Gomez Data Feed API

DEVELOPER GUIDE

Version 4.0

March 2010

The Web Performance Division of Compuware

Gomez, a division of Compuware Corporation • 10 Maguire Road • Lexington, MA 02420 • USA

Table of Contents

Chapter 1 Introduction to DataFeeds 4.0 1
Chapter 2 DataFeed Options 3
Chapter 3 DataFeed Web Services 8
Chapter 4 FTP DataFeeds 32

Chapter 1
Introduction to Data Feeds 4.0

The purpose of this document is to describe the details and use of the Application Programming Interface (API) that is used by Gomez partners and customers to perform the retrieval of collected data sets from Gomez.

Gomez will provide the raw data and associated data definitions so the partners and customers can consume Gomez performance data in a meaningful way and decide which data elements to request and how to integrate and exploit the data in their own portals and applications.

New Features and Functionality in V4.0

The latest version of the GPN Data Service API now supports a broader range of available Gomez Active data sets including:

- Backbone Streaming Data Sets
- Last Mile Data Sets
- Private Peer Data Sets
- Mobile Data Sets and Data Extensions

Support has been added for discrimination between legacy UTA test measurements and newer BROWSER based test measurements through the inclusion of a browser type attribute which is associated at the test level of every acquired data sample.

Support has been added to emit an extended data set for MOBILE tests.

A new request mode has been added for eliciting data sets which correspond to samples which have been reclassified as "data cuts" in the system. Data cut requests are supported across all available response models.

Data cut information can be elicited for any series of cut events *occurring within the last 4 days.*

A new request mode has been added to cause the data service to reinterpret the requested time range as a *platform arrival* time rather than the current interpretation as *test acquisition* time. This allows clients greater flexibility for use cases which must take into account latencies within the Gomez testing platform.

A companion schema document for the response payload is offered.

> NOTE: Responses from the 4.0 version of the service are not backward compatible with the previous version of the service. While many of the low level element attributes are preserved, many of the containing element names have changed. Specifically the following elements names have changed to better support typed schema definition:

The legacy element name "TEST" has now been specialized to better inform the response model being used:

- TXTEST – Indicates a Transaction Test Response element emanating from Backbone, Last Mile, Private Peer, or Mobile Systems
- TXTIMEGROUP – Indicates a Transaction Test Time Group Response element emanating from Backbone-based transactions.
- STREAMTEST – Indicates a Stream Test Response element emanating from Backbone Streaming System
- DATACUT – Indicates a Data Cut Sample element emanating from Backbone, Last Mile, Private Peer or Mobile Systems
- MESSAGE – Indicates a general purposes messaging element

Chapter 2
Data Feed Options

The Gomez monitoring tools provide a variety of graphics displays that satisfy the reporting and analytical needs of most Gomez customers. For those customers that want to correlate internal business data with Gomez data, Gomez provides several data feeds which customers can use to download their Gomez data.

Data feeds provide another ways for Gomez customers to access and structure their data. It can be tricky determining which feed is best for specific requirements. This document first discusses the two Gomez data feeds (FTP and Web Services) and then provides a number of recommendations to help you determine which data feed is best for your purposes.

This chapter covers the following topics:

- Which data feeds are available?
- Which data feed do I need?
- How do I get the data feed I want?
- Does the data feed provide the data I need?

Which Data Feeds are Available?

Gomez provides the following data feeds:

- Web Services
- FTP

For a large volume (hundreds) of tests, downloads can approach over one gigabyte (1,000 MB) of data in a day, and the File Transfer Protocol (FTP) data feed provides the most efficient way to download this quantity of data from Gomez. Most clients, however, download the results of only dozens of tests on average. For these clients, the Gomez Web Services data feed is the best way to download test data.

Web Services Data Feed

The Web Services data feed is the best way for obtaining most data from Gomez.

Customers can create batch programs to download test-level, page-level, and object-level data and to load that data into data warehouses to supplement business analytics and reporting. If you need support for creating programs to download data, contact Gomez Professional Services.

Contact Gomez Client Services to have your account enabled for Gomez Web Service data feeds.

FTP Data Feed

Use the FTP data feed to obtain large amounts of data at once from the Gomez platform. For users that have dozens of tests and large test volume (for example, service providers and ad networks), FTP is the preferred method for application integration at the customer premises. Many tests can generate up to a gigabyte of data, and files are created every hour by scheduled scripts. As with Web Services, this amount of data requires planning and management by technical personnel.

Contact Gomez Client Services to set up your FTP account.

Emailed Reports and Alerts

While these are not really data feeds, for many users it is critically important to receive an emailed report that is already aggregated, processed, and summarized. Email reports are sent directly to the customer and can be set up from the Saved Charts, Reports, or Alerts pages of the Gomez portal. Similarly, emailed alerts do not require any processing by the user. Their use is described in the online help.

If you need a custom or enhanced report, contact your Account Manager.

XML Data Feed

For users interested in programming or integrating small subsets of certain Last Mile data via an HTTPS (or HTTP) connection, the XML data feed can be used.

NOTE: The XML Data Feed is targeted for deprecation, to be replaced by the Web Services Data Feed. At this time, most customers would already be better served by the Web Services Data Feed.

Which Data Feed do I Need?

The choice depends on a number of factors, but the key factors are the total number of tests and the target application. You may wish to receive an aggregated weekly summary report of a dozen tests in Microsoft Excel format and Email reports are perfect for this. On the other hand, if you have a dedicated analytics team and a desire to correlate test measurements with a web user analytics system, a daily batch pull using a Gomez Web Service data feed is the best option.

Audience	Function	Number of Tests	Target Application or Format	Gomez Data Feed Recommendation
Business	SLA reports	Under 10	Excel, PDF, HTML, CSV	Email Report
Business	Aggregated	Any	Excel, PDF, HTML, CSV	Email Report
Business	Daily Test Results	Under 20	Excel, PDF, HTML, CSV	Email Report
Technical	Aggregated for Smart Phones	Any	HTML	Email Report
Technical	Daily test results	Any	Analytics or Dashboards	Web Services
Technical	Daily Batch Load	Any	Data Warehouse	Web Services
Technical	Hourly Batch Load	Over 20	Data Warehouse	FTP or Web Services
Any	Receive Alerts	Any	Email	Alert Email

How Do I Get the Data Feed I Need?

Once you have determined which data feed you are interested in, there are a number of different processes you need to follow, depending on the feed you want:

Data Feed	How to get it
Web Services	Have your Gomez account activated first by Client Services to access this feed.
FTP	Open a ticket with Client Services to get your FTP account set up.

Does the data feed provide the data I need?

The level of detail you need may determine which data feed to use since not all data feeds include all levels of detail. For instance, the Last Mile web service includes most, but not all, types of Backbone data. Use the following matrix to determine which feed is available for you to use.

Data Source	Data Type	Web Services	FTP
Backbone	Summary	✓	
	Page	✓	✓
	Object	✓	
	All	✓	
	Datacut	✓	
Streaming (Backbone)	Summary	✓	
	Page	✓	✓
	Object	✓	
	All	✓	
	Datacut	✓	
Last Mile	Summary	✓	
	Page	✓	✓
	Object	✓	
	All	✓	
	Datacut	✓	
Private Peer	Summary	✓	
	Page	✓	✓*
	Object	✓	
	All	✓	
	Datacut	✓	
Mobile	Summary	✓	
	Page	✓	✓*
	Object	✓	
	All	✓	
	Datacut	✓	

Data Source	Data Type	Web Services	FTP
Actual	Raw	✓	
	Chart	✓	
	Geo	✓	

* No Private Peer extended data attributes

The following table provides more detail on the data feed chart shown above:

Data Type	Description
Summary	Raw data for each test specified over the time frame requested.
Page	Raw data for each test for each page in the test over the time frame requested. The hourly page summary rolls up the data by test id, hour, node, and page.
Object	Every object in the test is provided, ordered by test id, timestamp, node, page sequence, and object sequence. NOTE: You cannot aggregate up to test results due to problems around HTTP concurrent connections, compression, and other technical issues.
All	Includes Summary, Page, Object, plus Host and Connection data.
All (for Streaming only)	Includes Summary plus Event, Trace, and Meta data.
Data Cut	Specific data sets for cut events occurring in the last 4 days.
Raw	Provides raw Actual data
Chart	Provides aggregated Actual data.
Geo	Provides geographically-based Actual data.

Chapter 3
Data Feed Web Services

Data Export Services

The following functions will be supported by the Data Feed Web Services interface:

Request	Results
OpenDataFeed	Initializes client data feed session based on a set of input parameters. A token in returned to the client for subsequent data pickup.
OpenDataFeed2	Initializes client data feed session based on a set of input parameters. A token in returned to the client for subsequent data pickup. This interface supports the specification of a mode of operation using relative rather than absolute time windows.
OpenDataFeed3	Initializes client data feed session based on a set of input parameters. A token in returned to the client for subsequent data pickup. This interface supports the specification of a mode of operation using ARRIVAL time rather than TEST time windows.
GetResponseData	Get next chunk of data for current feed session.
CloseDataFeed	Explicitly expires current feed session
GetErrorCodes	Exports raw error code definitions

A session is begun with one of the three supported OpenDataFeed requests. After a successful response completes the handshake, a GetResponseData request starts the actual data transfer. A successful GetResponseData response includes an XML file containing the data payload. A CloseDataFeed request starts the termination of the session after the desired data is transferred. With a successful CloseDataFeed response the session is completed. The following diagram illustrates how a typical session proceeds.

Data Services API

The following section outlines specific interfaces to the Data Services API.

OpenDataFeed Session

The *OpenDataFeed* web service method initializes a Gomez data feed session based on a number of user-defined criteria. A data feed session always exports a homogeneous dataset based on the MonitorClassDesignator and DataDesignator specified when the session is initialized.

OpenDataFeed Request Message Details

The request message includes parameters and validation criteria, as described in the following tables.

Parameters	Description
Username	GPN account username.
Password	GPN account password
MonitorIdSet	Requested set of GPN Monitor/Test reference Identifiers. This set will be cross referenced against the “MonitorClassDesignator” and only requested monitors meeting the specified class criteria will be returned. NOTE: The set of GPN monitor identifiers available to an account may be elicited via use of the “GPN Account Management Service”
SiteIdSet	Requested set of GPN Site reference Identifiers. Empty set denotes all sites. NOTE: The set of GPN site identifiers available to an account may be elicited via use of the “GPN Account Management Service”

Parameters	Description
MonitorClassDesignator	Specifies the type of data to be exported in the dataset based on the common class of tests. Supported values are as follows: UTATX — Data sets based on transaction tests running on UTA agents BROWSERTX — Data sets based on transaction tests running on browser agents. Note that this is a shared schema with UTA. STREAM — Data sets based on streaming test running on stream agents. PRIVATEPEER — Data sets based on tests running on Private Peer Network LASTMILE — Data sets based on tests running on Last Mile Network MOBILE — Data sets based on tests running on Mobile Network
DataDesignator	Specifies the levels of data in the dataset returned to the client for the type of data specified with the MonitorClassDesignator (see the preceding parameter description). Supported values are as follows: ALL — Returns all levels of data available. OBJECT — Returns SUMMARY, PAGE, and OBJECT level data. Mobile datasets will also include MOBILEEXT level data. PAGE — Returns SUMMARY and PAGE level data. Mobile datasets will also include MOBILEEXT level data. SUMMARY — Returns SUMMARY level data. Mobile datasets will also include MOBILEEXT level data. TIMEGROUP — Returns aggregation data over specified time window, up to 24 hours. (see following parameter descriptions).
StartTime	Specifies the starting time boundary for the dataset. The format is *YYYY-MM-DD HH:MM:SS*
EndTime	Specifies the ending time boundary for the dataset. The format is *YYYY-MM-DD HH:MM:SS*
OrderDesignator	Specifies how the dataset will be ordered. Supported values are as follows: TIME — Orders data by time. SITE — Orders data by site. TEST — Orders data by monitor class

Together, the DataDesignator and MonitorClassDesignator values define what will be included in the retrieved dataset. The supported combinations are listed in the following table.

DataDesignator	MonitorClassDesignator	DataSet Retrieved
ALL	UTATX	Summary, page, object, host, and connection-level.
	BROWSERTX	Summary, page, object, host, and connection-level.
	STREAM	Summary, event, metadata, and trace-level.
	PRIVATE PEER	Summary, page, object, host, and connection-level.
	LASTMILE	Summary, page, object, host, and connection-level.
	MOBILE	Summary, page, object, host, connection, and mobileext-level.
OBJECT	UTATX	Summary, page, and object level only.
	BROWSERTX	Summary, page, and object level only.
	PRIVATE PEER	Summary, page, and object level only.
	LASTMILE	Summary, page, and object level only.
	MOBILE	Summary, page, object, and mobileext-level.
PAGE	UTATX	Summary and page-level only.
	BROWSERTX	Summary and page-level only.
	PRIVATE PEER	Summary and page-level only.
	LASTMILE	Summary and page-level only.
	MOBILE	Summary, page, and mobileext-level.
SUMMARY	UTATX	Summary-level only.
	BROWSERTX	Summary-level only.
	STREAM	Summary-level only.
	PRIVATE PEER	Summary-level only.
	LASTMILE	Summary-level only.
	MOBILE	Summary-level only.
TIMEGROUP	UTATX	Up to 24 hours of aggregated data.
	BROWSERTX	Up to 24 hours of aggregated data.

In the following table are descriptions of the validation criteria for the OpenDataFeed request.

Validation Criteria	Description
Username	The supplied username is valid for the active GPN account.
Password	The supplied password is valid for the active GPN account.
MonitorIdSet	The set of Monitor IDs is valid for the active GPN account.
SiteIdSet	The set of Site IDs is valid for the active GPN account.
StartTime	The time window start boundary is valid.
EndTime	The time window end boundary is valid.
MonitorClassDesignator	The MonitorClassDesignator value is valid.
DataDesignator	The DataDesignator value is valid.
OrderDesignator	The OrderDesignator value is valid.
Data Volume	The volume of data does not exceed account constraints
Frequency	The request frequency does not exceed account constraints
Concurrency	The request concurrency does not exceed account constraints

OpenDataFeed Response Message

The *OpenDataFeed* web service method returns an xml message containing the following:

- Status of the operation
- Set of global operational constraints
- Set of account specific operational constraints
- Session Identifier
- Set of validated monitor identifiers
- Set of validated site identifiers

Sample of a Successful Response Message

```
<?xml version="1.0" encoding="utf-8" ?>
- <GpnOpenUtaDataFeedResponse xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://gomeznetworks.com/webservices/">
- <Status>
  <eStatus>STATUS_SUCCESS</eStatus>
  <sErrorMessage />
  </Status>
  <GPNGlobalConstraints MaxDataRequestsPerHourPerSession="60" MaxTotalSessions="50" />
  <GPNAccountConstraints MaxDataRecordsPerRequest="1000" MaxConcurrentSessions="1"
MaxConcurrentRequests="1" MaxRequestIntervalBeforeExpirationInMinutes="15" />
  <SessionToken>c59754b7-3ed3-4878-a0e8-c352daa0b8a1</SessionToken>
- <AcceptedMonitors>
  <mid>1162690</mid>
  </AcceptedMonitors>
- <AcceptedSites>
  <sid>101</sid>
  </AcceptedSites>
  </GpnOpenUtaDataFeedResponse>
```

Sample of an Unsuccessful Response Message

```
<?xml version="1.0" encoding="utf-8" ?>
- <GpnOpenUtaDataFeedResponse xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://gomeznetworks.com/webservices/">
- <Status>
  <eStatus>STATUS_FAILED</eStatus>
  <sErrorMessage>INVALID ORDER DESIGNATOR SPECIFIED. MUST BE (TIME | MONITOR |
SITE)</sErrorMessage>
  </Status>
  </GpnOpenUtaDataFeedResponse>
```

OpenDataFeed Response Message Details

The following table summarizes the elements and attributes in the XML response returned by this web service. Refer to the WSDL for the complete definition of this XML response.

Container Element	Data Element/Attribute	Description
OpenDataFeedResponse		
Status	eStatus	Contains STATUS_SUCCESS or STATUS_FAILED
	sErrorMessage	Contains an error message if STATUS_FAILED
GPNGlobalConstraints	MaxDataRequestsPerHourPerSession	Maximum number of data requests that can be made per hour within a given session is 60
	MaxTotalSessions	Maximum number of sessions which can handled by the service is 32
GPNAccountConstraints	MaxDataRecordsPerRequest	Maximum number of data records which may be obtained per GetResponseData request is 1000
	MaxConcurrentSessions	Maximum number of concurrent sessions which will be serviced is 1
	MaxConcurrentRequests	Maximum number of concurrent requests which can be made within the context of a given session is 1
	MaxRequestIntervalBeforeExpirationInMinutes	Maximum amount of time which can elapse between calls to "GetResponseData" before a session may be expired by the system is 15 minutes
	SessionToken	Session Identifier which is used in subsequent calls to GetResponseData and CloseDataFeed
	CurrentTime	Contains the time the request was received in GMT
	AcceptedMonitors	Contains a list of GPN monitor identifiers which have been validated by the system against a specified account
	AcceptedSites	Contains a list of GPN site identifiers which have been validated by the system against a specified account

OpenDataFeed2 Request Message Details (Alternate Version – Last N Samples Relative to Current Time)

Web method for initializing GPN UTA data feed session based on a number of user-defined criteria. A data feed session always exports a homogeneous dataset based on the specified MonitorClassDesignator and the DataDesignator tags provides when the session is initialized.
The request message includes parameters and validation criteria, as described in the following tables.

Parameters	Description
Username	GPN account username.
Password	GPN account password
MonitorIdSet	Requested set of GPN Monitor/Test reference Identifiers. This set will be cross referenced against the "MonitorClassDesignator" and only requested monitors meeting the specified class criteria will be returned. NOTE: The set of GPN monitor identifiers available to an account may be elicited via use of the "GPN Account Management Service"
SiteIdSet	Requested set of GPN Site reference Identifiers. Empty set denotes all sites. NOTE: The set of GPN site identifiers available to an account may be elicited via use of the "GPN Account Management Service"
MonitorClassDesignator	Specifies the type of data to be exported in the dataset based on the common class of tests. Supported values are as follows: UTATX – Data sets based on transaction tests running on UTA agents BROWSERTX – Data sets based on transaction tests running on browser agents. Note that this is a shared schema with UTA. STREAM – Data sets based on streaming test running on stream agents. PRIVATEPEER – Data sets based on tests running on Private Peer Network LASTMILE – Data sets based on tests running on Last Mile Network MOBILE – Data sets based on tests running on Mobile Network
DataDesignator	Specifies the levels of data in the dataset returned to the client for the type of data specified with the MonitorClassDesignator (see the preceding parameter description). Supported values are as follows: ALL – Returns all levels of data available. OBJECT – Returns SUMMARY, PAGE, and OBJECT level data. Mobile datasets will also include MOBILEEXT level data. PAGE – Returns SUMMARY and PAGE level data. Mobile datasets will also include MOBILEEXT level data. SUMMARY – Returns SUMMARY level data. Mobile datasets will also include MOBILEEXT level data. TIMEGROUP – Returns aggregation data over specified time window, up to 24 hours. (see following parameter descriptions).
Last N	Indicates a window based on the last N test samples relative to the "current time". A value of N=1 will spec a data export session which always returns the last test acquired relative to the current time. This parameter can contain any value up to cap enforced by the "MaxSamplesPerRequest" constraint. Any values specified for this parameter which exceed this constraint will be set to the constraint. Ignored when DataDesignator is "TIMEGROUP"

Parameters	Description
StartTime	Specifies the starting time boundary for the dataset. The format is *YYYY-MM-DD HH:MM:SS*
EndTime	Specifies the ending time boundary for the dataset. The format is *YYYY-MM-DD HH:MM:SS*
OrderDesignator	Specifies how the dataset will be ordered. Supported values are as follows: TIME Orders data by time. SITE Orders data by site. TEST Orders data by monitor class

Validation Criteria	Description
Username	The supplied username is valid for the active GPN account.
Password	The supplied password is valid for the active GPN account.
MonitorIdSet	The set of Monitor IDs is valid for the active GPN account.
SiteIdSet	The set of Site IDs is valid for the active GPN account.
StartTime	The time window start boundary is valid.
EndTime	The time window end boundary is valid.
MonitorClassDesignator	The MonitorClassDesignator value is valid.
DataDesignator	The DataDesignator value is valid.
OrderDesignator	The OrderDesignator value is valid.
Data Volume	The volume of data does not exceed account constraints
Frequency	The request frequency does not exceed account constraints
Concurrency	The request concurrency does not exceed account constraints

OpenDataFeed3 Request Message Details (Alternate Version – ARRIVAL vs TEST TIME Filter Interpretation)

Web method for initializing GPN UTA data feed session based on a number of user-defined criteria. A data feed session always exports a homogeneous dataset based on the specified MonitorClassDesignator and the DataDesignator tags provides when the session is initialized.

Parameters	Description
Username	GPN account username.
Password	GPN account password
MonitorIdSet	Requested set of GPN Monitor/Test reference Identifiers. This set will be cross referenced against the “MonitorClassDesignator” and only requested monitors meeting the specified class criteria will be returned. NOTE: The set of GPN monitor identifiers available to an account may be elicited via use of the “GPN Account Management Service”

Parameters	Description
SiteIdSet	Requested set of GPN Site reference Identifiers. Empty set denotes all sites. NOTE: The set of GPN site identifiers available to an account may be elicited via use of the "GPN Account Management Service"
MonitorClassDesignator	Specifies the type of data to be exported in the dataset based on the common class of tests. Supported values are as follows: UTATX — Data sets based on transaction tests running on UTA agents BROWSERTX — Data sets based on transaction tests running on browser agents. Note that this is a shared schema with UTA. STREAM — Data sets based on streaming test running on stream agents. PRIVATEPEER — Data sets based on tests running on Private Peer Network LASTMILE — Data sets based on tests running on Last Mile Network MOBILE — Data sets based on tests running on Mobile Network
DataDesignator	Specifies the levels of data in the dataset returned to the client for the type of data specified with the MonitorClassDesignator (see the preceding parameter description). Supported values are as follows: ALL — Returns all levels of data available. OBJECT — Returns SUMMARY, PAGE, and OBJECT level data. Mobile datasets will also include MOBILEEXT level data. PAGE — Returns SUMMARY and PAGE level data. Mobile datasets will also include MOBILEEXT level data. SUMMARY — Returns SUMMARY level data. Mobile datasets will also include MOBILEEXT level data. TIMEGROUP — Returns aggregation data over specified time window, up to 24 hours. (see following parameter descriptions).
Last N	Indicates a window based on the last N test samples relative to the "current time". A value of N=1 will spec a data export session which always returns the last test acquired relative to the current time. This parameter can contain any value up to cap enforced by the "MaxSamplesPerRequest" constraint. Any values specified for this parameter which exceed this constraint will be set to the constraint. Ignored when DataDesignator is "TIMEGROUP"
StartTime	Specifies the starting time boundary for the dataset. The format is *YYYY-MM-DD HH:MM:SS*
EndTime	Specifies the ending time boundary for the dataset. The format is *YYYY-MM-DD HH:MM:SS*
OrderDesignator	Specifies how the dataset will be ordered. Supported values are as follows: TIME — Orders data by time. SITE — Orders data by site. TEST — Orders data by monitor class

Parameters	Description
TimeDesignator	Designation indicating how the time filters and modes are to be applied. Supported values are as follows: TESTTIME_ABSOLUTE Time range filter to be applied relative to the time samples were acquired by remote agents. This is the current default mode of operation for legacy versions of the service. TESTTIME_RELATIVE Time range filter is unused and LastN must be specified. LastN interpreted as last N samples relative to the time samples were acquired by remote agents. ARRIVAL_ABSOLUTE Time range filter to be applied relative to the time samples arrived at the central data repositories. This is a new mode of operation. ARRIVAL_RELATIVE Time range filter is unused and LastN must be specified. LastN interpreted as last N samples relative to the time samples arrived at the central data repositories.

Validation Criteria	Description
Username	The supplied username is valid for the active GPN account.
Password	The supplied password is valid for the active GPN account.
MonitorIdSet	The set of Monitor IDs is valid for the active GPN account.
SiteIdSet	The set of Site IDs is valid for the active GPN account.
StartTime	The time window start boundary is valid.
EndTime	The time window end boundary is valid.
MonitorClassDesignator	The MonitorClassDesignator value is valid.
DataDesignator	The DataDesignator value is valid.
OrderDesignator	The OrderDesignator value is valid.
Time Designator	The TimeDesignator values are valid.
Data Volume	The volume of data does not exceed account constraints
Frequency	The request frequency does not exceed account constraints
Concurrency	The request concurrency does not exceed account constraints

GetResponseData Request Message Details

This web method returns the next chunk of the dataset for a specified transaction from each site. Range, content, and volume of data will be based on constraints setup when the session was opened with one of the OpenDataFeed web methods and the standard system and account level constraints.

Parameters	Description
SessionToken	Unique Identifier issued by the initial OpenDataFeed call for the session.

Validation Criteria

Validation Criteria	Description
Session Token	The Session Token is valid
Data Volume	The volume of data does not exceed account constraints
Frequency	The request frequency does not exceed account constraints
Concurrency	The request concurrency does not exceed account constraints

GetResponseData Response Message

This Web method returns an xml message containing :

- Status of operation
- Number of data records returned
- XML subdocument containing
  - Type of data returned
  - GPN Performance Dataset

GetResponseData Response Message Details

Container Element	Data Element/Attribute	Description
GetResponseData		
Status	eStatus	Contains STATUS_SUCCESS or STATUS_FAILED
	sErrorMessage	Contains an error message if STATUS_FAILED
GpnResponseData	NumRecords	Contains the number of records exported in the current response
	XmlDocument	Contains the performance data subdocument (see XSD)
	MonitorDesignator	Indicates the "type" of performance data exported in the current request. This will reflect the MonitorClassDesignator value specified in the initialization call to the OpenDataFeed service.
	Test	Top level container for test information Refer to XSD for further details

CloseDataFeed Request Details

Web method for closing an active GPN data feed session. Calling this method causes expiration of the Virtual Session Token.

Parameters	Description
SessionToken	Unique Identifier issued by the initial OpenDataFeed call for the session.

CloseDataFeed Response Message

Web method returns an xml message containing :

- Status of operation

CloseDataFeed Response Message Details

Container Element	Data Element/Attribute	Description
CloseDataFeed		
Status	eStatus	Contains STATUS_SUCCESS or STATUS_FAILED
	sErrorMessage	Contains an error message if STATUS_FAILED

Get ErrorCodes Method

Web method for exporting raw error codes and definitions. It returns an xml message containing all GPN Test Response Error Codes and definitions.

GetErrorCodes Request Message Details

Parameters	Description
n/a	

GetErrorCodes Response Message Details

Container Element	Data Element/Attribute	Description
GetErrorCodes		
Status	eStatus	Contains STATUS_SUCCESS or STATUS_FAILED
	sErrorMessage	Contains an error message if STATUS_FAILED
ErrorCodes	ErrorNum	Contains the error code number
	Description	Contains short error description text

Access to the Data Service

Production Url:
http://gpn.webservice.gomez.com/DataExportService40/GpnDataExportService.asmx

Production WSDL:
http://gpn.webservice.gomez.com/DataExportService40/GpnDataExportService.asmx?WSDL

Production XSD:
http://gpn.webservice.gomez.com/DataExportService40/xsd/GpnDataExportService40.xsd

The full specification is also included as Example #4 in this document. Please note that as this product is currently a Beta offering this address is subject to change.

Security

All interactions with GPN Data Services shall require authentication by user name and password or via issued session tokens.

All credentials presented to the service must be successful authenticated by the service as

- Being a valid GPN user in good standing
- Being explicit configured as having access and proper entitlements to the API.

Data Export Web Service 4.0 Response XSD

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

Description : GPN DATA EXPORT WEB SERVICE v4.0

Created     : 11.11.2009

Revision    : 1.1 - 11.11.2009 - Initial
              1.2 - 11.24.2009 - Mobile Extensions Added

*****************************************************
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

 <!--
       ************************************************
       Top Level Service Response
       ************************************************
       -->
 <xs:element name="GpnResponseData" type="CGpnUniversalXmlDocResponse" nillable="true"/>
 <xs:complexType name="CGpnUniversalXmlDocResponse">
  <xs:sequence>
   <!-- Response Status -->
   <xs:element name="Status" type="COpStatusData" minOccurs="0"/>
   <!-- Number of Data Records Contained in the Response -->
   <xs:element name="NumRecords" type="xs:int"/>
   <!-- Server Time Response Generated -->
   <xs:element name="CurrentTime" type="xs:string" minOccurs="0"/>
   <!-- Response Data Payload -->
   <xs:element name="XmlDocument" type="XmlDocumentType" minOccurs="0"/>
  </xs:sequence>
 </xs:complexType>

 <!--
       ************************************************
       Top Level Response Data Payload
       ************************************************
       -->
 <xs:complexType name="XmlDocumentType">
  <xs:sequence minOccurs="0">
   <xs:element name="GpnResponseData" type="GpnResponseDataType"/>
  </xs:sequence>
 </xs:complexType>

 <!--
       ************************************************
       Response Data Payload Type
       ************************************************
       -->
 <xs:complexType name="GpnResponseDataType">
  <xs:sequence>
   <!-- Monitor Designator - Reflects Requested Type -->
   <xs:element name="MonitorDesignator" type="MonitorClassDesignatorType"/>
   <!-- Specific Data Schema Type -->
   <xs:choice minOccurs="0" maxOccurs="unbounded">
    <!-- Transaction Test Data Type -->
    <xs:element name="TXTEST" type="TxTestType"/>
    <!-- Transaction Time Group Data Type -->
    <xs:element name="TXTIMEGROUP" type="TxTimegroupType"/>
    <!-- Stream Test Data Type -->
    <xs:element name="STREAMTEST" type="StreamTestType"/>
    <!-- Single Url Test Data Type-->
    <xs:element name="SINGLEURLTEST" type="SingleUrlTestType"/>
    <!-- Data Cut Data Type -->
    <xs:element name="DATACUT" type="DataCutTestType"/>
    <!-- General Message Data Type -->
    <xs:element name="MESSAGE" type="MessageType" minOccurs="0"/>
   </xs:choice>
  </xs:sequence>
 </xs:complexType>

 <!--
       ************************************************
       Transaction Timegroup Response Type
```

```
       Applies to Backbone Transactions Only
       ************************************************
       -->
<xs:complexType name="TxTimegroupType">
 <xs:attribute name="monitor_id" type="xs:int" use="required"/>
 <xs:attribute name="avail" type="xs:float" use="required"/>
 <xs:attribute name="avgresp" type="xs:float" use="required"/>
 <xs:attribute name="avgDNS" type="xs:float" use="required"/>
 <xs:attribute name="avgConnect" type="xs:float" use="required"/>
 <xs:attribute name="avgFirstByte" type="xs:float" use="required"/>
 <xs:attribute name="avgContent" type="xs:float" use="required"/>
</xs:complexType>

<!--
       ************************************
       Transaction Response Type
       ************************************
       -->
<xs:complexType name="TxTestType">
 <xs:sequence>
  <!-- Mobile Extension Element -->
  <xs:element name="MOBILEEXT" minOccurs="0" maxOccurs="1">
   <xs:complexType>
    <xs:attribute name="carrier">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:maxLength value="200"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="location">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:maxLength value="200"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="connspeed">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:maxLength value="30"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="conntype">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:maxLength value="30"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="signalstrength" type="xs:int"/>
   </xs:complexType>
  </xs:element>

  <!-- Transaction Page Element -->
  <xs:element name="PAGE" minOccurs="0" maxOccurs="unbounded">
   <xs:complexType>
    <xs:sequence>
     <!-- Transaction Object Element -->
     <xs:element name="OBJECT" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
       <!-- Unique Object Sequence Id -->
       <xs:attribute name="oseq" type="xs:int" use="required"/>
       <!-- Host Id (Reference to HOST.hid)-->
       <xs:attribute name="hid" type="xs:int"/>
       <!-- Connection Id (Reference to CONN.connid) -->
       <xs:attribute name="connid" type="xs:int"/>
       <!-- Return Status Code for Object -->
       <xs:attribute name="rc" type="xs:int"/>
       <!-- MIME Content Type of the Object (when rc < 300) -->
       <xs:attribute name="ctype">
        <xs:simpleType>
         <xs:restriction base="xs:string">
          <xs:maxLength value="32"/>
         </xs:restriction>
        </xs:simpleType>
       </xs:attribute>
```

```
        <!-- Response Time of the Object in Milliseconds -->
        <xs:attribute name="rtime" type="xs:int"/>
        <!-- Total Number of Object Bytes Downloaded -->
        <xs:attribute name="nbyte" type="xs:int"/>
        <!-- Start Timestamp when first "byte" (packet) was received for the object in GMT {yyyy-MM-dd
hh:mm:ss.fff}-->
        <xs:attribute name="fbstart">
         <xs:simpleType>
          <xs:restriction base="xs:string">
           <xs:maxLength value="32"/>
          </xs:restriction>
         </xs:simpleType>
        </xs:attribute>
        <!-- Time to download the first "byte" (packet) for the object in milliseconds -->
        <xs:attribute name="fbtime" type="xs:int"/>
        <!-- Start Timestamp when second "byte" (packet) was received for the object in GMT {yyyy-MM-dd
hh:mm:ss.fff} -->
        <xs:attribute name="contstart">
         <xs:simpleType>
          <xs:restriction base="xs:string">
           <xs:maxLength value="32"/>
          </xs:restriction>
         </xs:simpleType>
        </xs:attribute>
        <!-- Time to download the second "byte" (packet) for the object in milliseconds-->
        <xs:attribute name="conttime" type="xs:int"/>
        <!-- Root Url of the Host from which the object was downloaded (scheme + host)-->
        <xs:attribute name="uhost">
         <xs:simpleType>
          <xs:restriction base="xs:string">
           <xs:maxLength value="256"/>
          </xs:restriction>
         </xs:simpleType>
        </xs:attribute>
        <!-- Url of the object -->
        <xs:attribute name="upage">
         <xs:simpleType>
          <xs:restriction base="xs:string">
           <xs:maxLength value="2048"/>
          </xs:restriction>
         </xs:simpleType>
        </xs:attribute>
        <!-- URL Object request parameters -->
        <xs:attribute name="uparam">
         <xs:simpleType>
          <xs:restriction base="xs:string">
           <xs:maxLength value="2048"/>
          </xs:restriction>
         </xs:simpleType>
        </xs:attribute>
        <!-- If a redirect, the object ID of the redirection (References OBJECT.oseq) -->
        <xs:attribute name="redirobjid" type="xs:int"/>
        <xs:attribute name="csum" type="xs:int"/>
       </xs:complexType>
      </xs:element>
     </xs:sequence>
     <!-- Unique Page Sequence Id -->
     <xs:attribute name="pseq" type="xs:int" use="required"/>
     <!-- Response time of the page in milliseconds -->
     <xs:attribute name="rtime" type="xs:int"/>
     <!-- Count of objects downloaded on the page -->
     <xs:attribute name="nobj" type="xs:int"/>
     <!-- Count of object with a return code in the range between 200 and 299 (inclusive) -->
     <xs:attribute name="rc200" type="xs:int"/>
     <!-- Count of object with a return code in the range between 300 and 399 (inclusive) -->
     <xs:attribute name="rc300" type="xs:int"/>
     <!-- Count of object with a return code in the range between 400 and 499 (inclusive) -->
     <xs:attribute name="rc400" type="xs:int"/>
     <!-- Count of object with a return code in the range between 500 and 599 (inclusive) -->
     <xs:attribute name="rc500" type="xs:int"/>
     <!-- Binary flag to indicate network related errors. Default is 0 -->
     <xs:attribute name="rcnet" type="xs:int"/>
     <!-- Binary flag to indicate (internal) site related errors. Default is 0 -->
     <xs:attribute name="rcint" type="xs:int"/>
     <!-- Count of bytes downloaded for the page -->
     <xs:attribute name="nbyte" type="xs:int"/>
     <!-- Count of unique hosts referenced on the page -->
     <xs:attribute name="nhost" type="xs:int"/>
```



```
     <!-- Count of unique connections referenced on the page -->
     <xs:attribute name="nconn" type="xs:int"/>
     <!-- Content Match Text (where applicable). Contains match source text  -->
     <xs:attribute name="cmtext">
      <xs:simpleType>
       <xs:restriction base="xs:string">
        <xs:maxLength value="512"/>
       </xs:restriction>
      </xs:simpleType>
     </xs:attribute>
     <!-- Binary Content Match Status Indicator (where applicable). 0 = Not Matched, 1 = Matched -->
     <xs:attribute name="cmstat" type="xs:int"/>
     <!-- Binary flag to indicate a socket time-out error -->
     <xs:attribute name="tostat" type="xs:int"/>
     <!-- Binary flag to indicate a User Script error -->
     <xs:attribute name="usstat" type="xs:int"/>
     <!-- Binary flag to indicate a byte limit exceeded error -->
     <xs:attribute name="blstat" type="xs:int"/>
     <!-- Root object sequence number of the first object with a 200 return code.
                                          This id can be used to determine which object caused the page
to fail
                                               (References OBJECT.oseq) -->
     <xs:attribute name="rtobjseq" type="xs:int"/>
     <!-- Sum of all DNS lookup times (HOST.dnstime) for all the connections associated with this page in
milliseconds -->
     <xs:attribute name="dnssum" type="xs:int"/>
     <!-- Count of all hosts for all the connections associated with this page -->
     <xs:attribute name="ndns" type="xs:int"/>
     <!-- Sum of all vonnection times (CONN.conntime) for all the connections associated with this page in
milliseconds -->
     <xs:attribute name="connsum" type="xs:int"/>
     <!-- Count of all vonnections associated with this page -->
     <xs:attribute name="nconn1" type="xs:int"/>
     <!-- Sum of all SSL times (CONN.ssltime) for all the secure connections associated with this page in
milliseconds -->
     <xs:attribute name="sslsum" type="xs:int"/>
     <!-- Count of all secure connections associated with this page -->
     <xs:attribute name="nssl" type="xs:int"/>
     <!-- Sum of all first "byte" (packet) for all the objects associated with this page in milliseconds -
->
     <xs:attribute name="fbsum" type="xs:int"/>
     <!-- Count of all first "byte" (packet) objects associated with this page -->
     <xs:attribute name="nfb" type="xs:int"/>
     <!-- Sum of all second "byte" (packet) + subseqent for all the objects associated with this page in
milliseconds -->
     <xs:attribute name="contsum" type="xs:int"/>
     <!-- Count of all second "byte" (packet) + subsequenct objects associated with this page -->
     <xs:attribute name="ncont" type="xs:int"/>
     <!-- Processing / Client Time Time in Milliseconds -->
     <xs:attribute name="ptime" type="xs:int"/>
    </xs:complexType>
   </xs:element>
   <!-- Transaction Host Element -->
   <xs:element name="HOST" minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
     <xs:sequence>
      <!-- Transaction Connection Element -->
      <xs:element name="CONN" minOccurs="0" maxOccurs="unbounded">
       <xs:complexType>
        <!-- Unique Connection Id (Referenced from OBJECT.connid) -->
        <xs:attribute name="connid" type="xs:int" use="required"/>
        <!-- Unique Page Sequence Id (References PAGE.pseq) -->
        <xs:attribute name="page" type="xs:int"/>
        <!-- Response time to download all objects on this connection in milliseconds -->
        <xs:attribute name="rtime" type="xs:int"/>
        <!-- Start Timestamp when connection was established in GMT {yyyy-MM-dd hh:mm:ss.fff}-->
        <xs:attribute name="connstart">
         <xs:simpleType>
          <xs:restriction base="xs:string">
           <xs:maxLength value="32"/>
          </xs:restriction>
         </xs:simpleType>
        </xs:attribute>
        <!-- Time to establish the connection in milliseconds -->
        <xs:attribute name="conntime" type="xs:int"/>
        <!-- Start Timestamp when SSL handshake was established in GMT {yyyy-MM-dd hh:mm:ss.fff}-->
        <xs:attribute name="sslstart">
         <xs:simpleType>
```

```
          <xs:restriction base="xs:string">
           <xs:maxLength value="32"/>
          </xs:restriction>
         </xs:simpleType>
        </xs:attribute>
        <!-- Time to complete the SSL handshake phase of the connection in milliseconds -->
        <xs:attribute name="ssltime" type="xs:int"/>
        <!-- Total number of bytes downloaded on this connection -->
        <xs:attribute name="nbyte" type="xs:int"/>
        <!-- Total number of errors occurring on this connection -->
        <xs:attribute name="nerr" type="xs:int"/>
        <!-- Total number of objects downloaded on this connection -->
        <xs:attribute name="nobj" type="xs:int"/>
       </xs:complexType>
      </xs:element>
     </xs:sequence>
     <!-- Unique Host Id (Referenced from OBJECT.hid) -->
     <xs:attribute name="hid" type="xs:int" use="required"/>
     <!-- Unique Page Sequence Id (References PAGE.pseq) -->
     <xs:attribute name="page" type="xs:int"/>
     <!-- Start Timestamp when DNS lookup was completed in GMT {yyyy-MM-dd hh:mm:ss.fff}-->
     <xs:attribute name="dnsstart">
      <xs:simpleType>
       <xs:restriction base="xs:string">
        <xs:maxLength value="32"/>
       </xs:restriction>
      </xs:simpleType>
     </xs:attribute>
     <!-- Time to complete the DNS lookup in milliseconds -->
     <xs:attribute name="dnstime" type="xs:int"/>
     <!-- Total number of bytes downloaded from this host -->
     <xs:attribute name="nbyte" type="xs:int"/>
     <!-- Total number of errors detected from this host -->
     <xs:attribute name="nerr" type="xs:int"/>
     <!-- Total number of errors downloaded from this host -->
     <xs:attribute name="nobj" type="xs:int"/>
     <!-- Total number of connections established to this host -->
     <xs:attribute name="nconn" type="xs:int"/>
     <!-- IP Address of this host -->
     <xs:attribute name="ip">
      <xs:simpleType>
       <xs:restriction base="xs:string">
        <xs:maxLength value="20"/>
       </xs:restriction>
      </xs:simpleType>
     </xs:attribute>
    </xs:complexType>
   </xs:element>
  </xs:sequence>
  <!-- Test Id : Monitor Id (Backbone), Monitor Instance Id (Last Mile / Private Peer / Mobile) -->
  <xs:attribute name="mid" type="xs:int" use="required"/>
  <!-- Location Id : Site Id (Backbone), Population Id (Last Mile / Private Peer / Mobile) -->
  <xs:attribute name="sid" type="xs:int" use="required"/>
  <!-- Test Timestamp in GMT. {yyyy-MM-dd hh:mm:ss.fff}-->
  <xs:attribute name="ttime">
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:maxLength value="32"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <!-- Backbone Site Name -->
  <xs:attribute name="sname" use="optional">
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:maxLength value="600"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <!-- Last Mile / Private Peer / Mobile Population Name -->
  <xs:attribute name="popname" use="optional">
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:maxLength value="600"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <!-- End-To-End Response Time in Milliseconds -->
```

```
 <xs:attribute name="rtime" type="xs:int"/>
 <!-- Count of Total Pages Tested -->
 <xs:attribute name="tpt" type="xs:int"/>
 <!-- Count of Total Pages Successful -->
 <xs:attribute name="tps" type="xs:int"/>
 <!-- Ciybt if Total Pages Failed -->
 <xs:attribute name="tpf" type="xs:int"/>
 <!-- Count of Total Object Tested -->
 <xs:attribute name="tot" type="xs:int"/>
 <!-- Count of Total Objects Successful -->
 <xs:attribute name="tos" type="xs:int"/>
 <!-- Count of Total Objects Failed -->
 <xs:attribute name="tof" type="xs:int"/>
 <!-- Total Number of Bytes Downloaded -->
 <xs:attribute name="nbyte" type="xs:int"/>
 <!-- Collection Agent Type Discriminator -->
 <xs:attribute name="browsertype">
  <xs:simpleType>
   <xs:restriction base="xs:string">
    <xs:maxLength value="50"/>
    <xs:pattern value="UTA|IE7|FF35"/>
   </xs:restriction>
  </xs:simpleType>
 </xs:attribute>
</xs:complexType>

<!--
       **********************************
       Data Cut Response Type
       **********************************
       -->
<xs:complexType name="DataCutTestType">
 <!-- Test Cut Event Timestamp in GMT. {yyyy-MM-dd hh:mm:ss.fff}-->
 <xs:attribute name="cutdate">
  <xs:simpleType>
   <xs:restriction base="xs:string">
    <xs:maxLength value="32"/>
   </xs:restriction>
  </xs:simpleType>
 </xs:attribute>
 <!-- Test Id : Monitor Id (Backbone), Monitor Instance Id (Last Mile / Private Peer / Mobile) -->
 <xs:attribute name="mid" type="xs:int" use="required"/>
 <!-- Location Id : Site Id (Backbone), Population Id (Last Mile / Private Peer / Mobile) -->
 <xs:attribute name="sid" type="xs:int" use="required"/>
 <!-- Test Timestamp in GMT. {yyyy-MM-dd hh:mm:ss.fff}-->
 <xs:attribute name="ttime">
  <xs:simpleType>
   <xs:restriction base="xs:string">
    <xs:maxLength value="32"/>
   </xs:restriction>
  </xs:simpleType>
 </xs:attribute>
</xs:complexType>

<!--
       **********************************
       Message Response Type
       **********************************
       -->
<xs:complexType name="MessageType">
 <!-- Test Id : Monitor Id (Backbone), Monitor Instance Id (Last Mile / Private Peer / Mobile) -->
 <xs:attribute name="mid" type="xs:int" use="required"/>
 <!-- General Message Text -->
 <xs:attribute name="Msg" type="xs:string" use="required"/>
</xs:complexType>

<!--
       **********************************
       Monitor Class Designator Type
       Common element in all responses.
       **********************************
       -->
<xs:complexType name="MonitorClassDesignatorType">
 <xs:attribute name="MonitorClassDesignator">
  <xs:simpleType>
   <xs:restriction base="xs:string">
    <xs:maxLength value="20"/>
   </xs:restriction>
```

```
  </xs:simpleType>
 </xs:attribute>
</xs:complexType>

<!--
      **********************************
      Streaming Response Type
      **********************************
      -->
<xs:complexType name="StreamTestType">
 <xs:sequence>
  <!-- Event Element -->
  <xs:element name="EVENT" minOccurs="0" maxOccurs="unbounded">
   <xs:complexType>
    <!-- Unique event id -->
    <xs:attribute name="sevt_id" type="xs:int" use="required"/>
    <!-- Event name -->
    <xs:attribute name="name">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:maxLength value="100"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <!-- Offset of event occurrence from start of test in milliseconds -->
    <xs:attribute name="offset" type="xs:int"/>
   </xs:complexType>
  </xs:element>
  <!-- Metadata Element -->
  <xs:element name="META" minOccurs="0" maxOccurs="unbounded">
   <xs:complexType>
    <!-- Unique sequence id of metadata element -->
    <xs:attribute name="metaseq" type="xs:int" use="required"/>
    <!--
                                    Unique metadata key name for metadata element
                                    Common key values:
                                           'mediaFormat'
                                           'frameSize'
                                           'totalBytes'
                                           'frameRate'
                                           'recommendedFrameRate'
                                           'recommendedBandwidth'
                                           'recommendedDuration'
                                    -->
    <xs:attribute name="smetakey">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:maxLength value="255"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <!--
                                    Metadata value for metadata element. Type depends on metadata key
                                        'mediaFormat' - Data type is string(255)
                                           'frameSize' - Data type is unsigned long
                                           'totalBytes' - Data type is unsigned long
                                           'frameRate' - Data type is unsigned long
                                           'recommendedFrameRate' - Data type is unsigned long
                                           'recommendedBandwidth' - Data type is unsigned long
                                           'recommendedDuration' - Data type is unsigned long
                                    -->
    <xs:attribute name="smetaval">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:maxLength value="255"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
   </xs:complexType>
  </xs:element>
  <!-- Trace Route Element -->
  <xs:element name="TRACE" minOccurs="0" maxOccurs="unbounded">
   <xs:complexType>
    <!-- Unique trace sequence / hop id-->
    <xs:attribute name="hopnum" type="xs:int" use="required"/>
    <!-- IP address of destination -->
    <xs:attribute name="ip">
     <xs:simpleType>
```

```
      <xs:restriction base="xs:string">
       <xs:maxLength value="20"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <!-- Host name of destination -->
    <xs:attribute name="host">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:maxLength value="100"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <!-- Time delay in milliseconds -->
    <xs:attribute name="delay" type="xs:int"/>
   </xs:complexType>
  </xs:element>
 </xs:sequence>
 <!-- Test Id : Monitor Id (Backbone), Monitor Instance Id (Last Mile / Private Peer / Mobile) -->
 <xs:attribute name="mid" type="xs:int" use="required"/>
 <!-- Backbone Site Id -->
 <xs:attribute name="sid" type="xs:int" use="required"/>
 <!-- Test Timestamp in GMT. {yyyy-MM-dd hh:mm:ss.fff}-->
 <xs:attribute name="ttime" type="xs:dateTime" use="required"/>
 <!-- Page sequence id for transaction integrated streaming tests. -1 for standalone tests. -->
 <xs:attribute name="pseq" type="xs:int" use="required"/>
 <!-- Unique stream sequence id -->
 <xs:attribute name="sseq" type="xs:int" use="required"/>
 <!-- Start Timestamp in GMT. {yyyy-MM-dd hh:mm:ss.fff}-->
 <xs:attribute name="stime" type="xs:dateTime" use="required"/>
 <!-- System (high-level) error code -->
 <xs:attribute name="errrsyscode" type="xs:int" use="required"/>
 <!-- System error description -->
 <xs:attribute name="errdesc">
  <xs:simpleType>
   <xs:restriction base="xs:string">
    <xs:maxLength value="1000"/>
   </xs:restriction>
  </xs:simpleType>
 </xs:attribute>
 <!-- Raw (low-level) System error description -->
 <xs:attribute name="errrawcode" type="xs:int"/>
 <!-- Stream protocol. Sample values include (but not limited to): {'http'|'rtmp'|'rtspt'|'rtspu'} -->
 <xs:attribute name="sproto">
  <xs:simpleType>
   <xs:restriction base="xs:string">
    <xs:maxLength value="100"/>
   </xs:restriction>
  </xs:simpleType>
 </xs:attribute>
 <!-- IP Address of target host -->
 <xs:attribute name="ip">
  <xs:simpleType>
   <xs:restriction base="xs:string">
    <xs:maxLength value="20"/>
   </xs:restriction>
  </xs:simpleType>
 </xs:attribute>
 <!-- Url of tested stream -->
 <xs:attribute name="url">
  <xs:simpleType>
   <xs:restriction base="xs:string">
    <xs:maxLength value="4365"/>
   </xs:restriction>
  </xs:simpleType>
 </xs:attribute>
 <!-- Duration of stream test in milliseconds -->
 <xs:attribute name="durinms" type="xs:int"/>
 <!-- Frames per second -->
 <xs:attribute name="fps" type="xs:int"/>
 <!--
             Stream End Type Designator
                    SUCCESS - Test completed successfully
                    EXPIRED - Stream expired before end of test was reached
                    FAILED  - Test Failed
             -->
 <xs:attribute name="endtype">
  <xs:simpleType>
```

```
    <xs:restriction base="xs:string">
     <xs:maxLength value="7"/>
     <xs:pattern value="SUCCESS|EXPIRED|FAILED"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <!-- Bites per second -->
  <xs:attribute name="bps" type="xs:int"/>
  <!-- Media Server Type -->
  <xs:attribute name="stype">
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:maxLength value="100"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <!-- Server Platform -->
  <xs:attribute name="splat">
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:maxLength value="100"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <!-- Server Version -->
  <xs:attribute name="sver">
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:maxLength value="100"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <!-- DNS Offset Time from start of test in milliseconds -->
  <xs:attribute name="dnsoffset" type="xs:int"/>
  <!-- Total DNS time in milliseconds -->
  <xs:attribute name="dnstime" type="xs:int"/>
  <!-- Connection Offset Time from start of test in milliseconds -->
  <xs:attribute name="connoffset" type="xs:int"/>
  <!-- Total connection time in milliseconds -->
  <xs:attribute name="conntime" type="xs:int"/>
  <!-- Initial buffer time in milliseconds -->
  <xs:attribute name="initbuftime" type="xs:int"/>
  <!-- Total re-buffer time in milliseconds -->
  <xs:attribute name="rebuftime" type="xs:int"/>
  <!-- Count of rebuffer events -->
  <xs:attribute name="rebufcnt" type="xs:int"/>
  <!-- Initial play time in milliseconds -->
  <xs:attribute name="initplytime" type="xs:int"/>
  <!-- Start play time in milliseconds -->
  <xs:attribute name="spt" type="xs:int"/>
  <!-- Initial re-buffer time in milliseconds -->
  <xs:attribute name="initrebuftime" type="xs:int"/>
  <!-- Count of PLAY events -->
  <xs:attribute name="plycnt" type="xs:int"/>
 </xs:complexType>

 <!--
       ************************************
       Status Response Type
       ************************************
       -->
 <xs:complexType name="COpStatusData">
  <xs:sequence>
   <!-- Status Type Indicator -->
   <xs:element name="eStatus" type="StatusType"/>
   <!-- Error message when eStatus == "STATUS_FAILED" -->
   <xs:element name="sErrorMessage" type="xs:string" minOccurs="0"/>
  </xs:sequence>
 </xs:complexType>
 <xs:simpleType name="StatusType">
  <xs:restriction base="xs:string">
   <!-- Operation Successful -->
   <xs:enumeration value="STATUS_SUCCESS"/>
   <!-- Operation Failed - see ErrorMessage for details -->
   <xs:enumeration value="STATUS_FAILED"/>
  </xs:restriction>
 </xs:simpleType>
</xs:schema>
```

Chapter 4
FTP Data Feeds

Introduction

The FTP data feed is the preferred method of downloading large amounts of data from the Gomez Platform for integration with customer applications.

Getting Started

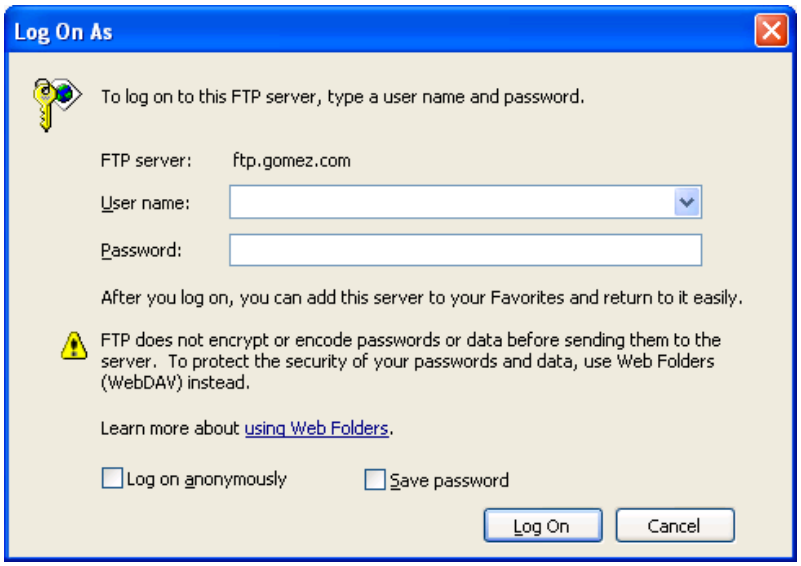
To start receiving an FTP Data Feed, you need to do two things:

- Create tests using the Gomez Portal
- Obtain an FTP data feed login from Gomez Client Services

After you obtain your FTP data feed login, you automatically receive test results in your FTP account directory.

To access your FTP account directory:

1. Open Windows Explorer.
2. In the **Address** field enter: `ftp://ftp.gomez.com/`
3. In the menu bar, navigate to **File > Login As**. The Log On As dialog box is displayed.



4. Enter your FTP data feed credentials in the **User name** and **Password** fields.

Depending on the tests you have set up in your Gomez account, Gomez deposits one or more of the corresponding FTP data feed in your FTP account directory:

Data Feed Type	Data Source
T	UTA
LM	Last Mile
TO	UTA object-level

Each data feed is a ZIP file containing one or more files with names in the following format:

[t|to|lm|lmo|monitor|site]-year-month-day_hour-minute.*

All of the zip files contain at least one text file that contains test result data. The T zip file contains two additional files:

- T file contains test result data
- Monitor file contains monitor-specific information
- Site file contains site-specific information

Each text file in the data feed is in "|"-separated-value format. Each value is encased in double quotes, since many values contain commas and spaces.

T File Layout

This file contains the following UTA test data:

Field	Description
monitor_id	Unique test identifier
testtime	Test time in GMT
site_id	Node where the test ran
mrresponsetime	Response time in milliseconds. Sum all the `troresponsetimes` for a given `seqno` in this file to obtain the `mrresponsetime`. This field is NULL, if the test fails.
seqno	Page number of the test
status	"0" if success; GPN error code, if failure
troresponsetime	Response time in milliseconds of the page. Sum all the `troresponsetimes` for a given `seqno` in this file to obtain the `mrresponsetime`. This field is NULL, if the test fails.
url	URL of the page tested
name	Name of the node where the test ran
RealURLFlag	Always "0"

Monitor File Layout

This file lists the following test (monitor) data:

Field	Description
Monitor_id	Unique test identifier
Description	Description of the test
Class	TRANSACTION (for UTA Agent)
URL	If class=MONITOR then URL of the test: otherwise, NULL
ComponentFlag	1 if page objects are downloaded, 0 otherwise
Status	ACTIVE, INACTIVE, or DELETE
Frequency	Interval that the test runs in number of milliseconds. For example, frequency=3600 means that the test runs once every hour.
CreateDate	Date when the test was created.
ModDate	Date when the test was last changed.

Site File Layout

This file lists the following node (site) data:

Field	Description
site_id	Unique node identifier
name	Name of the node
Status	ACTIVE, INACTIVE, or DELETE
backbone	Internet Service Provider of the data center node
City	City of the data center node
State	State of the data center node
CountryCode	Country code of the data center node
ipaddress	IP address of the data center node

LM File Layout

This file contains the following Last Mile test data:

Field	Description
mgbmonitor_id	Unique test identifier
testtime	Test time in GMT
mrresponsetime	Response time in milliseconds; NULL, if test fails
seqno	Page number of the test
status	"0" if success; GPN error code if failure
troresponsetime	Response time in milliseconds of the page. Sum all the troresponsetimes for a given seqno in this file to obtain the mrresponsetime. This field is NULL, if the test fails.
url	URL of the page tested
name	Name of the node where the test ran
IP	IP address of the url
LMA_id	Unique identifier of the Last Mile computer
LMA_REGION_NAME	Region of the Last Mile computer
LMA_ISP_NAME	Internet Service Provider of the Last Mile computer
LMA_IP_ADDRESS	IP address of the Last Mile computer
avgthruput	Average throughput in bits per second of the Last Mile machine
compbytes	Total bytes downloaded; "0" if test fails

TO File Layout

This data feed requires special set up. Contact Gomez Client Services to activate this data feed. The TO file contains the following backbone UTA object-level test data:

Field	Description
monitor_id	Unique test identifier
testtime	Test time in GMT
site_id	Unique node identifier of the site where the test ran
sitename	Name of the node where the test ran
mrresponsetime	Response time in milliseconds; NULL, if test fails
pageseqnum	Page number of the test
pagestatus	GOOD if success; GPN error code if failure
pageresponsetime	Response time in milliseconds of the page. Sum all the pageresponsetimes for a given seqno in this file to obtain the mrresponsetime. This field is NULL, if the test fails.
url	URL of an object on the page
compReturnCode	HTTP return code of the object
compDNS	Usually "0", but if the first object, the time in milliseconds to DNS connect
compConnect	Time in milliseconds to connect to the object
compSSL	Time in milliseconds to SSL connect to the object
compFirstByte	Time in milliseconds to download the first byte of the object
compContent	Time in milliseconds to download the object
compIP	IP address of the server hosting the object
objseqnum	Sequence number of the object relative to the entire test
host_id	Sequence number of the host relative to the entire test
connection_id	Sequence number of the connection relative to the entire test

LMO File Layout

This data feed requires special set up. Contact Gomez Client Services to activate this data feed. The LMO file contains the following Last Mile object-level test data:

Field	Description
monitor_id	Unique test identifier
testtime	Test time in GMT
site_id	Unique node identifier of the site where the test ran
mrresponsetime	Response time in milliseconds; NULL, if test fails
pageseqnum	Page number of the test
pagestatus	GOOD if success; GPN error code if failure
pageresponsetime	Response time in milliseconds of the page. Sum all the pageresponsetimes for a given seqno in this file to obtain the mrresponsetime. This field is NULL, if the test fails.
url	URL of an object on the page
compReturnCode	HTTP return code of the object
compDNS	Usually "0", but if the first object, the time in milliseconds to DNS connect
compConnect	Time in milliseconds to connect to the object
compSSL	Time in milliseconds to SSL connect to the object
compFirstByte	Time in milliseconds to download the first byte of the object
compContent	Time in milliseconds to download the object
compIP	IP address of the server hosting the object
objseqnum	Sequence number of the object relative to the entire test
host_id	Sequence number of the host relative to the entire test
connection_id	Sequence number of the connection relative to the entire test
peer_region_name	Name of the region that the peer is located in
peer_isp_name	ISP used by the peer
peer_ip_address	IP address of the peer
avgthruput	Average throughput in bits per second of the Last Mile machine
numbytes	Total bytes downloaded for the object