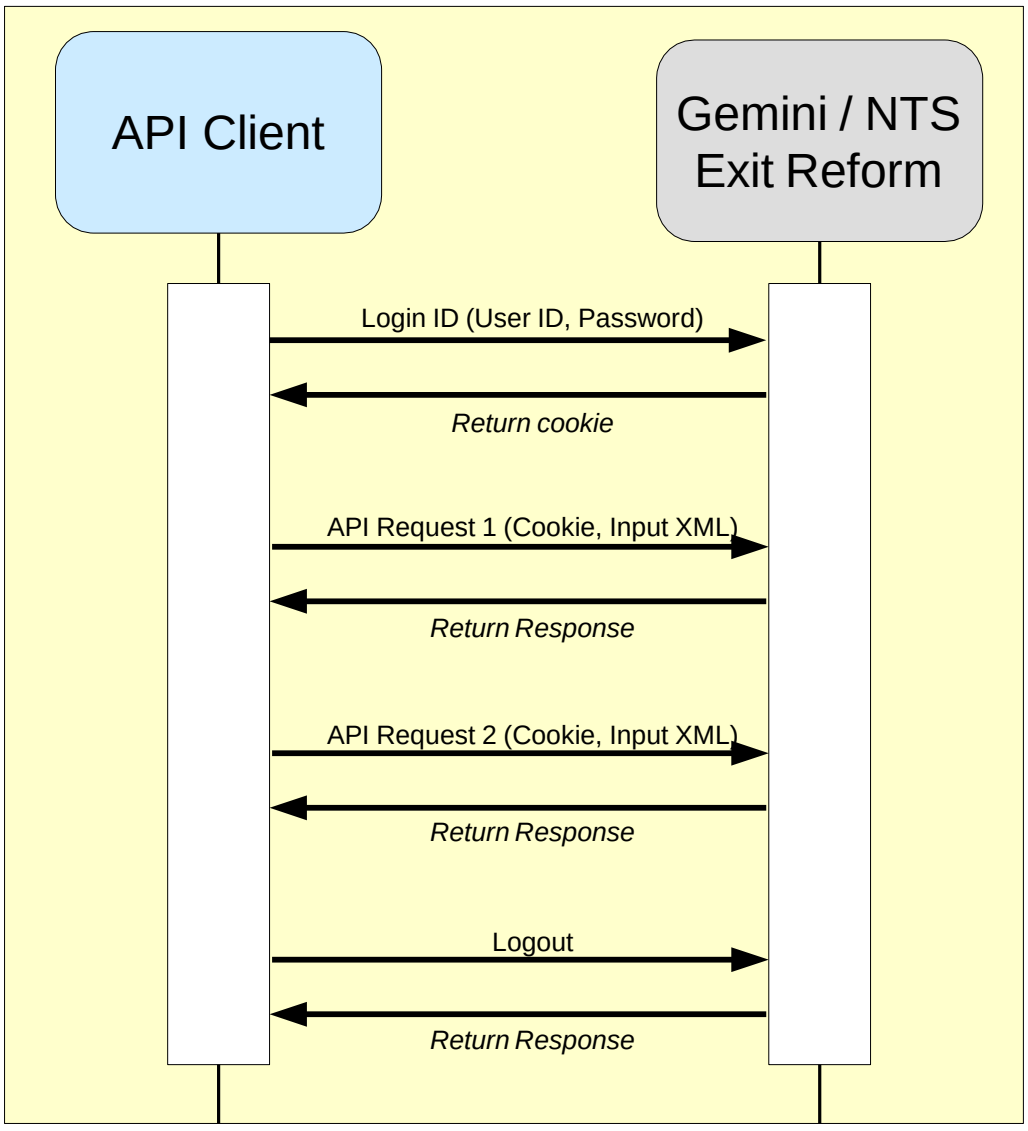


Figure 2: API Client Sequence of Interactions with Gemini / NTS Exit Reforms

The following is the sequence of interactions illustrated above:

- API clients are authenticated using login credentials (user id and password). To trigger this, the API client invokes the login URL and provides a user id and associated password. These are passed through the HTTP headers.
- Gemini / NTS Exit Reform authenticates the user credentials supplied by the API client and, if successful, returns a session cookie as a response. A session cookie is sent to the API client along with each HTTP response. API clients must send the latest received cookie for the session with each subsequent request for an API.

- The API client constructs an XML document containing the input query parameters (view API) or data (update API). The XML document must conform to the corresponding request schema definition of the API being invoked. All API schemas are located on a Xoserver server.

  Note: Update APIs are available only in respect to Gemini, and not NTS Exit Reform.

- The API client sends the XML document, along with the latest received cookies, to the URL specified for the required API.
- Gemini / NTS Exit Reform receives the request, parses the associated XML and services the requests. The response is sent as an XML document embedded in an HTTP message.
- The API client examines the HTTP response to check that the request was successful.
- If the request was successful, then the API client can extract the XML and interpret it as per the response schema definition of the API.
- If the request was not successful, then the API client must handle the reported error condition(s).
- One or more calls to the same or different APIs can be made in time sequence. The response (whether successful or not) to an API call must be received before submitting another API call within the same session.
- If the timeout period is exceeded between successive API calls then the session is timed out and the login step must be repeated to establish a new session before re-commencing API calls.
- The API client invokes the logout API to terminate the session when all desired requests/responses have been processed.
- Cookies must remain 'within session', that is cookies returned by Gemini / NTS Exit Reform in one session must be not be sent to Gemini / NTS Exit Reform in another session.

Further clarification, on a number of points relating to this interaction is provided below:

- HTTPS and SSL
- Authentication and Authorization
- Maintaining the Session
- Authentication/Authorization Failures
- Request a Compressed Response to your API Client

2.1. HTTPS and SSL

APIs are accessed using the HTTPS protocol for secure data transfer. Gemini / NTS Exit Reform web servers have certificates from a valid Certification Authority. API clients must validate and retain the Gemini / NTS Exit Reform web server certificates. SSL is used for access.

To develop API clients any of the SSL toolkits that support TLSv1 may be used (for example OpenSSL, JSSE (provided by Sun Microsystems), etc.).

2.2. Authentication and Authorization

Gemini and NTS Exit Reform are secure systems. As such, the URLs associated with both systems are protected[2]. To access a protected URL, the user must have 'authenticated' themselves and be 'authorized' to do so.

Authentication

This is the act of identifying yourself to Gemini / NTS Exit Reform by providing valid login credentials (user id and associated password).

Any request to a protected URL that does not supply a valid session cookie (see 2.3 maintaining the Session) will trigger authentication by Gemini / NTS Exit Reform. As such, it must include valid login credentials (see 3.1 Login API) if it is to succeed.

Authorization

This is the process of confirming that the user is permitted to invoke the functionality provided by the URL being accessed. Typically this is controlled via the roles assigned to the user.

It follows that authorization must follow authentication, since it is necessary to identify the user before confirming their access rights.

2.3. Maintaining the Session

HTTP is a stateless protocol. In HTTP client/server interaction cookies are used to maintain a state/session. API client interaction with Gemini / NTS Exit Reform follows this approach.

If an API request to Gemini / NTS Exit Reform does not result in an authentication/authorization failure, then Gemini / NTS Exit Reform returns session cookies in the API response. The next API request in that session must return the last received cookies.

It is essential that the last received cookies be returned. The API client must not, for example, attempt to send the cookies received at login if subsequent requests in that session have been processed. This will result in an authentication/ authorization failure, as the supplied session cookies will not be valid.

Remember also to remain synchronous within a session, as Gemini / NTS Exit Reform maintains only one set of state information pertaining to a session. Therefore, if you submit asynchronous requests within a session, you run the risk of corruption or loss of data.

2.4. Authentication / Loss of Session / Authorization Failures

To invoke the functionality associated with any protected Gemini / NTS Exit Reform URL, you must be authenticated and authorized. All Gemini / NTS Exit Reform functional APIs are invoked via protected URLs.

When accessing Gemini / NTS Exit Reform, failures can arise relating to the security infrastructure. These broadly fall into three categories:

- Authentication Failure
- Loss of Session
- Authorization Failure

[2] The exceptions to this are the URLs associated with the change password screen and the change password API. Of course, these services still require valid login credentials to be supplied.

Table 1, 'Authentication / Loss of Session / Authorization Failures', gives a detailed breakdown of possible causes of each of these failure categories. Each category is briefly explained below.

2.4.1. Authentication Failure

Authentication of the supplied user credentials has failed. It follows that this scenario will not occur if your request supplied a session cookie, whether valid or not, because in that instance any supplied user credentials will be ignored. If your request did not supply a session cookie, then Gemini / NTS Exit Reform will demand that valid user credentials be supplied.

2.4.2. Loss of Session

This arises when there is a problem with the session cookie supplied by the API client. In this case the session is lost.

If user credentials were also supplied, (that is along with an invalid session cookie), then Gemini / NTS Exit Reform will attempt to establish a new session using those credentials. However, a request would not normally supply user credentials along with a session cookie.

If a valid session cookie is supplied along with user credentials, then the user credentials are not checked.

2.4.3. Authorization Failure

The request has been made within a valid, successfully authenticated session. However, the user is not authorized to access the specific functionality requested. This will be because the access to the requested API is not granted within the roles assigned to the calling user.

2.5. Request a Compressed Response to your API Client

Your API clients can request that the response returned to them be compressed. This will result in a reduction (typically 50% to 60%) in the data trafficked to your API client.

To activate this compression, the API client must send the following HTTP request header value.

Name of Header: Accept-Encoding

Value of Header: gzip

The client must also be able to uncompress the response on receipt.

If your client does not send this HTTP request header value, then the response it receives will not be compressed.

Table 1: Authentication / Loss of Session / Authorisation Failures

Category	Cause(s)	Necessary Action	Further Remarks
Authentication Failure	Unknown user or Incorrect Password	Supply valid user credentials.	If you attempt to login in three times consecutively with an incorrect password then the API account will be locked.

	Maximum failed number of login attempts exceeded or user disabled	Raise a help line call via your Local Security Officer (LSO) to get the account released.	A user account will be locked if the maximum number of consecutive failed login attempts (currently set to three) is reached. See 'Unknown User or Incorrect Password'. One further login attempt is then allowed every 30 minutes.

			A user can be disabled because the account has been inactive for longer than a set period, if this is configured. Alternatively, a SiteMinder administrator may have disabled the account for some reason.
	Change Password Forced, Password Expired, Immediate Password Change Required	Change the user's password before proceeding, either via the change password screen, or via the change password API.	There are some subtle distinctions between the causes. <u>Change Password Forced</u> Occurs if no password policies are in place and an administrator resets a password while dictating, 'change on first use'. Gemini / NTS Exit Reform are configured with password policies in place (For example expiry after thirty days, no reuse within twelve months, etc.). Consequently, this cause should not occur. <u>Password Expired</u> Gemini / NTS Exit Reform are configured such that passwords will expire and must be changed after a set period of time. Password expiry will only take effect at the next login, i.e. if the password expires while you are in session then the session is not dropped. <u>Immediate Password Change Required</u> Similar to Change Password Forced, but this cause is expected to occur for Gemini / NTS Exit Reform as it corresponds to having password policies in place.
Loss of Session	Invalid Session, Revoked Session, Expired Session, Invalid Session Id	Repeat the login process to re-establish a valid session and proceed.	A session will expire if inactivity exceeds the session timeout period. This is currently configured to be after one hour of inactivity. Invalid session and invalid session identifier arise if the server cannot match the session cookie supplied with the request to a valid session. The most likely explanation is a bug in the API client's cookie handling.
Authorization Failure	User not Authorized	Either contact IS Security, via your LSO, to get the role assignments of the user id changed or access the correct API.	If this scenario occurs then the user id is not authorized to access the particular API requested. There are two possible explanations: 1. The user id does not have appropriate roles assigned, 2. The API client is attempting to access the wrong API.

Several of the scenarios above make reference to policies in place in respect of passwords or sessions.

These policies are detailed in the following table. This information is provided for guidance only and must not be considered binding or guaranteed. The same is true of any reference to these policies elsewhere in this document. These policies are configurable items and as such are subject to change, for example, as security policy dictates.

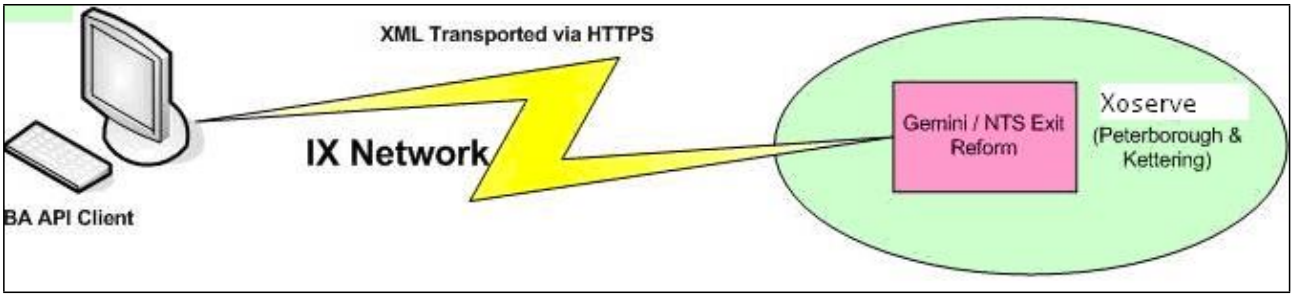
Table 2: Password and Session Policies

Policy Type	Policy Sub-Type	Policy
Password	Expiration	Passwords expire (password change forced) if not changed for 31 days. Expiration warnings are issued for five days ahead of a password expiry. Accounts are disabled after three successive attempts to login with an incorrect password. Thirty minutes later one further attempt is allowed. If an incorrect password is given again, then 30 minutes later one further attempt is allowed, and so on. A user id is disabled if not accessed for 90 days.
	Composition	Passwords must be between six and eight characters in length. No special characters; only lower and upper case letters and digits are allowed. At least one digit must be included.
	Restriction	– Before a password can be reused a minimum of 365 days must have elapsed or at least five other passwords must have been used since that password was last used.
Session	Expiry	– A session expires after sixty minutes of inactivity.

3. API Configuration Details

3.1. API Technology Overview

The following figure illustrates the API client communication with Gemini/NTS Exit Reform.



API clients specify a URL to access an API. These URLs are different from those of Gemini and NTS Exit Reform screens. API clients must issue a Hypertext Transfer Protocol Secure (HTTPS) request that contains API input parameters in Extensible Mark-up Language (XML) format. After processing the request, Gemini / NTS Exit Reform will return an HTTPS response with the output data, also in XML format, in the body.

Each API interaction is therefore a request/response pair. Communication between BAs and Gemini / NTS Exit Reform for API access is via the IX network.

The core technologies used to interact with Gemini / NTS Exit Reform APIs are Hypertext Transfer Protocol (HTTP1) and XML. HTTP is used as the communications protocol. XML is used to represent the data that must pass between the API client and the Gemini / NTS Exit Reform APIs to invoke the relevant business functionality.

[1] Or, more precisely HTTPS, since data transfer between API clients and Gemini NTS Exit APIs is secured. However, terms HTTP and HTTPS are used interchangeably without necessarily making the distinction.

Note 1: where the HTTP interaction between API client and Gemini / NTS Exit Reform APIs is described in this guide, this is in fact the standard interaction between a web client (normally a web browser) and a web server, supported by the HTTP protocol. In this respect, API client interaction with Gemini / NTS Exit Reform APIs differs only in that it is a two-way, XML-based message exchange alternative to the serving of HTML to a web browser.

Note 2: No XML is exchanged with the Session Management APIs, since they do not implement business functionality.

3.2. Network Requirements

- ✓ You must use port 4445 to submit requests to the Gemini API service. Therefore, if you connect to the IXN via a firewall you must have this port open.
- ✓ You must resolve the FQDN prod-ix.geminints.com to the IP address 194.129.160.242 on your network. Xoserve does not provide a DNS service for access via the IXN.
- ✓ Requests to 194.129.160.242 should be routed to Xoserve Data Centres via the IXN.

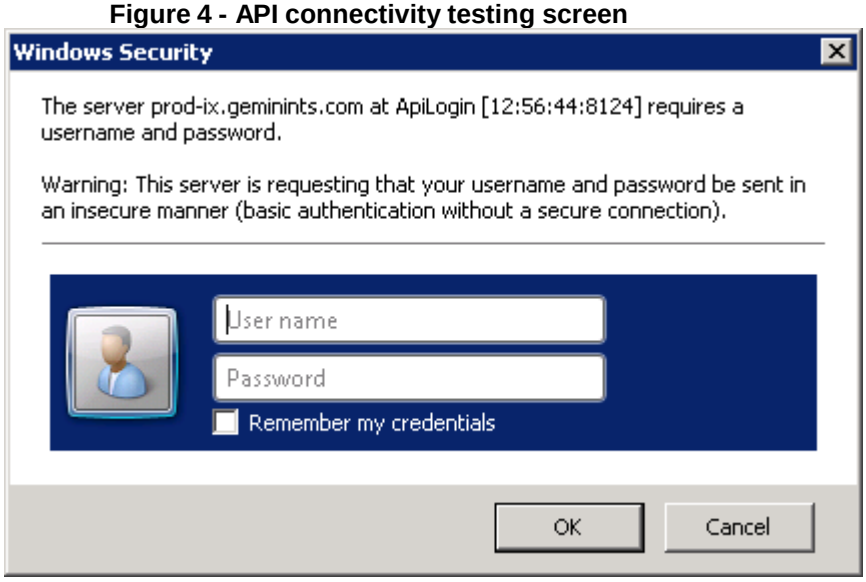
3.3. API Service Connectivity Confirmation

1. Open Internet Explorer on your desktop.

2. Enter the URL prod-ix.geminints.com:4445/gemini/controllers/APILogin

3. The following dialogue should appear. If it does then you've successfully confirmed API

Figure 4 - API connectivity testing screen



4. Once the above login page appears, the actual API login can be initiated from the API client from your desktop.

4. Specifications for Session Management APIs

This section contains specifications for the session management APIs. These are the following APIs:

- Login
- Logout

- Change Password

These are generic APIs that manage the sessions necessary to invoke any of the functional APIs. The functional APIs are the APIs that actually invoke Gemini / NTS Exit Reform business logic. Refer Section 5 Functional API Specifications for further detail on functional APIs.

Code Samples

Along with the specifications we provide code samples to illustrate the core session management activities. These code samples are written using the Java programming language.

National Grid does not mandate the use of Java to write API clients. API clients must honor the mandated behavior described in this document and the functional API specifications. The mandated behavior is defined in terms of HTTP interaction and associated XML message exchange. Provided your API client conforms to this mandated behavior, it can be constructed using any development technology.

Java has been used for illustrative code samples because:

1. It is in common use.
2. It does not abstract the underlying HTTP interaction so much as to mask the concepts that we are seeking to illustrate.

By contrast, Visual Basic API clients can be constructed that make use of the Microsoft provided XMLHTTPRequest object. This object encapsulates much of the default web browser behavior that, as explained previously, must be mimicked by API clients. In doing so it masks that behavior.

In summary, it is conceivable that the client interaction in your API clients may look very different from our Java samples, especially if you employ library objects that encapsulate much the default web browser behavior.

Note: Sample code is provided for guidance only. It does not constitute code licensed or supported by National Grid. It is not covered by any warranty. National Grid does not provide or support API clients. It does support the API interface and its conformance to specification.

4.1. Login API

The Login API is used solely to establish a Gemini / NTS Exit Reform session prior to invoking functional APIs. The Login API is simply a “dummy” protected URL which, when accessed, prompts the security infrastructure to authenticate and authorize the user. If authentication is successful, then a session is established.

If preferred, it is also possible to establish a session by calling one of the functional APIs. These URLs are also protected. If a request to any one of them does not contain a session cookie, then user credentials are checked and, if valid, a session will be established and the functionality of the API invoked in a single step.

The Login API simply allows the separation of these two steps, establishing a session and invoking API functionality.

However you choose to establish your sessions, it is important that you continue to use open sessions where possible, rather than creating new sessions, as this places an unnecessary burden on Gemini / NTS Exit Reform to the detriment of all users.

Note: Gemini / NTS Exit Reform user ids authorized for screen access will not generally be authorized for API access and vice versa. You must be sure to request a user id that is authorized to access APIs for use by API clients.

URL to Access the API

API clients must invoke the following URL to access this API's functionality:

/gemini/controllers/ApiLogin/

Note the trailing slash in this particular API URL. It is important that this is included.

4.1.1. HTTPS Request Headers

API clients can invoke the Login URL for authentication. They must provide login credentials as a user id and password passed through HTTP headers.

Table 3: Login API Request Headers

	Request Header Key	Value
1	Cookie	SMCHALLENGE=YES Required by the Computer AssociatesSiteMinder software to invoke authentication.
2	Authorization	HTTP basic authentication is adopted for authenticating the user. The user id and the password must be concatenated with a ":" delimiter, that is *userid: password*. The combined string must be encoded using Base64 encoding. The Base64 encoded value must be passed with this header. See http://www.ietf.org/rfc/rfc2617.txt for details of the Basic Access Authentication scheme and associated Base64 encoding.

Sample Code

```
/* Connect to Gemini Exit */

url = new URL("https://<server>:<port>/<LoginURL>");
urlConnection=(HttpsURLConnection)url.openConnection();
urlConnection.setRequestMethod("POST");

/* Request Headers with ID and password sent to Gemini Exit */

urlConnection.setRequestProperty("Cookie","SMCHALLENGE=YES");
String encodedLogin = base64Encode("userID", "password");
urlConnection.setRequestProperty("Authorization",  encodedLogin);
```

4.1.2. HTTPS Response Headers

Case 1: Successful Authentication/Authorization

On successful authentication/authorization the following response will be sent to the API client:

Table 4: Login API Response Headers - Successful Authentication/ Authorization

	Request Header Key	Value
1	Set-Cookie	GEMINIAPIAUTHENTICATION=2001 GEMINIAPIAUTHENTICATION is used to indicate by Gemini / NTS Exit Reform to indicate the authentication outcome of any HTTPS request. 2001 indicates success and 4001 indicates failure.
2	Set-Cookie	GEMINIAPIAUTHORIZATION=2002 GEMINIAPIAUTHORIZATION is used to indicate by Gemini / NTS Exit Reform to indicate the authorization outcome of any HTTPS request. 2002 indicates success and 4002 indicates failure.
3	Set-Cookie	SMSESSION Cookie containing encrypted session ID. API clients must send the latest received cookies with every subsequent API invocation request.

Successful authentication/authorization via the Login API will return an HTTP response 404 (file not found). 404 is returned on successful use of the Login API because it is a 'dummy' URL (see Section 3.1 Login API) with no response page to be served. API clients should trap this response to ensure that it is not handled as an error.

By contrast, successful invocation of functional APIs (whether including user login or not) will return a HTTP response 200 (OK) as functional APIs do correspond to an underlying, functional URL that serves a response page.

We have included some information on expected HTTP response codes in this guide for your information. It is recommended, however, that HTTP response codes are not used to infer the success or failure or API calls. Rather, API clients must be able to trap HTTP 400 and 500 series responses and handle them carefully, since they may not indicate an error.

To diagnose errors you should instead rely on a combination of the following:

- GEMINIAPIAUTHENTICATION/GEMINIAPIAUTHORIZATION to determine the authentication/ authorization success or failure status.
- SMAUTHREASON (see below) to assist in determining the course of action necessary in the event of authentication/ authorization failures.
- XML success or error responses (see 4 Error Handling) to detect and diagnose functionality errors.

Case 2: Authentication/Authorization Failed

In the case of authentication/authorization failure the following response will be returned to the API client.

Table 5: Authentication/Authorization Failed

	Request Header Key	Value
1	Set-Cookie	GEMINIAPIAUTHENTICATION=4001
2	Set-Cookie	GEMINIAPIAUTHORIZATION=4002 Only for session related authentication/ authorization failures. See **Error! Reference source not found.**
3	LOCATION	SMAUTHREASON=<value> Can provide additional diagnostic information on authentication/ authorization failure depending on the exact failure scenario. See "SMAUTHREASON" below and Table 6: Redirection Scenarios. **NB:** SMAUTHREASON is described here in relation to authentication/ authorization failure, as it is usually associated with this scenario. However, there is at least one instance (password expiry warning) in which SMAUTHREASON provides information in relation to a **successful** authentication.

SMAUTHREASON

In the event of an authentication/authorization failure, a redirect instruction is often returned in response to the client. Under the HTTP protocol, such a redirection instruction is conveyed in the LOCATION header field.

Typically, the redirect will be to the password maintenance service for user action. If Gemini / NTS Exit Reform screens are being used, then the web browser will follow this redirect instruction so that the user can take the necessary action (For example: change of password).

It is not appropriate for an API client to follow the redirect to a screen based service. However, the redirect instruction contains important information as to the cause of the authentication/authorization failure. This is encapsulated as a name/value pair parameter in the redirect URL. The parameter name is SMAUTHREASON.

Session related authorization/authentication failures (invalid session, revoked session, expired session and invalid session identifier) and the invalid user credentials scenarios do not trigger a redirect instruction in response to the client. In these situations, a password maintenance action is not appropriate. Furthermore, since APIs use basic and not forms-based authentication, there is no login form to redirect to for the purposes of re-establishing a session or correcting invalid user credentials.

In order to obtain supplementary information about the cause of failure, the API client must check for the LOCATION Response Header Key. The "SMAUTHREASON" name/value pair may contain supplementary information.

Sample Code

```
/* Response Header from Gemini Exit */

boolean failureFlag = false;

int count = 0;

while(urlConnection.getHeaderFieldKey(count++) != null)
```

```
{
      String sKey = urlConnection.getHeaderFieldKey(count);
      String sValue = urlConnection.getHeaderField(count);
      if(sKey.equals("Set-Cookie"))
      {
            if(sValue.indexOf("GEMINIAPIAUTHENTICATION=2001")!=-1)
            {
            System.out.println("Successful  authentication");
            }
            if(sValue.indexOf("GEMINIAPIAUTHORIZATION=2002")!=-1)
            {
            System.out.println("Successful  authorization");
            }
            if(sValue.indexOf("GEMINIAPIAUTHENTICATION=4001")!=-1)
            {
            System.out.println("Authentication  failed");
            failureFlag = true;
            }
            if(sValue.indexOf("GEMINIAPIAUTHORIZATION=4002")!=-1)
            {
            System.out.println("Authorization  failed");
            failureFlag = true;
            }
            /* In case of successful authentication/authorization store the
            session cookie */
            if(!failureFlag  &&  sValue.indexOf("SMSESSION")!=-1)
            {
            setLatestSessionCookie(sValue);
            }
      }
      /* LOCATION header key needs to be checked only in case of
      authentication/authorization failure to obtain supplementary information */
      if(failureFlag  &&  sKey.equalsIgnoreCase("LOCATION"))
            {
            if(sValue.indexOf("SMAUTHREASON=1")!=-1)
```

```
            {
            System.out.println("User must change password");
            }
      }
      /* Similar checks must be done for other SMAUTHREASON values */}
}
```

4.1.3. HTTP Redirection

HTTP redirection is characterized by one of the 3xx series of HTTP status codes. An explanation of redirection 3xx status codes, which includes a brief description of HTTP redirection, can be found at the W3C web site. The specific redirection status code that will be encountered in interactions with Gemini / NTS Exit Reform is the 302 Found status code.

To accurately diagnose the causes of authentication/authorization failure in your API clients, you must understand HTTP redirection. Furthermore you must use a technology that allows you to trap and interpret the “intermediate” (see below) response in a HTTP redirection.

In a web client/server interaction that does not involve a redirection 3xx status code there is one request/ response pair. The web client issues an HTTP request and receives an HTTP response and at that point the interaction ends.

This is illustrated in the following diagram:

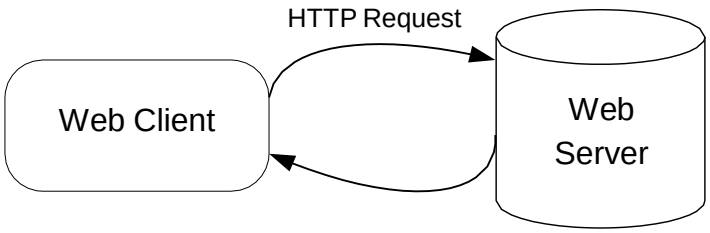


Figure 3: Interaction between Web Client and Web Server

In a web client/server interaction that does involve a redirection of 3xx status code, there are (at least) two request/response pairs. The web client issues an initial HTTP request and receives a “follow me” redirect HTTP response. This redirect response instructs the web client to issue a redirected HTTP request to a location that it specifies. When the web client follows this instruction via a redirected HTTP request, the web server issues the final HTTP response and the interaction ends.

This is illustrated in the following diagram:

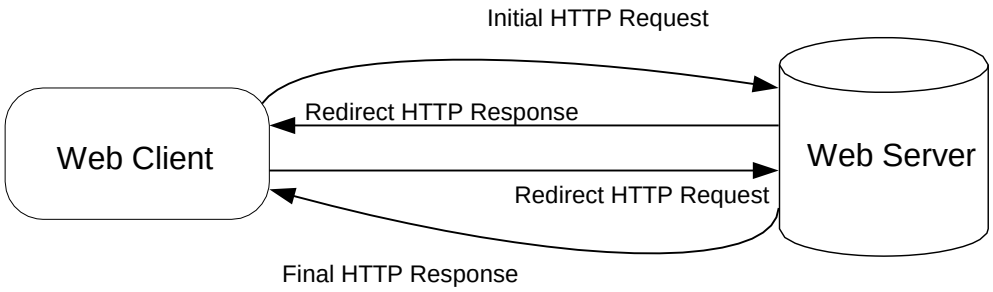


Figure 4: Interaction between Web Client and Web Server involving a redirection of 3xx status code

It should be noted that the behavior described is mandated for web clients/servers that implement the HTTP protocol as specified. In this respect the behavior expected of your API client is no different to the behavior expected, and honored, by any web browser. Web browsers and API clients are merely sub- classes of web clients expected to conform to the HTTP standards expectations of a web client.

The redirect HTTP response is characterized by having a 302-status code. It also contains the (Uniform Resource Indicator) URI that the web client is instructed to go to via a redirected HTTP request. This URI is specified in the LOCATION header field of the redirect HTTP response.

An explanation of the LOCATION header field can again be found at the W3C web site. Again, none of this is specific to Gemini / NTS Exit Reform.

API Client Considerations

Most objects simulating web client behavior will, by default, automatically handle redirect requests and only return the final response. Therefore, if the contents of a redirect HTTP response need to be trapped, then this must be facilitated by the choice of development tools for API clients. To illustrate this, we discuss two real life examples.

1. Use of the MSXML2.ServerXMLHTTP40 Object in Visual Basic

   API developers using the MSXML2.ServerXMLHTTP40 object within a Visual Basic API client have on occasion contacted us. They report that they cannot intercept and interpret the redirect response.

   Our investigations have revealed that this is a known problem with this object. Further information can be found on the following Internet forum:

   http://support.microsoft.com/default.aspx?scid=kb;EN-US;q308607

   Briefly, the advice given is that if you want to intercept redirects you should use the WinHttpRequest object instead.

   This is an example of an object that implements the standard behaviors of a web client and exposes them for a wraparound language to access. One such standard behavior is the following of a redirect. The point illustrated is that if the object you use does not allow the wrap around language to trap and interpret the redirect, then this behavior is not exposed.

2. Use of the HttpsURLConnection Object in Java

   Similarly the HttpsURLConnection object in Java will, by default, follow redirects to the final response and deny the Java code sight of the redirect response and redirected request.

   This behavior can be changed using the setInstanceFollowRedirects method, as follows:
   HttpsURLConnection httpsConn = (HttpsURLConnection)url.openConnection();
   httpsConn.setInstanceFollowRedirects(false);
   Where URL is of the URL object type.

Gemini / NTS Exit Reform Redirection Scenarios

NB: This section is deliberately not called Gemini / NTS Exit Reform API Redirection Scenarios, as these redirections are not specific to APIs. They are encountered when accessing the screens but are also silently handled by your web browser without your knowledge.

Firstly, in all of these redirection scenarios there will only be a single redirection before the final response is reached. The HTTP protocol allows for multiple redirections.

The scenarios in which you can expect to receive a redirection are as follows:

1. Any authentication/authorization failure scenario that provides further information via an SMAUTHREASON code.
2. The password expiry warning response. In this instance, there has not been an authentication/ authorization failure but a warning that password expiry is imminent is notified via an SMAUTHREASON code in the LOCATION header field.
3. The Logout API issues a redirect request when used to terminate a session.

The complete behavior in these scenarios is detailed in the following table.

Table 6: Redirection Scenarios

	HTTP Response				
	Redirect				Final
	HTTP Status	SMAUTHREASON	GEMINIAPIAUTHENTICATION	GEMINIAPIAUTHORIZATION	HTTP Status
Authentication / Authorization Failure	302		4001	*None*	200
Maximum Failed Number of Login Attempts Exceeded		24[3]			
User Disabled (by Administrator)		7			
User Disabled (Due to Inactivity)		25			
Change Password Forced		1			
Password Expired		19			
Immediate Password Change Required		20			
Password Expiry Warning4		18	2001		
Logout API		0	*No*		

[3] This will appear on the third consecutive login attempt with an incorrect password. One further attempt is then allowed every half hour. This too will return SMAUTHREASON=24 if the password is again incorrect.

Notes:

- GEMINIAPIAUTHENTICATION header field is not present in the final response following a redirection.
- Any SMAUTHREASON value returned present in the LOCATION header of the redirect response will then appear in the URL query string of the final response. This is as a direct consequence of following the redirect.
- In the case of a Password Expiry warning, if you automatically follow the redirect to the final URL then no session cookies will be returned. As a consequence you will not be able to call functional APIs. If you wish to ignore the password expiry warning and change your password at a later time then you must intercept the redirect response in order to use the session cookies provided.

For completeness, the results of the authentication/authorization failure scenarios that do not involve redirection are included in the following table.

Table 7: Scenarios not involving a Redirection

	HTTP Response		
	HTTP Status	**GEMINIAPIAUTHENTICATION**	**GEMINIAPIAUTHORIZATION**
Authentication / Authorization			
Unknown Unser	401	4001	*None*
Incorrect Password			
User not Authorized		*None*	
Session			
Invalid Session	401	4001	4002
Revoked Session			
Expired Session			
Invalid Session ID			

[4] Although it is detailed in this table, this is not actually an authentication/loss of session/authorization failure scenario. A password expiry warning informs the client that the password is due to expire shortly. However, authentication has been successful.

Note that loss of session failures would not be expected to occur when accessing the Login API, since the sole purpose for accessing the Login API is to establish a session. In this instance, no session cookies associated with an existing session would be supplied. Session loss would normally be associated with functional API calls within the session. However, the full list of response codes has been presented together, including the Login API specification.

An illustration of how logic might be constructed to determine the cause of authentication/loss of session/authorization failures is given below. Remember, this diagram is provided to illustrate a possible not mandated or recommended approach.

4.1.4. HTTPS Response Body

The HTTPS response body returned by the login API is null.

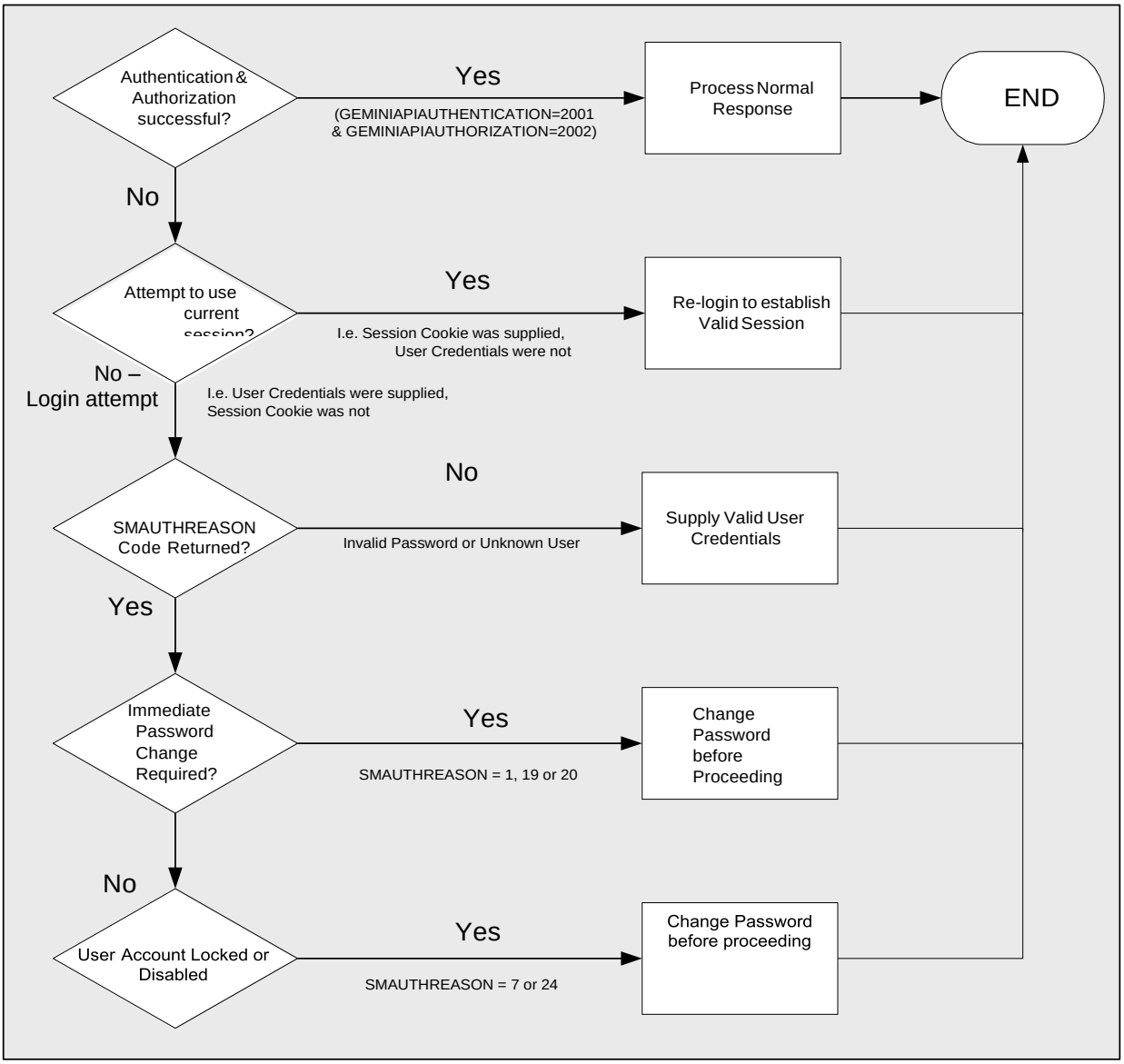


Figure 5: Example Logic for Determining the Cause of Authentication/Loss of Session/ Authorization Failures

4.2. Functional APIs

The basic principles of invoking functional APIs are described here. The specifications of each of the functional APIs reside in separate documents; see 5 Functional API Specifications.

4.2.1. HTTPS Request Headers

The cookies that were last received by the API client must be passed via HTTPS headers on the next API call. API clients must use the POST method to submit their requests.

Note: When invoking functional APIs you must not set a value of “text/xml” for the Content-Type header key. This can result in GEM_API_ERROR_0001 / NEX_API_ERROR_0001 messages (“XML document is not valid”) being returned for perfectly valid XML.

Table 8: Functional API Request Headers

	Request Header Key	Value
1	Cookie	All the latest received cookies provided by Gemini / NTS Exit Reform web servers must be returned with every subsequent API invocation request.

Sample Code

```
/* Connect to Gemini Exit */

url = new URL("https://<server>:<port>/<API-URL>’’);
urlConnection=(HttpsURLConnection)url.openConnection();
urlConnection.setRequestMethod("POST");
int count = 0;
while(urlConnection.getHeaderFieldKey(count++) != null)
{
      String sKey = urlConnection.getHeaderFieldKey(count);
      String sValue = urlConnection.getHeaderField(count);

      if(sKey.equals("Set-Cookie"))
      {
            urlConnection.setRequestProperty("Cookie", sValue);
      }
}
```

4.2.2. HTTPS Request Body

The API client must send an XML document that contains the input parameters required by the specific API being called[5]. This is passed via the INPUT name/value pair in the HTTPS request body. For a single API request only one name/value pair can be sent.

Table 9: Functional API Request Body

	Request Header Key	Value
1	INPUT	Valid XML input document that contains input parameters sent to the server for processing in response to the request.

Sample Code

```
/* Request body sent to Gemini Exit */

String strXMLParams = "INPUT="+xmlDocument.toString();
outputStream.write(strXMLParams.getBytes());
outputStream.flush();
```

[5] There is one exception to this principle and that is the System Status Information API. This API requires no input data since it simply returns the latest system status information.

4.2.3. HTTPS Response Headers

The following table describes the Functional API Response headers.

Table 10: Functional API Response Headers

	Request Header Key	Value
1	Set-Cookie	SMSESSION and other values. (All latest received cookies provided by Gemini / NTS Exit Reform web servers must be returned with every subsequent API invocation request. It is not necessary to understand or interpret the Set-Cookie content, as long as the general principle of echoing it back is followed.)

Sample Code

```
/* Response header from Gemini Exit */

int count = 0;
while(urlConnection.getHeaderFieldKey(count++) != null)
```

```
{
      String sKey = urlConnection.getHeaderFieldKey(count);
      String sValue = urlConnection.getHeaderField(count);

      if(sKey.equals("Set-Cookie"))
      {
            /*Retrieve All the Cookies send as response.
            These cookies must be returned with every subsequent API
            invocation request. */

      }
}
```

4.2.4. HTTPS Response Body

A valid XML document will be sent as the response body. This XML will conform to the output schema definition of the corresponding API. Any failure to do so should be reported to National Grid as an error.

4.3. Logout API

The Logout API is a 'session killer'. If it is invoked with a valid session cookie, then that session is logged out. It is a good practice that when you have finished using a session you invoke the Logout API. Inactive sessions will eventually be time expired, but using the Logout API helps to manage the number of open sessions.

URL to Access the API

API clients must invoke this URL to access this API's functionality:

/home/common/jsp/smlogout.jsp

4.3.1. HTTPS Request Headers

The following table describes the HTTP Request Header.

Table 11: HTTPS Request Headers

	Request Header Key	Value
1	Cookie	All the latest received cookies provided by the Gemini / NTS Exit Reform web servers must be returned with the request to the Logout API.

When successful (that is session has been logged out), the logout API return a 302 HTTP response code.

4.4. Change Password API

This API permits the Gemini / NTS Exit Reform application password changes. User Id, Old Password and New Password are required as input parameters of the HTTP Post request. On execution of this API, the API

Client will receive an HTTP response with either the confirmation of the password change, or an error message. The implementation of this API differs from the other Gemini / NTS Exit Reform functional APIs (presented in this document) as it invokes SiteMinder DMS APIs for the password change.

URL to Access the API

API clients must invoke this URL to access this API's functionality:

/gemini/controllers/ChangePasswordControllerAPI/

4.4.1. HTTP Request Headers

No headers identified.

4.4.2. HTTP Request Body

The API client passes the following input parameters as the HTTP request body.

Table 12: Change Password API Request Parameters

Name	Value
USER_ID	String containing User Id
OLD_PASSWORD	String containing old password
NEW_PASSWORD	String containing new password
CONFIRM_NEW_PASSWORD	String containing new password

4.4.3. Sample Request Code

Sample Request coded in Java:

```
/* Connect to Gemini Exit */
url = new
URL("https://<server>:<port>/gemini/controllers/ChangePasswordControllerAPI");
urlConnection = (HttpsURLConnection)url.openConnection();
urlConnection.setRequestMethod("POST");

/* Declare and set the string variables USER_ID, OLD_PASSWORD, NEW_PASSWORD
and CONFIRM_NEW_PASSWORD with the appropriate values */

/* Request body sent to Gemini Exit */
OutputStream outputStream = urlConnection.getOutputStream();
outputStream.write( ("USER_ID=" + USER_ID + "&").getBytes());
outputStream.write( ("OLD_PASSWORD=" + OLD_PASSWORD + "&").getBytes());
outputStream.write( ("NEW_PASSWORD=" + NEW_PASSWORD + "&").getBytes());
outputStream.write( ("CONFIRM_NEW_PASSWORD=" +
CONFIRM_NEW_PASSWORD).getBytes() );
outputStream.flush();
```

4.4.4. HTTP Response Body

The following response codes will be returned as part of the HTTP response body.

Table 13: Change Password API Response Codes

Response Code	Message
GEM_API_SEC_1000	Your new password has been set. Use this new password the next time you log into your account.
GEM_API_SEC_ERR_1001	Your password change was not accepted. Please try again.
GEM_API_SEC_ERR_1002	Please match your new password and confirmation.
GEM_API_SEC_ERR_1003	Access is restricted to authorized users only.
GEM_API_SEC_ERR_1004	System encountered an error. Please try again after some time.
GEM_API_SEC_ERR_1005	You cannot access your account because you have exceeded the limit of login attempts. Please contact your Security Administrator or Help Desk.

4.4.5. Sample Response Code

```
/* Process response code in API Client program */

/* It is assumed the URL has been requested. Please refer to the earlier code
snippet */

InputStream is = urlConnection.getInputStream();
InputStreamReader ins = new InputStreamReader(is);
BufferedReader br = new BufferedReader( ins );
/* To read the response code */
String CODE = br.readLine();

/* To read the message */
String MESSAGE = br.readLine();

If ( CODE.equals(GEM_API_SEC_1000) )
{
     System.out.println("Password Changed successfully");
     ...
     ...
     ...
```

```
}


br.close();
ins.close();
is.close();


urlConnection.disconnect();
```

5. Error Handling

It is useful to consider error handling by Gemini / NTS Exit Reform APIs in terms of hierarchy, with the most severe errors at the top and the least severe at the bottom. This is illustrated in the following diagram:

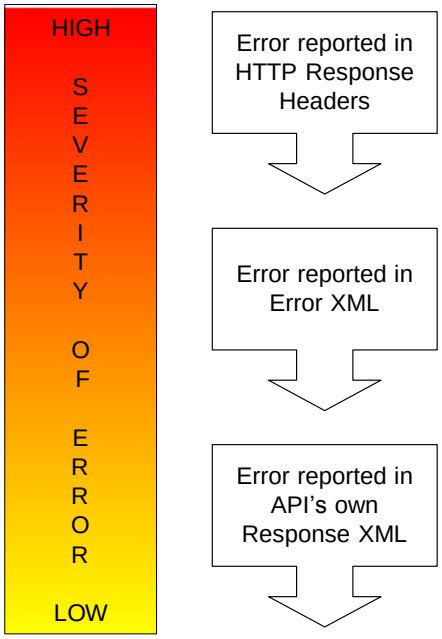


Figure 6: Error Handling by Severity of Error

Errors reported via the HTTP response headers will generally relate to authentication/authorization failures. See Sections 2.4 Authentication / Loss of Session / Authorization Failures and 3.1.2 HTTPS Response Headers for further information on how to recognize and handle these errors.

Assuming that an error of this severity has not occurred, any errors will be reported via XML. API clients can check the HTTPS response header "content-type" for the value "text/xml" to determine if the response from the API contains an XML document. If the response does not contain an XML document, then this header value will be set to something other than "text/xml".

Note that errors reported via XML are not relevant to the session management APIs (see Section 3 Specifications for Session Management APIs) as these APIs do not exchange XML with the API client.

The rest of this Error Handling section concerns errors reported via XML.

5.1. Errors Reported via XML

Generally, when the processing of a request from an API Client fails, an XML response is returned containing the error message in an Error XML structure. This Error XML structure is defined in the following sections. There is however an exception to this rule for APIs that add or modify Gemini / NTS Exit Reform data.

Update APIs process one record at a time, and a success or failure message can be returned for each record in the specific API's own response XML. This is explained further through the scenarios detailed below.

Note: Update APIs are available only in respect to Gemini, and not NTS Exit Reform.

Query APIs

For APIs that query data from Gemini / NTS Exit Reform, API Clients can expect one of the following three scenarios:

Table 14: Error Scenarios for Query APIs

Scenario		Response XML	Error XML
a)	Successful Query – A successful transaction that returns the query results in the API's response XML as defined in the relevant API specification under "API Response" for that API.	Yes	No
b)	Generic Error – A generic or system error (example Invalid XML) that prevents the transaction from executing successfully. No query results are returned. An Error XML containing the appropriate error code is generated and returned to the API Client.	No	Yes Error Codes in Section 4.4
c)	API Specific Error – An API specific error (example Invalid Meter Id for the BA) that prevents the transaction from executing successfully. No query results are returned. An Error XML containing the appropriate error code is generated and returned to the API Client.	No	Yes Error Codes in Individual API Specifications

Update APIs

Note: Update APIs are available only in respect to Gemini, and not NTS Exit Reform.

For APIs that add or modify Gemini data, one of the following scenarios can occur:

Table 15: Error Scenarios for Update APIs

Scenario		Response XML	Error XML
a)	Fully Successful Update – A transaction in which **all** input records were successfully added or updated. The API Client receives a response XML as defined in the relevant API specification under "API Response" for that API. Against each input record, this XML contains Message Code (MSG_CD) and Message Description (MSG_DESC) indicating that the record was updated successfully. Response codes for these messages are of the format **GEM_API_MSG_nnnn**.	Yes Response Codes in Individual API Specifications	No
b)	Partially Successful Update – A transaction in which **some** input records were successfully added or updated, while others were erroneous. The API Client receives a response XML as defined in the relevant API specification under "API Response" for that API. Against each input record, the status for that record is provided through the Message Code (MSG_CD) and Message Description (MSG_DESC) elements. Records that were updated successfully have response codes of the format **GEM_API_MSG_nnnn**, while records that were not updated contain error codes of the format **GEM_API_ERROR_nnnn**, along with a message describing the error.	Yes Response Codes in Individual API Specifications	No
c)	Unsuccessful Update – A transaction in which **none** of the input records were successfully added or updated. The API has processed each input record, but none of the records were successful. The API Client receives a response XML as defined in the relevant API specification under "API Response" for that API. Against each input record, the error for that record is indicated through the Message Code (MSG_CD) and Message Description (MSG_DESC) elements. Error codes are of the format **GEM_API_ERROR_nnnn**.	Yes Response Codes in Individual API Specifications	No
d)	Generic Error – A generic or system error (example Invalid XML) which prevents the transaction from executing successfully. The API has not processed any input records. No records are updated. An Error XML containing the appropriate error code is generated and returned to the API Client.	No	Yes Error Codes in Section 4..4

5.2. Error XML Specification

The following table describes the specifications of Error XML.

Table 16: Error XML Specification

Hierarchy	Data Element	Description	Data Type	Data Length	Mandatory
0	Errors	Top-level hierarchy for errors.			

1	errInfo	Top-level hierarchy for error elements. One or many errors may be returned. Attribute 'ID' of this element, shows the sequence identifier of the error.			
2	errCode	Error Code	String	18	Y
2	errDesc	Error Description	String	400	Y

5.3. Error Schema Definition

URL of file (Gemini): **/gemini/api/schema/geminiapierror.xsd**

(NTS Exit Reform): **/exit/api/schema/exapierror.xsd**

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="errors">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="errInfo" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="errCode" type="xs:string"/>
                            <xs:element name="errDesc" type="xs:string"/>
                        </xs:sequence>
                        <xs:attribute name="ID" type="xs:int" use="optional"/>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

5.4. Generic Error Codes

The table below lists error codes that may be returned by a call to any API; where the GEM prefix relates to Gemini, and the NEX prefix relates to NTS Exit Reform. These errors are returned in the Error XML specified in Sections 4.2 and 4.3.

Note: although error codes will not change, error message text is subject to revision without formal notice.

Table 17: Generic API Error Codes

Error Code	Error Message
GEM_API_ERROR_0000 / NEX_API_ERROR_0000	System error
GEM_API_ERROR_0001 / NEX_API_ERROR_0001	XML document is not valid[6]
GEM_API_ERROR_0002 / NEX_API_ERROR_0002	No record(s) found
GEM_API_ERROR_0003 / NEX_API_ERROR_0003	The record could not be saved.
GEM_API_ERROR_0007	Request not serviced. Usage limit exceeded.

[6] BAs may use XML schema definitions published by National Grid to validate input and output XML documents. This is particularly important for your input XML. You are responsible for ensuring that this conforms to the XSD specification.

6. Functional API Specifications

Gemini / NTS Exit Reform API specifications for functional APIs are contained in separate documents, one for each API. The sections contained within these documents are as follows:

- Overview
- API URL
- XSD URL
- Request XML Specification
- Response XML Specification
- API Specific Error Messages

API URL

This is the URL that is used in the API request to access specific information.

All URLs (both API and XSD) are given relative to the server root path for the API service.

Absolute URLs are not given. These will reflect, for example, National Grid's latest domain policy and are particularly susceptible to change. They are therefore communicated separately from the UK-Link controlled documentation set.

API client developers are advised to parameterise URLs.

XSD URL

The URLs of the request and response XSD associated with the API.

Request and response XSDs are made available on the Gemini / NTS Exit Reform web servers for API developers to use to validate the format of their XML input documents[7]. This is separate from and can be used in addition to the error messages returned by APIs, see Section 4 Error Handling.

Request XML Specification

This defines the XML structure that requests to the API must conform to. This is shown both as actual XSDs and in a more descriptive tabular format.

Note: Data Type columns in the tables are general descriptions for the benefit of the reader. For exact XML data types, please refer to the corresponding Schema Definition.

Unless specified otherwise the values for the XML fields identified in these documents should not be padded.

Response XML Specification

This defines the XML structure that will be sent to the API client from Gemini / NTS Exit Reform.

Note: Data Type columns in the tables are general descriptions for the benefit of the reader. For exact XML data types, please refer to the corresponding Schema Definition.

In some instances a data length is not specified in the descriptive tables for response data elements. This is where the length is not known in advance, (example dependent on calculation or not directly taken from a database record, etc.) and so the upper limit is not known in advance.

[7] The error XSD is also available on our web servers. See Section 4.3 'Error Schema Definition' in this document. Of course, the error XSD is not specific to particular functional APIs.

Error Handling

Contains details of error messages specific to the API in question, including details of where returned error messages would appear.

Note: For Update APIs, messages indicating a successful update are also defined.

6.1. API Scope

Note that the implementation date of different APIs may vary. You should refer to the individual API specification documents to understand when that API is available from.

6.1.1. Entry Capacity

The following table describes Entry Capacity APIs.

Table 18: Entry Capacity

API Name	Equivalent Gemini Online Screen	Accessible by
All Active Bids	Deal → Bid Capture Reports → All Active Bids	Shipper
Bid Information Shipper View – Normal Pricing Strategy	Deal → Bid Capture Reports → Bid Information	Shipper
Capacity Trade Details	Trade → Entry Capacity Trade → Trading → Query Trade Details	Shipper
Capacity Trade Registration	Trade → Entry Capacity Trade → Trading → Register Trade	Shipper
Daily Auctions Summary Report – Bids / Shippers	Product → Publish Reports → MoS Results → Bids / Shippers	Shipper
Daily Auctions Summary Report – Max / Min Price Information	Product → Publish Reports → MoS Results → Max / Min Price Info	Shipper
Daily Auctions Summary Report – Sold / Unsold	Product → Publish Reports → MoS Results → Sold / Unsold	Shipper
Daily Auctions Summary Report – WAP	Product → Publish Reports → MoS Results → WAP	Shipper
Revised Entitlements	Product → Entitlements-NET	Shipper
IP Entry Entitlement API	Product → Entitlements-NET	Shipper

6.1.2. Energy Balancing

The following table describes Energy Balancing APIs.

Table 19: Energy Balancing

API Name	Equivalent Gemini Online Screen[8]	Accessible by
Add / Update Nominations	Nominations → Nominations → (Update)	Shipper
Confirm Multi Locational	OCM → Maintain OCM Renominations → (Confirm	Shipper

API Name	Equivalent Gemini Online Screen[8]	Accessible by
Trades (Update)	Multi Loc)	
Daily Cashout Tolerance Breakdown	Allocations → Cashout → Cashout Tolerance Breakdown	Shipper
Gas Trades Breakdown	OCM → Trade Details → OCM / OTC Trade Breakdown Summary	Shipper / Market Operator
Maintain OCM Renominations (View)	OCM → Maintain OCM Renominations	Shipper
Maintain Physical Renominations (Update)	OCM → Maintain OCM Renominations → (Confirm NBP Phy)	Shipper
Meter to Zone Relationship	Meter Details → Setup Meter Details → Maintain Relationship	Shipper / Claims Validation Agents
Price Information History	OCM → Price Information History	Shipper / Market Operator
Register Physical/ Locational Trades	OCM → Physical Locational Trades	Market Operator
Register Title Swaps	OCM → Title Swap Trades → Register Title	Market Operator
Shipper EOD Noms (Hour Bar)	Nominations → INS → BA EOD/INS Imbalance	Shipper
Shipper Preliminary Balance	Nominations → Balance → Business Associate Balance	Shipper / Market Operator
Shipper Total Energy Forecast	Nominations → Demand Attribution → BA Total Energy Forecast	Shipper
System Nomination Balance	Nominations → Balance → System Balance	Shipper
System Status History	Nominations → Balance → System Status Information - History	Shipper
System Status Information	Nominations → Balance → System Status Information	Shipper
Update INS Nominations	Nominations → INS → Nominations → (Add or Modify)	Shipper
Update Renominations	Nominations → Renominations (Update)	Shipper
View INS Nominations	Nominations → INS → Nominations	Shipper
View Renomination Details	Nominations → Renominations → (V)	Shipper
View Renominations	Nominations → Renominations	Shipper
View Shipper Trade Details	OCM → View OCM Trade Details → View Shipper Trade Details	Shipper
View Storage Output Claims	Allocations → Pre Closeout Claims → Storage Output Claims	Claims Validation Agents
View WCF/SF Values	Allocations → LDZ → Demand Attribution → WCF / SF	Shipper
Add Update IP Nomination	Nominations → EU Nominations → Update	Shipper
View IP Nomination	Nominations → EU Nominations	Shipper

6.1.3. NTS Exit Reform Capacity

The following table describes the NTS Exit Capacities APIs.

Table 20: NTS Exit Capacity Functional APIs

API Name	Equivalent NTS Exit Reform Online Screen	Accessible by
All Active Requests	Deal → Capture → Request Information → All Active Requests Report	Shipper / DNO
Auction Request Information	Deal → Capture → Request Information → Auction Request Information Report	Shipper / DNO
Max / Min Price Information – Daily Auction	Publish→ Reports → MoS Reports → Post Allocation Information Reports → Max/Min Price Info Report	Shipper / DNO
Sold / Unsold – Daily Auction	Publish→ Reports → MoS Reports → Post Allocation Information Reports → Sold/Unsold Report	Shipper / DNO
Entitlement Report	Publish→ Reports → User Reports → Entitlement Report	Shipper / DNO / Allocation Agent
IP Exit Entitlement	Publish → Reports → User Reports → Entitlement Report	Shipper

6.2. Date Formats

Gemini / NTS Exit Reform API schemas use the standard data type formats defined by W3C. All date fields use the CCYY-MM-DD format. All timestamp values use the CCYY-MMDDThh:mm:ss format.

- **C** represents a digit used in the thousands and hundreds components, the "century" component of the time element "year". Legal values are from 0 to 9.
- **Y** represents a digit used in the tens and units components of the time element "year". Legal values are from 0 to 9.
- **M** represents a digit used in the time element "month". The two digits in an MM format can have values from 1 to 12.
- **D** represents a digit used in the time element "day".
    - The two digits in a DD format can have values from 1 to 28 if the month value equals 2
    - 1 to 29 if the month value equals 2 and the year is a leap year
    - 1 to 30 if the month value equals 4, 6, 9 or 11 and
    - 1 to 31 if the month value equals 1, 3, 5, 7, 8, 10 or 12.
- **h** represents a digit used in the time element "hour". The two digits in an hh format can have values from 0 to 23.
- **m** represents a digit used in the time element "minute". The two digits in an mm format can have values from 0 to 59.
- **s** represents a digit used in the time element "second". The two digits in an ss format can have values from 0 to 59.

7. Element Name Abbreviations

The following abbreviations for API element names are used throughout the API specifications. We include tables giving the full names here for convenience.

7.1. Entry Capacity APIs

The following table describes the Entry Capacity APIs.

Table 21: Entry Capacity APIs

Shortened Element Name	Element Description
BIDS_OUT	All Active Bids Output
ALL_ACTV_BIDS_QRY	All Active Bids Query
BID_CPCTY	Bid Capacity
BID_INFO_NRML_PRCNG_OUT	Bid Information Normal Pricing Output
BID_INFO_NRML_PRCNG_QRY	Bid Information Normal Pricing Query
BID_PRC	Bid Price
BID_STS	Bid Status
BID_TMSTMP	Bid Time Stamp
BID_TYP	Bid Type
BID_VAL	Bid Value
BIDWND_END_DT	Bid Window End Date
BIDWND_STRT_DT	Bid Window Start Date
BUY_BA	Buying BA
CNTRCT_TYP	Contract Type
CPCTY_DAY	Capacity Day
CPCTY_TRD_DTL_OUT	Capacity Trade Detail Output
CPCTY_TRD_DTL_QRY	Capacity Trade Details Query
CPCTY_TRD_REGN_INP	Capacity Trade Registration Input
CPCTY_TRD_REGN_OUT	Capacity Trade Registration Output
ENTLMNT	Entitlement
ENTLMNT_PRC	Entitlement Price
ENTLMNTS_OUT	Entitlements Output
ENTLMNTS_QRY	Entitlements Query
ERLST_EXRCSBL_DAY	Earliest Exercisable Day
ERLST_EXRCSBL_TM	Earliest Exercisable Time
EXER_PRC	Exercise Price
FLOW_STRT_TM	Flow Start Time
LCTN	Location

LCTNS	Locations
LTST_EXRCSBL_DAY	Latest Exercisable Day
LTST_EXRCSBL_TM	Latest Exercisable Time
MAX_QTY	Maximum Quantity
MIN_QTY	Minimum Quantity
MOS	Method Of Sale
NET_CPCTY	Net Capacity
NET_FRM_CPCTY	Net Firm Capacity
NET_INTR_CPCTY	Net Interruptible Capacity
NET_SCLD_INTR_CPCTY	Net Scaled Interruptible Capacity
OPTN_BID	Option Bid
PRD_FRM	Period From
PRD_TO	Period To
PRDT	Product
PRDT_CTGRY	Product Category
PRDT_TYP	Product Type
PRM_PRC	Premium Price
QRY_CRTR_1	Query Criteria 1
QRY_CRTR_2	Query Criteria 2
QRY_CRTR_3	Query Criteria 3
RMNG_EXRCSBL_DAYS	Remaining Exercisable Days
RSN_FOR_RJCTN	Reason For Rejection
SELL_BA	Selling BA
SRVC_TYP	Service Type
SUBTNDR_ID	Sub Tender ID
SUBTX_END_DT	Sub Transaction End Date
SUBTX_PRD_FRM	Sub Transaction Period From
SUBTX_PRD_TO	Sub Transaction Period To
SUBTX_STRT_DT	Sub Transaction Start Date
TMSTMP	Time Stamp
TRD_PRC	Trade Price
TRD_QTY	Trade Quantity
TRD_REF_NMBR	Trade Reference Number
TRD_STS	Trade Status
TRD_TM	Trade Time
TRNCH_NMBR	Trance Number
TX_END_DT	Transaction End Date

TX_STRT_DT	Transaction Start Date

7.2. Entry Balancing APIs

The following table describes the energy Balancing APIs.

Table 22: Energy Balancing APIs

Shortened Element Name	Element Description
ACTV_IND	Active indicator
ACTVTY	Top level hierarchy for activity elements
ACTVTY_NBR	Activity number
AGNT	Agent for whom the storage output claims are retrieved
AGNT_NM	Indicates the agent name.
ALLCTD_FCAST	Allocated forecast.
ALLOC_CPCTY	Allocated Bid Capacity
AMT_BGT	Amount Bought
AMT_SALE	Sale Amount
BA_CD	The BA three character short code
BID_CPCTY	Bid Capacity
BID_ID	Bid Identifier
BID_ORGNTOR	Unique identification code for a BA.
BID_STS	Bid Status
BID_TMSTMP	Bid Timestamp
BID_TYPE	The field indicating whether the bid is a locational or a physical bid
BIDWND_END_DT	Bid Window End Date
BIDWND_STRT_DT	Bid Window Start Date
BTCH_ID	Identification for Transco physical/ locational Trades
BUY_DTL	Top-level hierarchy for buy details.
BUY_SELL	Buy Sell Indicator
BUY_SELL_IND	Buy Sell Indicator
CLM_DTL	Top-level claim details.
CLMD_QTY	Indicates the claimed quantity
CLNDR_DAY	Calendar Date
CLRNG_PRC	Clearing Price
CMT_TMSTMP	Commit timestamp
CNG_SNCE_LST_NM	The percentage change since the last NDMA Forecast.
CNTRCT_TYP	Contract Type

CRT_DT	Creation date
CRT_TM	Creation time
CSHOUT_BRKDN_DTL	Top-level hierarchy for cash out breakdown details.
CSHOUT_BRKDN_OUT	Top level hierarchy for cash out tolerance breakdown output elements
CSHOUT_BRKDN_QRY	Top level hierarchy for cash out tolerance breakdown query elements
CSHOUT_TLRNCE	BA's cash out tolerance quantity
CST	Cost
CURR_DMND	The current forecast for the EOD demand (mcm)
CV	Calorific value
DAY_RATE	Day Rate
DEEMED_ALLOC_NDMA_LOAD	Deemed Allocated NDMA Load Quantity for the BA, for the gas day
DLY_CSHOUT_BRKDN_DTL	Top-level hierarchy for daily cash out tolerance breakdown details.
DM_SHR	Sum of all the DMC DMA and Shrinkage Nominations for a BA
DMC_TYPE	The sub type for a Daily Consumer (DC) meter
DTL	Detail (a top level hierarchy for header fields)
EFF_END_DT	Effective end date
EFF_ST_DT	Effective start date
END_ACTVTY_NBR	End activity number
ENRG_DTL	Top level hierarchy for energy detail
ENRGY_TOTLS	Top-level hierarchy for the energy totals
EXER_PRC	Exercise Price
FCAST_DEVTN	BA's forecast deviation for a gas day
FIRM_METER_NO	The firm meter number for the storage output meter
FLEXBLTY	Sum of all flexibility trades (Buy or Sell) for a BA
FLOW_TYPE	Indicates if the Flow Type is firm or interruptible
FRST_NM	First nomination
GAS_DAY	Gas day
GAS_TRD_BRKDN_OUT	Top level hierarchy for Gas Trades Breakdown output elements
GAS_TRD_BRKDN_QRY	Top level hierarchy for Gas Trades Breakdown query elements
GAS_TRD_DTL	Top-level hierarchy for Gas Trade details
HDR_DTL	Top-level Header details
HR_BR	Hour bar

HSTRY_DTL	Top level hierarchy for the History Detail Elements
I_O_IND	Input Output Indicator (a flag to indicate flow direction)
IGNR_TLRNCE	Ignore Tolerance
INS_NM_DTL	Top-level hierarchy for INS Nomination Headers.
INS_NOM	The value of the INS Nomination
INTR_METER_NO	The interruptible meter name for the storage output meter.
IP_BAL	The total amount of energy requested to be flowed into the NBP
LCN	Meter ID
LCTN	Location
LCTN_CD	Location (Zone) Identifier
LCTN_DESC	Location (Zone) Description
LCTNS	Locations
LDZ	The LDZ Id
LDZ_NM	The LDZ name
LNPK_CMNT	Additional details related to the Status Information are entered here
MATCH_YN	Matching trades flag Matched- Yes Unmatched- No
MAX_ACPTD_PRC	Maximum Accepted Price
METADATA	Top level hierarchy for meta information elements
METER_ID	Meter Id for which storage output claims are retrieved
METER_TYPE	Specifies Type of meter
MIN_ACPTD_PRC	Minimum Accepted Price
MIN_QTY	Minimum Quantity
MKT_OP_CD	Market Operator for whom Trade Breakdown details are retrieved.
MOS	Method Of Sale
MSG	The Message to be shown to BA
MSG_CD	Message Code
MSG_DESC	Message Description
MSRD_QTY	Quantity of energy measured for the specified Meter ID/Gas day.
MTR_ID	Meter Id
MTR_NAME	Meter Name
MTR_TYPE	Meter Type
MTR_ZON_DTL_HDR	Meter Zone Relationship Header Details
MTR_ZON_DTLS_INF	Meter Zone Relationship Information Details

NA_RNM_UPDTD	This shows whether the Renomination is created for the NDMA meters of the shipper. If the percentage change of the Total NDMA Forecast is greater than or equal to the set percentage value, then this indicator will be set to .Y..
NDM	The energy details for Non-Daily Meters
NDMA_OP_RNM_FCAST	NDMA output re-nomination forecast for the BA, for the gas day
NET_BAL	The difference between the requested input balance and requested output balance
NET_CURR_ALLOC_QTY	Net Current Allocated Quantity
NET_NMTD_QTY	Net Nominated Quantity
NET_REQ_NRG	The difference between the requested input and the requested output energy
NET_SCHD_NRG	The difference between the scheduled energy bought and sold.
NM_SYS_HSTRY_OUT	Top level hierarchy for output elements
NM_SYS_HSTRY_QRY	Top level hierarchy for query elements
NO_ACPTD_BIDS	Number of Accepted Bids
NO_BID_SHPRS	Number of Bidding Shippers
NO_OF_ACTVTS	Number of activities
NO_OF_STPS	Number of re-nomination steps
NO_SUCS_SHPRS	Number of Successful Shippers
NRG_DTL	Top-level hierarchy for energy details
NRG_FCAST_DTL	Top-level hierarchy for Shipper Total Energy Forecast Headers.
OBO	On behalf of
OCM_NBP_BUYS	Sum of all Title Swap buy trades for a BA
OCM_NBP_SELLS	Sum of all Title Swap sell trades for a BA
OCM_PHY	Sum of all OCM Physical Nominations (Buy or Sell) for a BA
OP_BAL	The total amount of energy requested to be flowed out of the NBP
OPN_LNPK	The line pack at the start of the gas day (mcm)
OPP_FL	Opposite flow indicator
OPP_FL	Opposite flow indicator
PCLP1	The Projected Closing Line pack 1 (mcm)
PCLP2	The Projected Closing Line pack 2 (mcm)
PCTG	Tolerance percentage set for Each Meter Type.
PR_INFO_HSTRY_DTL	Top level hierarchy for the Individual record elements
PR_INFO_HSTRY_OUT	Top level hierarchy for output elements

PR_INFO_HSTRY_QRY	Top level hierarchy for query elements
PRC	The price at which the energy was sold/purchased
PRCNT_CHNG	Percentage change
PRD_FRM	Period From
PRD_TO	Period To
PRDT	Product
PRM_PRC	Premium Price
PROJ_EOD_IMBAL	The projected day imbalance for INS Nomination
PRTY_CD	Party Code of the BA
QTY_BGT	Quantity Bought Back
QTY_NOT_BGT	Quantity Not Bought Back
QTY_SLD	Sold Quantity
QTY_UNSLD	Unsold Quantity
REG_DTL	Top level hierarchy for set of registered records.
REG_PHY_LOC_DTL	Top level hierarchy for individual elements
REG_PHY_LOC_OUT	Top level hierarchy for output elements
REG_PHY_LOC_TRD	Top level hierarchy for individual input records
REG_PHYS_LOC_TRDS	Top level hierarchy for Register Physical/Locational Trade elements
REG_PHYS_LOC_TRDS_DTL	Top level hierarchy for individual input records
REG_TTL_SWP	Top level hierarchy for individual input records
REG_TTL_SWP_DTL	Top level hierarchy for individual elements
REG_TTL_SWP_OUT	Top level hierarchy for output elements
REG_TTL_SWP_RGSTR	Top level hierarchy for Register Title Swap elements
RENOM_DTL_INFO	The detail record for re-nomination
REQ_INP_NRG	The requested input energy value at the specified hour bar.
REQ_NRG	The amount of energy nominated by the shipper
REQ_NRG_DTL	Top-level hierarchy for requested energy details.
REQ_NRG_TOT	The sum of the requested energies
REQ_OP_NRG	The requested output energy value at the specified hour bar
REQ_STS	Requested status
RNM_DTL	Top-level hierarchy for re-nominations details
RNM_DTL_INF	The detail record for Re-nomination
RNM_DTLS_INF	The detail record for the re-nomination
RNM_HDR	Top-level hierarchy for re-nominations headers.
RNM_INF	The detail record for re-nomination info

RSN	Reason for modification of a record
RSN_CD	Reason for rejecting the Bid.
RSN_FOR_RJCTN	Reason For Rejection/pro-rating the capacity
RT_SCHD	Type of BA service
RVNU	Revenue
SAP	System Average Price (p/kWh)
SCHD_INP_NRG	The scheduled input energy value at the specified hour bar.
SCHD_NRG	Energy approved by the O&T against the nominated value.
SCHD_NRG_BGT	The approved energy bought.
SCHD_NRG_DTL	Top-level hierarchy for scheduled energy details.
SCHD_NRG_SELL_STS	The Status of the approved energy sold.
SCHD_NRG_SMRY	Top-level hierarchy for scheduled energy Summary.
SCHD_NRG_SOLD	The approved energy sold.
SCHD_NRG_SUMM	Top-level hierarchy for Scheduled Energy Summary.
SCHD_NRG_TOT	The sum of the scheduled energies
SCHD_OP_NRG	The scheduled output energy value at the specified hour bar
SCHD_STS	Scheduled status
SCLNG_FCTR	Scaling factor
SELL_DTL	Top-level hierarchy for Sell details.
SERVC_ID	Unique identifier for a Shipper's service
SF	Special function
SHCD_NRG_BUY_STS	The status of the approved energy bought
SHPR	The BA's Abbreviated Name.
SHPR_EOD_NMS_OUT	Top level hierarchy for output elements
SHPR_EOD_NMS_QRY	Top level hierarchy for query elements
SHPR_PRLIM_BAL_OUT	Top level hierarchy for query elements
SHPR_PRLIM_BAL_QRY	Top level hierarchy for meta information elements
SHPR_TOT_NRG_FCAST_OUT	Top level hierarchy for output elements
SHPR_TOT_NRG_FCAST_QRY	Top level hierarchy for query elements
SHPR_TRD_DTL	Top-level hierarchy for Shipper Trade details.
SHPR_TRD_DTL_OUT	Top level hierarchy for Shipper Trade details Output elements
SHPR_TRD_DTL_QRY	Top level hierarchy for Shipper Trade details query elements
SMP_BUY	System Marginal Price Buy (p/kWh)

SMP_SELL	System Marginal Price Sell (p/kWh)
STEP	Top level hierarchy for step elements
STEP_NRG	Step energy
STR_TIME	Start time
STRG_IP	Sum of all input nominations at storage meters for a BA
STRG_OP	Sum of all output nominations at storage meters for a BA
STRG_OP_CLMS_OUT	Top level hierarchy for Storage Output Claims output elements
STRG_OP_CLMS_QRY	Top level hierarchy for Storage Output Claims query elements
STRG_OP_DTL	Top-level hierarchy for Storage output details.
STRT_ACTVTY_NBR	Start activity number
STRT_TM	Start time
STS	Status (Accepted/Rejected) of the Trade
STS_OUT	Top level hierarchy for output elements
SUBTNDR_ID	Sub Tender ID
SYS_BAL_NOM_OUT	Top level hierarchy for System Nomination Balance Output elements
SYS_BAL_NOM_QRY	Top level hierarchy for System Nomination Balance query elements
SYS_STATS_HSTRY_DTL	Top level hierarchy for the Individual record elements
SYS_STS_INFO	Top level hierarchy for the record elements
TLRNCE_FLG	Flag to indicate as to whether the tolerance check needs to be carried out or ignored
TLRNCE_QTY	Tolerance quantity for the meter type
TMSTMP	Timestamp
TMSTMP_MAX_PRC	Timestamp of allocation of bid with the maximum accepted price
TMSTMP_MIN_PRC	Timestamp of allocation of bid with the minimum accepted price
TOT	Total
TOT_ALLOC_CPCTY	Total Allocated Capacity
TOT_ALLOC_QTY	Indicates the total net current allocated quantity.
TOT_BID_CPCTY	Total Demanded Bid Capacity
TOT_CLM_QTY	Indicates the total claimed quantity.
TOT_FCAST	Shipper's Total NDMA forecast across all the LDZs for the selected gas day
TOT_IP	Total input quantity
TOT_MIN_QTY	Total Minimum Quantity
TOT_MSMT_ALLOC	Total allocated/measured quantity for the meter type

TOT_OP	Total output quantity
TOT_SCHD_NRG_BGT	The sum of all the approved energy bought by all Shippers.
TOT_SCHD_NRG_NET	The sum of net energy of each Shipper.
TOT_SCHD_NRG_SOLD	The sum of all the approved energy sold by all Shippers.
TRD_BRKDN_DTL	Top-level hierarchy for Gas Trade Breakdown details
TRD_BUY	Sum of all OTC NBP buy trades for a BA
TRD_PTNR	The Shipper Abbreviated Name involved in the Trade
TRD_QTY	Total energy bought/sold by the BA
TRD_SELL	Sum of all OTC NBP sell trades for a BA
TRD_SUMM_DTL	Top-level hierarchy for Trade Summary details
TRD_TYP	Trade Type
TRLR_DTL	Top-level hierarchy for Trailer details
TRNS_IP	Sum of all nominations at input meters (other than storage input meters) for a BA
TX_END_DT	Transaction End Date
TX_STRT_DT	Transaction Start Date
UPDT_DT	Updated date
UPDT_INS_NMS	Top level hierarchy for update elements
UPDT_INS_NMS_DTLS	Top level hierarchy for update detail elements
UPDT_INS_NMS_OUT	Top level hierarchy for update elements
UPDT_RNM	Top level hierarchy for the Update Re-nominations API
UPDT_RNM_DTL	Top level hierarchy for the details of update Re-nominations
UPDT_RNM_OUT	Top level hierarchy for update elements
UPDT_TM	The latest time when the System Status details were updated.(hh:mm:ss)
VOL	Volume
VOL_BGT_MAX_PRC	Volume Bought at Maximum Price
VOL_BGT_MIN_PRC	Volume Bought at Minimum Price
VOL_REQ_MAX_PRC	Volume Requested at Maximum Price
VOL_REQ_MIN_PRC	Volume Requested at Minimum Price
VOL_SLD_MAX_PRC	Volume Sold at Maximum Price
VOL_SLD_MIN_PRC	Volume Sold at Minimum Price
VW_INS_NMS_OUT	Top level hierarchy for INS Nomination elements
VW_INS_NMS_QRY	Top level hierarchy for INS Nomination query elements
VW_RNM_DTLS_OUT	Top level hierarchy for output elements
VW_RNM_DTLS_QRY	Top level hierarchy for query elements
VW_RNM_OUT	Top level hierarchy for output elements

VW_RNM_QRY	Top level hierarchy for query elements
VW_WCF_SF_VAL_DTL	Top level hierarchy for the Individual record elements
VW_WCF_SF_VAL_OUT	Top level hierarchy for output elements
VW_WCF_SF_VAL_QRY	Top level hierarchy for View WCF/SCF Values elements
WAP	Weighted Average Price
WAP_TOP50_ALOC_QTY	WAP of Top 50% of Allocated Quantity
WCF	Weather correction factor

7.3. NTS Exit Capacity APIs

The following table describes the NTS Exit Capacity APIs.

Table 23: NTS Exit Capacity APIs

Shortened Element Name	Element Description
BA_CODE	Business Associate
ALLOC_QTY	Allocated Quantity
ENTITLEMENT_SWAP	Entitlement Swap
FLOW_ST_TIME	Flow Start Time
GAS_DAY	Gas Day
LCTN	Location
LCTNS	Locations
MAX_ACC_PRC	Max Accepted Price
MAX_QTY	Maximum Requested Capacity
MIN_ACC_PRC	Min Accepted Price
MIN_QTY	Minimum Requested Capacity
MOS	Method of Sale
NET_CAPACITY	Net Capacity
NET_FIRM	Net Firm
NET_SCALED_OFFPEAK	Net Scaled Off-peak
ORIGINAL_OFFPEAK	Original Off-peak
PERIOD_FROM	Period From
PERIOD_TO	Period To
PRODUCT	Product
PRODUCT_TYPE	Product Type
QTY_ALL_MAX_PRC	Quantity Allocated at Max Price
QTY_ALL_MIN_PRC	Quantity Allocated at Min Price
QTY_BOUGHT_BACK	Quantity Bought Back

QTY_OFFERED	Quantity Offered
QTY_REQUESTED	Quantity Requested
QTY_SOLD	Quantity Sold
QTY_UNSOLD	Quantity Unsold
REASON	Reason
REQ_ID	Request ID
REQ_PRICE	Request Price
REQ_STATUS	Request Status
REQ_TIMESTAMP	Timestamp
REQ_TRANS_FLAG	Request Transfer Flag
REQ_TYPE	Request Type
REQ_VALUE	Request Value
STATUS	Status
SUB_TX_PERIOD	Sub Transaction Period
TMSTMP	Timestamp
TX_PERIOD	Transaction Period

7.4. IP Location API

8. Debugging for API Client Developers

This section contains various miscellaneous hints and tips for API client developers.

- If you are debugging a login API client then you may make several unsuccessful attempts to login. It is useful to intersperse these attempts with a login to the Gemini / NTS Exit Reform screen service via Citrix. This achieves two things:

  1. It confirms that your login credentials are correct and any failure to login via the login API is down to something else.
  2. It resets your consecutive unsuccessful login attempts count for that user id. This is important to prevent the account being locked.

- Be aware that when you login to the Gemini / NTS Exit Reform screens via an API user id you may see an error to the effect that the application has failed to build a menu. This can be ignored for user ids with API access only. It indicates that no screens are allocated to the user id.
- During API client development use the XML Schema Definitions (XSDs) on the Gemini / NTS Exit Reform servers to validate your request XML before sending it. There are various third party tools that will validate XML against an XSD.
- Avoid physically storing the request/response XML unless you have to. Input/output operations are resource expensive.
- Document Object Model (DOM) parsers are more memory intensive for read operations. If you have the choice, use Simple API for XML (SAX) parsers for read and DOM for generate/update. SAX is much quicker than DOM for read operations.
- The error XML contains the error code and associated error messages. If you would rather display a different message you can map our error codes to your own specific messages.
- Remember that API sessions will time expire after sixty minutes of inactivity. If your client is inactive for this period, you will have to login again to re-establish a session.
- Make the URLs that you access configurable. Do not hard code them within your API clients.

  This is especially important for the root URL of the API service. Changes to domain naming policies, for example, can affect these URLs. For this reason, we specify URLs in all documentation relative to the root URL of the API service, designated by a single "/".

  It is useful to separately hold the root URL of the API service that you are accessing, which will be common to all API calls, and the relative URLs, which will be specific to each individual API.

- API clients must not be multi-threaded.
- To diagnose errors try to trap your raw HTTP responses and requests. APIs are intended to be client language independent and the requirements for API client interaction with the API are deliberately specified terms of HTTP and associated XML structures.

  If you are investigating a suspected API error then it will greatly assist both yourselves and us to see the interaction in terms of HTTP messages. What happens in various client languages to send/receive those messages might be very different.

- If you schedule changes of password via the change password API before your password expires, then you avoid the complexity of having to react to a password expiry warning or actual expiration. Current passwords will expire after thirty days of use. You are warned five days in advance.

9. Document Control

9.1. Superseded Documents

Title	Version	Reference	Date
API_Usage_Guidelines_v3_20051013.pdf	3.0	http://www.xoserve.com/Gemini/Technical_Documentation/Gemini_API_Bulk_Uploads_Downloads/API_Usage_Guidelines_v3_20051013.pdf	13-October-2005

9.2. Version History

Version	Status	Date	Author(s)	Summary of Changes
0.1	Draft	29-Jul-2011	Elaine Hall, Xoserve	Initial draft
0.2	Draft	11-Aug-2011	Elaine Hall, Xoserve	Updated following initial review
0.3	Draft	19-Aug-2011	Elaine Hall, Xoserve	Updated following second review
0.4	Draft	06-Sep-2011	Elaine Hall, Xoserve	Updated to re-incorporate Gemini APIs, following comments from IS Ops
0.5	Draft	15-Sep-2011	Elaine Hall, Xoserve	Updated, following review by Wipro
0.6	For Representation	14-Oct-2011	Elaine Hall, Xoserve	Date and Version No. only
0.7	For Representation	09-Dec-2011	Elaine Hall, Xoserve	Section 4.4 Generic Error codes: (i) Reference to Error_0004 corrected to Error_0007 for Gemini; (ii) Reference to this code for Gemini Exit has been removed, following confirmation from IS Ops that it is not required for Gemini Exit.
1.0	Approved	12-Jan-2012	Elaine Hall, Xoserve	Changes to document control only, following approval at UK Link Committee.
1	For Representation	13-Sep-2011	Dhawal Singh, TCS	Amended diagram in section 1.1 to show Xoserve data centre
1.1	For Representation	10-Dec-2012	Dhawal Singh, TCS	Section referencing corrected; SSSLv2 reference removed from section 2.1
2.0	Approved	11-Dec-2012	Dhawal Singh, TCS	Approved
2.1	For Representation	05-Jul-2013	GRP Project Team	Document proof read for accuracy post GRP (minor amendments made)
3.0	Approved	3-Sept-2013	GRP Project Team	Document published on website post GRP
3.1	For Review	23-Nov-2016	Gemini Application Support	Modified as part of API Document update

4.0	Approved	27-03-2017	Gemini Application Support	Document Restructured