Data Pulse

User Guide

Confidentiality Notice

Version 1.7

Table of Contents

Introduction

Keynote Data Pulse^{tm} provides a near real-time feed of performance data for Transaction Perspective, Application Perspective, and Mobile Web Perspective measurements. (This document covers only Data Pulse for Transaction Perspective and Application Perspective; Mobile Web Perspective will be covered in a future update.)

Data Pulse provides a high degree of security, reliability, and flexibility for today's operational environments. Keynote data delivered through Data Pulse is updated dynamically and routed through a fault-tolerant infrastructure with low latency.

Subscribers to Data Pulse can seamlessly integrate Keynote data into company tools and applications. Network operation centers can use Data Pulse to create custom alerts or non-standard alerts and/or integrate the data source with their enterprise management systems (EMS). Furthermore, Data Pulse can be used to integrate Keynote data with other company data sources. The data can then adapted to the user's needs through manipulation or visualization in internal or customer facing applications.

Benefits

Data Pulse provides the following benefits to Keynote customers:

- Integrate real-time performance data with other operational data, without purchasing additional hardware or software.

- Manipulate, transform, and visualize near real-time data in nearly limitless ways using a standardized XML format.

- Design your own alarm and event criteria using the real-time data feed.

- Receive faster alarm notification when integrated with company management systems.

- Depend on a highly reliable source of up-to-the-minute mission-critical performance data to the enterprise.Your sensitive company data is provided the highest security. **Other Keynote Data Feed Services**

- Keynote Tab-Delimited DataFeed - a data feed for 24 hour or 15-minute dumps of Keynote data in a flat-file, tab-delimited format.

- Keynote XML DataFeed - a data feed for 24 hour or 15-minute dumps of Keynote data in XML format.

Contact Keynote for more information about these services.

Data Pulse Service Overview

Keynote Data Pulse provides on-demand secure access to frequently-updated raw performance data via a secure server. Users who access this service are authenticated with a Keynote login and password. The data format is an XML grammar developed by Keynote. Keynote Data Pulse currently supports near real-time data for Application Perspective, Transaction Perspective, and Mobile Web Perspective.

Data Pulse users can access the data source as frequently as once every minute, although customer processing loads may prevent users from accessing updated data this frequently. Each user request can retrieve up to a 15 minute window of rolling data from the Data Pulse system.

The Data Pulse system provides updated data based on the inserted time of the data point. This means that delayed data points (measured during a different interval than is requested by the customer) that are caused by communication latency in the Keynote measurement infrastructure may appear in the retrieved data. For example, a request for the last five minutes of data may also receive a data point that was measured at some point earlier in time than the last five minutes.
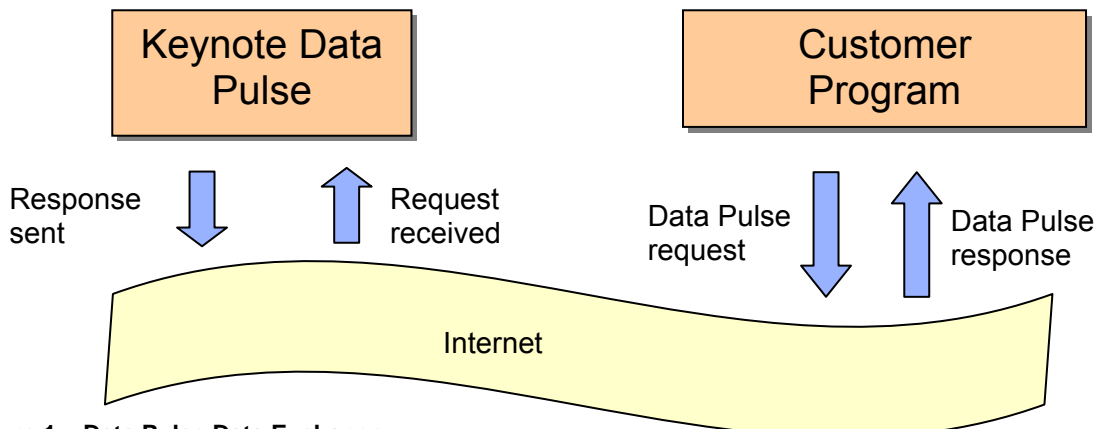
DATA PULSE System Overview

Figure 1 – Data Pulse Data Exchange

Customers can connect from any network having Internet connectivity (presuming proxy settings allow clients to establish outbound connections on ports 443 and/or 80).

Data Exchange Protocol

The communications scheme used for access to the Data Pulse system is XML over HTTPS. (Customer programs will parse the XML for measurement data.) Keynote deploys a dedicated set of HTTP/S servers as a portal for interacting with our internal Data Pulse system. Connecting to these servers, clients will submit Data Pulse requests in the form of an XML document.

The Data Pulse server name is:
https://Data Pulse.keynote.com/dps/xmlpost

Data feed requests are encapsulated using an XML grammar developed by Keynote.

The content of the XML data feed request is referred to as an XML request "payload." The payload is included as part of the HTTP POST method and sent with the connection request to the server. When the POST request is received and processed, an HTTP response is returned. The HTTP response generated by our server also includes an XML payload. The response payload will provide status information regarding the success or failure of the data feed request along with the resulting measurement data (if successful).

Security

Data Pulse security is addressed at three levels:

- **Encryption.** Leveraging HTTPS/SSL, data sent over the wire is encrypted on the client side and decrypted on the server side. This will protect the content of the data during transmission of requests and responses.

- **Authentication.** By utilizing HTTP Basic Authentication, username/password values are required to gain access to the real-time data.

- **Access Control.** Once authenticated, the agreement-id specified in the request is cross-referenced to the company associated with the authenticated login. If the agreement does not belong to the company associated with the authenticated login, access will be denied.

HTTP Basic Authentication is used to authenticate client requests. Basic Authentication transmits the username:password pair in a base64 encoded form from client to server. While this approach does not encrypt the username/password values, use of SSL (HTTPS) provides encryption over the base64 encoding to protect these values along with the data itself.

Customers who do not provide the proper HTTP Authentication header and values will receive an "HTTP 401 Authorization Required" response from the server and will be denied data access until proper authentication values are provided. In addition, customers who request data for an agreement that is not associated with the company to which they belong will also be denied access.

Connection/Request Frequency

Clients will be allowed to issue data feed requests as frequently as once every 60 seconds. The data retrieved will be based on the agreement and the time range specified by the customer in the request payload. If customers request data more frequently than 60 seconds, they will be denied access.

Latency

As previously mentioned, data point latency (defined as the difference between the time a measurement was taken and the time it is available for the customer) will on average be no more than 60 seconds.

Data Window

Data will be reported based on its arrival time into the Data Pulse system, rather than on the time the measurement itself was taken. The difference between the two approaches is that data can be delayed coming in from the measurement agents. In this case, if a

measurement were delayed by a couple of minutes and a customer asked for the last two minutes of data, that measurement would not appear in the data response. However, because the arrival time is used as the discriminator for Data Pulse, delayed measurements will be delivered as soon as they become available. The actual time the measurement was taken is always reported in each record.

Meta-Data

Keynote meta-data is information about Keynote agents used to take measurements. Meta-data information is presented to map Keynote Slot ID and Agent ID to actual values. Data Pulse users will have the option of retrieving this data together with the measurement payload or retrieving it through use of a separate request/response payload.

Customers who currently retrieve their data through the Keynote Secure Data Access service will find meta-data in an index file located in their agreement folder. Keynote meta-data is updated every 15-minutes. Meta-data changes could be caused by agent changes, which are infrequent, or by customer changes. Since changes in meta-data take at least one hour to be implemented, the 15-minute updates of meta-data provided by Data Pulse will capture any changes.

Input Request Grammar

The DTD below describes the XML grammar to be used when submitting data feed REQUEST payloads:

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

<!ELEMENT DPS_REQUEST (GET_MEASUREMENTS_BY_AGREEMENT) >
<!ELEMENT GET_MEASUREMENTS_BY_AGREEMENT EMPTY >
<!ATTLIST GET_MEASUREMENTS_BY_AGREEMENT
        request_id          CDATA #REQUIRED
        agreement_id        CDATA #REQUIRED
        delta_time          CDATA #REQUIRED
        service             CDATA #REQUIRED
        meta_data           CDATA "false"
        subservice          CDATA ""
>
```

Request Element/Attribute Descriptions

Special Considerations:

Unique values for request_id attribute: For future use, it is possible that several elements in the XML request payload may contain an attribute named request_id. The value for this attribute must be unique for all request_id values within the request XML payload regardless of which elements declare them. This value is used to associate response objects with the requests from which the responses initiated.

DPS_REQUEST

The DPS_REQUEST element is at the root of the XML request grammar. It has no

attributes and currently supports exactly one
GET_MEASUREMENTS_BY_AGREEMENT child element.

GET_MEASUREMENTS_BY_AGREEMENT

The GET_MEASUREMENTS_BY_AGREEMENT element is a child of the
DP_REQUEST element. It is used to query the system for measurement data. This
element consists of the following attributes:

Attributes

request_id

The request_id is an alpha-numeric identifier that must be unique from all other
request_id attribute values within the XML file. It can be any combination of numbers
and/or letters.

agreement_id

Specifies the agreement ID that contains the measurements of interest. This should be a
numeric value and must already exist in the Keynote database.

delta_time

Specifies the time window (in seconds) for which measurements will be received. As an
example, a value of 120 specifies interest in receiving the last two minutes of
measurement data for all slots in the specified agreement.

Note: Data is returned based on the time stamped on arrival to our servers, not when
the measurement was taken. This ensures that if data arrives late (due to agent delivery
delays), it will still be delivered to clients according to its arrival time. The time when the
measurement was taken is indicated within the record as 'datetime'. See _____.

service

Since agreements may contain slots belonging to multiple services, this attribute is used
to specify which type of slot measurement data should be returned for this request.
Legal value for Transaction Perspective and Application Perspective is TRANSACTION.

meta_data

The attribute specifies whether meta-data should be returned with the measurement
data. Meta-data is the equivalent of the 'index' file data provided with Keynote's
DataFeed service.

Subservice

Optional. Reserved for future use.

Request Element/Attribute Sample File

```
<?xml version="1.0"?>



<!DOCTYPE DPS_REQUEST SYSTEM "http://Data Pulse.keynote.com/dps/config/dps-
request.dtd">
```

```
<!-- This is the input XML that customer's will post to get a Data Pulse
          set of data. DPS_REQUEST attributes are as follows:

     AGREEMENT_ID            = Agreement ID
     DELTA_TIME        = Last (n) seconds of data
     SERVICE           = [ STANDARD | TRANSACTION ]
     META_DATA         = [ true | false ]

          The following example asks for the last 2 minutes of TRANSACTION
          (TxP) data from all slots under agreement 12345. Meta data
          is false, so the index info (from DataFeed) will NOT be included.
-->


    <DPS_REQUEST>
          <GET_MEASUREMENTS_BY_AGREEMENT request_id="1"
agreement_id="12345"
delta_time="120" service="TRANSACTION" meta_data="false" />
</DPS_REQUEST>
```

Response Output

The Data Pulse server responds to a valid request with an XML payload. This payload contains an XML document that consists of three subcomponents:

- Status information

- Meta-data

- Measurement information

Status Information [DPS_STATUS]

The status information is contained in the dp_status element. This tag has the following XML form:

```
<!ELEMENT dps_status (status+) >

<!ELEMENT status ( code, severity, msg, description, args*)>

<!ATTLIST status request_id CDATA #REQUIRED>

<!ELEMENT code (#PCDATA)>

<!ELEMENT severity (#PCDATA)>

<!ELEMENT msg (#PCDATA)>

<!ELEMENT description (#PCDATA)>

<!ELEMENT args ( argname, argvalue )+ >

<!ELEMENT argname (#PCDATA)>

<!ELEMENT argvalue (#PCDATA)>
```

Multiple `status` blocks can be nested under the `dps_status` element. Each `status` contains a `code` element. The `code` element provides a unique value for each specific status type. A description of the status types can be found in the response declarations table towards the end of this document.

Meta-Data [TXN_META_DATA]

In the meta-data section `TXN_META_DATA` is used for the Transaction Perspective and Application Perspective. The meta data consists of two parts – one for *agent* meta-data and the other for *slot* meta-data. This data represents a mapping of agent and slot ids to the detailed information about the agents and slots associated with the specified

agreement. The data in these sections provides the same information found in the INDEX file that is part of the basic Keynote DataFeed service.

Note: This section is optional and will only be included when the meta_data attribute value is declared «true» in the XML request payload.

Measurement Information [DP_TXN_MEASUREMENTS]

In measurement data `DP_TXN_MEASUREMENTS` is used for the Transaction Perspective and Application Perspective. These sections contain the actual measurements returned by the Data Pulse cache.

Transaction Perspective and Application Perspective

For Transaction Perspective and Application Perspective `DP_TXN_MEASUREMENTS` is an outer tag that encapsulates all measurements. Nested under this tag are zero or more `TXN_MEASUREMENT` elements, each instance representing a specific transaction measurement.

Each `TXN_MEASUREMENT` element then supports one or more `TXN_PAGE` elements, with one instance for each page of the transaction. Each `TXN_PAGE` element contains sub-elements that define page-level information. If there are content-errors or redirects on a page the content details will be present in the `TXN_PAGE_DETAILS` sub-element of the `TXN_PAGE`. `TXN_PAGE_DETAILS` consists of the following sub-elements:

TXN_DETAIL_ERROR

Represents a transaction playback error record. This element is sent when there was an error that caused the playback engine to abort the transaction. This would usually be caused by changes to the page after recording the script.

TXN_REDIRECT

Represents a redirect detail record. Page-level redirects are sent whenever a server-side redirect occurs. There can be zero or more redirects associated with each page of the transaction. There can also be redirect elements associated with page content (elements), however, these redirects are only sent when there is a content error detected on the page.

TXN_BASE_PAGE

Represents a base page detail record. There will be exactly one of these elements present (per page). However, this element is only sent if a Content Error (or Redirect) occurred on the page.

TXN_PAGE_ELEMENT

Represents a page element detail record. There can be zero or more of these elements representing content on the page. However, these elements are only sent if there was at least one content error on the page.

For the detailed element and attribute specification please refer to the DTD for the Transaction Perspective and Application Perspective response payload found in the Output Response Grammar section of this document.

Output Response Grammar

This section contains the DTDs (Document Type Definitions) for the output response payloads for Transaction Perspective and Application Perspective.

Transaction Perspective/Application Perspective Output Response Grammar

The DTD below describes the XML grammar used for the Transaction Perspective and Application Perspective RESPONSE payload:

```
<!ELEMENT DP_RESPONSE ( dps_status, TXN_META_DATA? ,
DP_TXN_MEASUREMENTS? ) >


<!ELEMENT dps_status (status+) >
<!ELEMENT status ( code, severity, msg, description, args*)>
<!ATTLIST status request_id CDATA #REQUIRED>
<!ELEMENT code (#PCDATA)>
<!ELEMENT severity (#PCDATA)>
<!ELEMENT msg (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT args ( argname, argvalue )+ >
<!ELEMENT argname (#PCDATA)>
<!ELEMENT argvalue (#PCDATA)>



<!ELEMENT TXN_META_DATA (AGENT_META_DATA+ , SLOT_META_DATA+) >
<!ATTLIST TXN_META_DATA agreement_id CDATA #REQUIRED >
```

```
<!ELEMENT AGENT_META_DATA EMPTY >
<!ATTLIST AGENT_META_DATA      agent_id       CDATA #REQUIRED >
<!ATTLIST AGENT_META_DATA      description    CDATA #REQUIRED >
<!ATTLIST AGENT_META_DATA      weight   CDATA #REQUIRED >
<!ATTLIST AGENT_META_DATA      ip             CDATA #REQUIRED >
<!ATTLIST AGENT_META_DATA      backbone       CDATA #REQUIRED >
<!ATTLIST AGENT_META_DATA      instance_id    CDATA #REQUIRED >


<!ELEMENT SLOT_META_DATA EMPTY >
<!ATTLIST SLOT_META_DATA       slot_id  CDATA #REQUIRED >
<!ATTLIST SLOT_META_DATA       slot_alias     CDATA #REQUIRED >
<!ATTLIST SLOT_META_DATA       url            CDATA #REQUIRED >
<!ATTLIST SLOT_META_DATA       pages          CDATA #REQUIRED >



<!-- DP_TXN_MEASUREMENTS consist of 0 or more TXN_MEASUREMENT
elements.... -->
<!ELEMENT DP_TXN_MEASUREMENTS (TXN_MEASUREMENT*) >



<!ELEMENT TXN_MEASUREMENT (TXN_SUMMARY,TXN_ERROR?,TXN_PAGE+) >
<!ATTLIST TXN_MEASUREMENT slot                CDATA #REQUIRED >
<!ATTLIST TXN_MEASUREMENT agent               CDATA #REQUIRED >
<!ATTLIST TXN_MEASUREMENT agent_inst  CDATA #REQUIRED >
<!ATTLIST TXN_MEASUREMENT datetime    CDATA #REQUIRED >
<!ATTLIST TXN_MEASUREMENT target       CDATA #REQUIRED >

<!ELEMENT TXN_SUMMARY EMPTY >
<!ATTLIST TXN_SUMMARY delta_user_msec CDATA #REQUIRED >
```

```
<!ATTLIST TXN_SUMMARY delta_msec              CDATA #REQUIRED >
<!ATTLIST TXN_SUMMARY content_errors   CDATA #REQUIRED >


<!ELEMENT TXN_ERROR EMPTY >
<!ATTLIST TXN_ERROR code              CDATA #REQUIRED >
<!ATTLIST TXN_ERROR page              CDATA #REQUIRED >



<!ELEMENT TXN_PAGE (TXN_PAGE_PERFORMANCE, TXN_PAGE_OBJECT,
TXN_PAGE_STATUS, TXN_PAGE_DETAILS?) >
<!ATTLIST TXN_PAGE page_seq              CDATA #REQUIRED >



<!ELEMENT TXN_PAGE_PERFORMANCE EMPTY >
<!ATTLIST TXN_PAGE_PERFORMANCE delta_msec        CDATA #IMPLIED >
<!ATTLIST TXN_PAGE_PERFORMANCE delta_user_msec        CDATA
#IMPLIED >
<!ATTLIST TXN_PAGE_PERFORMANCE start_msec        CDATA #IMPLIED >
<!ATTLIST TXN_PAGE_PERFORMANCE dns_lookup_msec        CDATA
#IMPLIED >
<!ATTLIST TXN_PAGE_PERFORMANCE first_byte_msec        CDATA
#IMPLIED >
<!ATTLIST TXN_PAGE_PERFORMANCE system_delta        CDATA
#IMPLIED >
<!ATTLIST TXN_PAGE_PERFORMANCE connect_delta        CDATA
#IMPLIED >
<!ATTLIST TXN_PAGE_PERFORMANCE first_packet_delta        CDATA
#IMPLIED >
<!ATTLIST TXN_PAGE_PERFORMANCE ssl_handshake_delta        CDATA
#IMPLIED >
<!ATTLIST TXN_PAGE_PERFORMANCE request_delta        CDATA
#IMPLIED >
```

```
<!ATTLIST TXN_PAGE_PERFORMANCE remain_packets_delta        CDATA
#IMPLIED >


<!ELEMENT TXN_PAGE_OBJECT EMPTY >
<!ATTLIST TXN_PAGE_OBJECT element_count                CDATA #REQUIRED >
<!ATTLIST TXN_PAGE_OBJECT redir_delta          CDATA #IMPLIED >
<!ATTLIST TXN_PAGE_OBJECT redir_count          CDATA #IMPLIED >
<!ATTLIST TXN_PAGE_OBJECT page_bytes           CDATA #REQUIRED >


<!ELEMENT TXN_PAGE_STATUS EMPTY >
<!ATTLIST TXN_PAGE_STATUS content_errors                CDATA #REQUIRED >
<!ATTLIST TXN_PAGE_STATUS error_code           CDATA #IMPLIED >


<!ELEMENT TXN_DETAIL_PERFORMANCE EMPTY >
<!ATTLIST TXN_DETAIL_PERFORMANCE start_msec                CDATA
#IMPLIED >
<!ATTLIST TXN_DETAIL_PERFORMANCE system_delta                CDATA
#IMPLIED >
<!ATTLIST TXN_DETAIL_PERFORMANCE connect_delta                CDATA
#IMPLIED >
<!ATTLIST TXN_DETAIL_PERFORMANCE dns_delta          CDATA #IMPLIED >
<!ATTLIST TXN_DETAIL_PERFORMANCE element_delta                CDATA
#IMPLIED >
<!ATTLIST TXN_DETAIL_PERFORMANCE first_packet_delta CDATA #IMPLIED >
<!ATTLIST TXN_DETAIL_PERFORMANCE request_delta                CDATA
#IMPLIED >
<!ATTLIST TXN_DETAIL_PERFORMANCE remain_packets_delta        CDATA
#IMPLIED >
<!ATTLIST TXN_DETAIL_PERFORMANCE ssl_handshake_delta        CDATA
#IMPLIED >


<!ELEMENT TXN_DETAIL_OBJECT EMPTY >
<!ATTLIST TXN_DETAIL_OBJECT conn_string_text    CDATA #IMPLIED >
```

```
<!ATTLIST TXN_DETAIL_OBJECT content_type          CDATA #IMPLIED >
<!ATTLIST TXN_DETAIL_OBJECT content_bytes         CDATA #IMPLIED >
<!ATTLIST TXN_DETAIL_OBJECT header_bytes          CDATA #IMPLIED >
<!ATTLIST TXN_DETAIL_OBJECT header_code           CDATA #IMPLIED >
<!ATTLIST TXN_DETAIL_OBJECT request_bytes         CDATA #IMPLIED >
<!ATTLIST TXN_DETAIL_OBJECT ip_address       CDATA #IMPLIED >
<!ATTLIST TXN_DETAIL_OBJECT object_text      CDATA #IMPLIED >


<!ELEMENT TXN_DETAIL_STATUS EMPTY >
<!ATTLIST TXN_DETAIL_STATUS error_code       CDATA #IMPLIED >
<!ATTLIST TXN_DETAIL_STATUS status_code           CDATA #IMPLIED >


<!ELEMENT TXN_PAGE_DETAILS ( TXN_DETAIL_ERROR | TXN_REDIRECT |
TXN_BASE_PAGE | TXN_PAGE_ELEMENT )*          >
<!ATTLIST TXN_PAGE_DETAILS page                   CDATA #IMPLIED >


<!ELEMENT TXN_DETAIL_ERROR EMPTY >
<!ATTLIST TXN_DETAIL_ERROR record_seq        CDATA #IMPLIED >
<!ATTLIST TXN_DETAIL_ERROR url               CDATA #IMPLIED >
<!ATTLIST TXN_DETAIL_ERROR title             CDATA #IMPLIED >
<!ATTLIST TXN_DETAIL_ERROR html_element_text  CDATA #IMPLIED >
<!ATTLIST TXN_DETAIL_ERROR error_text        CDATA #IMPLIED >


<!ELEMENT TXN_REDIRECT
(TXN_DETAIL_PERFORMANCE,TXN_DETAIL_OBJECT,TXN_DETAIL_STATUS) >
<!ATTLIST TXN_REDIRECT record_seq                 CDATA #REQUIRED >
<!ATTLIST TXN_REDIRECT record_subseq         CDATA #REQUIRED >
```

```
<!ELEMENT TXN_BASE_PAGE
(TXN_DETAIL_PERFORMANCE,TXN_DETAIL_OBJECT,TXN_DETAIL_STATUS) >
<!ATTLIST TXN_BASE_PAGE record_seq          CDATA #REQUIRED >



<!ELEMENT TXN_PAGE_ELEMENT
(TXN_DETAIL_PERFORMANCE,TXN_DETAIL_OBJECT,TXN_DETAIL_STATUS) >
<!ATTLIST TXN_PAGE_ELEMENT record_seq          CDATA #REQUIRED >
```

Output Response Samples

This section contains response sample output from Transaction Perspective/Application Perspective and Mobile Web Perspective.

Transaction Perspective/Application Perspective Output Response Sample

The sample XML document shown below illustrates the RESPONSE payload of a Transaction Perspective or Application Perspective request:

```
<?xml version="1.0"?>
<!DOCTYPE DP_RESPONSE SYSTEM "http://Data Pulse.keynote.com/dps/config/dp-
txn-response.dtd">
<DP_RESPONSE>
 <dps_status>
 <status request_id="1">
 <code>12</code>
 <severity>1</severity>
 <msg>Payload Summary</msg>
 <description>Your payload contains 3 slot(s), 8 measurement(s), 3 slot preference(s) and
22 agent preference(s).</description>
 <args>
 <argname>SLOT_PREFERENCE_COUNT</argname>
 <argvalue>3</argvalue>
 <argname>SLOT_COUNT</argname>
 <argvalue>3</argvalue>
 <argname>MEASUREMENT_COUNT</argname>
 <argvalue>8</argvalue>
```

    <argname>AGENT_PREFERENCE_COUNT</argname>
    <argvalue>22</argvalue>
    </args>
    </status>
    <status request_id="0">
    <code>9</code>
    <severity>1</severity>
    <msg>Request Summary</msg>
    <description>Successfully processed 1 out of 1 requests with 0 error(s), 0 warning(s) and 1 note(s).</description>
    <args>
    <argname>NOTES</argname>
    <argvalue>1</argvalue>
    <argname>SUCCESS</argname>
    <argvalue>1</argvalue>
    <argname>TOTAL</argname>
    <argvalue>1</argvalue>
    <argname>ERRORS</argname>
    <argvalue>0</argvalue>
    <argname>WARNINGS</argname>
    <argvalue>0</argvalue>
    </args>
    </status>
    </dps_status>
    <TXN_META_DATA agreement_id="110923">
    <AGENT_META_DATA agent_id="36939" description="Boston - Transaction v4 Boston UUnet" weight="1.0" ip="208.254.44.43" backbone="UUNET" instance_id="37301"/>
    <AGENT_META_DATA agent_id="36939" description="Boston - Transaction v4 Boston UUnet" weight="1.0" ip="208.254.44.50" backbone="UUNET" instance_id="36940"/>
    <AGENT_META_DATA agent_id="36941" description="Chicago - Transaction v4 Chicago UUNET" weight="1.0" ip="65.216.70.106" backbone="UUNET" instance_id="37302"/>
    <AGENT_META_DATA agent_id="36941" description="Chicago - Transaction v4 Chicago UUNET" weight="1.0" ip="65.216.70.113" backbone="UUNET" instance_id="36942"/>
    <AGENT_META_DATA agent_id="36943" description="Dallas - Transaction v4 Dallas UUNET" weight="1.0" ip="65.194.51.239" backbone="UUNET" instance_id="37303"/>
    <AGENT_META_DATA agent_id="36943" description="Dallas - Transaction v4 Dallas UUNET" weight="1.0" ip="65.194.51.246" backbone="UUNET" instance_id="36944"/>
    <AGENT_META_DATA agent_id="36945" description="Detroit - Transaction v4 Detroit Level3" weight="1.0" ip="63.211.22.78" backbone="LEVEL3" instance_id="37304"/>
    <AGENT_META_DATA agent_id="36945" description="Detroit - Transaction v4 Detroit Level3" weight="1.0" ip="63.211.22.85" backbone="LEVEL3" instance_id="36978"/>

Meta-data begins ►

<AGENT_META_DATA agent_id="36946" description="Houston - Transaction v4 Houston UUNET" weight="1.0" ip="63.123.70.41" backbone="UUNET" instance_id="37305"/>
 <AGENT_META_DATA agent_id="36946" description="Houston - Transaction v4 Houston UUNET" weight="1.0" ip="63.123.70.48" backbone="UUNET" instance_id="36947"/>
 <AGENT_META_DATA agent_id="36948" description="Los Angeles - Transaction v4 Los Angeles Qwest" weight="1.0" ip="63.150.145.171" backbone="QWEST" instance_id="37306"/>
 <AGENT_META_DATA agent_id="36948" description="Los Angeles - Transaction v4 Los Angeles Qwest" weight="1.0" ip="63.150.145.178" backbone="QWEST" instance_id="36949"/>
 <AGENT_META_DATA agent_id="36950" description="New York - Transaction v4 New York AT&T" weight="1.0" ip="63.240.19.173" backbone="AT&T" instance_id="37307"/>
 <AGENT_META_DATA agent_id="36950" description="New York - Transaction v4 New York AT&T" weight="1.0" ip="63.240.19.180" backbone="AT&T" instance_id="36951"/>
 <AGENT_META_DATA agent_id="36952" description="Philadelphia - Transaction v4 Philadelphia Level3" weight="1.0" ip="64.157.236.170" backbone="LEVEL3" instance_id="37308"/>
 <AGENT_META_DATA agent_id="36952" description="Philadelphia - Transaction v4 Philadelphia Level3" weight="1.0" ip="64.157.236.177" backbone="LEVEL3" instance_id="36953"/>
 <AGENT_META_DATA agent_id="36954" description="San Francisco - Transaction v4 San Francisco Sprint" weight="1.0" ip="205.163.212.41" backbone="SPRINT" instance_id="37309"/>
 <AGENT_META_DATA agent_id="36954" description="San Francisco - Transaction v4 San Francisco Sprint" weight="1.0" ip="205.163.212.48" backbone="SPRINT" instance_id="36955"/>
 <AGENT_META_DATA agent_id="36956" description="Washington - Transaction v4 Washington DC AT&T" weight="1.0" ip="63.240.196.235" backbone="AT&T" instance_id="37310"/>
 <AGENT_META_DATA agent_id="36956" description="Washington - Transaction v4 Washington DC AT&T" weight="1.0" ip="63.240.196.242" backbone="AT&T" instance_id="36957"/>
 <AGENT_META_DATA agent_id="37322" description="Washington - Transaction v4 Washington DC Genuity" weight="1.0" ip="128.11.121.170" backbone="Undefined" instance_id="37324"/>
 <AGENT_META_DATA agent_id="37322" description="Washington - Transaction v4 Washington DC Genuity" weight="1.0" ip="128.11.121.177" backbone="Undefined" instance_id="37323"/>
 <SLOT_META_DATA slot_id="427373" slot_alias="Yahoo Transaction" url="Yahoo

<table>
<tr><td>

Measurement data begins

►

First measurement begins

►

</td><td>

Transaction" pages="5"/>
 <SLOT_META_DATA slot_id="428213" slot_alias="Keynote test for Leslie" url="Keynote test for Leslie" pages="1"/>
 <SLOT_META_DATA slot_id="428214" slot_alias="Infospace test for Leslie" url="Infospace test for Leslie" pages="1"/>
 </TXN_META_DATA>
 <DP_TXN_MEASUREMENTS>
 <TXN_MEASUREMENT slot="427373" agent="36954" agent_inst="36955" datetime="07/12/2002 23:43:38" target="466317">
 <TXN_SUMMARY delta_user_msec="5107" delta_msec="4656" content_errors="0"/>
 <TXN_PAGE page_seq="1">
 <TXN_PAGE_PERFORMANCE start_msec="0" dns_lookup_msec="7" first_byte_msec="351" system_delta="2" connect_delta="78" first_packet_delta="85" request_delta="0" remain_packets_delta="562" delta_msec="734" delta_user_msec="871"/>
 <TXN_PAGE_OBJECT element_count="16" page_bytes="60391"/>
 <TXN_PAGE_STATUS content_errors="0"/>
 </TXN_PAGE>
 <TXN_PAGE page_seq="2">
 <TXN_PAGE_PERFORMANCE start_msec="3860" first_byte_msec="20" system_delta="6" connect_delta="76" first_packet_delta="95" request_delta="0" remain_packets_delta="10" delta_msec="506" delta_user_msec="651"/>
 <TXN_PAGE_OBJECT element_count="7" redir_delta="241" redir_count="1" page_bytes="14512"/>
 <TXN_PAGE_STATUS content_errors="0"/>
 <TXN_PAGE_DETAILS>
 <TXN_REDIRECT record_seq="1" record_subseq="1">
 <TXN_DETAIL_PERFORMANCE start_msec="0" system_delta="1" connect_delta="76" element_delta="153" first_packet_delta="76" request_delta="0" remain_packets_delta="0"/>
 <TXN_DETAIL_OBJECT conn_string_text="http://www.yahoo.com" content_type="UNKNOWN" content_bytes="0" header_bytes="87" request_bytes="359" ip_address="64.58.76.224" object_text="/r/in"/>
 <TXN_DETAIL_STATUS status_code="399"/>
 </TXN_REDIRECT>
 <TXN_BASE_PAGE record_seq="1">
 <TXN_DETAIL_PERFORMANCE start_msec="156" system_delta="6" connect_delta="77" dns_delta="87" element_delta="428" first_packet_delta="95" request_delta="0" remain_packets_delta="10"/>
 <TXN_DETAIL_OBJECT conn_string_text="http://dir.yahoo.com" content_type="text/html" content_bytes="3580" header_bytes="509" request_bytes="387" ip_address="64.58.76.81" object_text="/Computers_and_Internet/Internet/"/>
 <TXN_DETAIL_STATUS status_code="200"/>

</td></tr>
</table>

```
</TXN_BASE_PAGE>
</TXN_PAGE_DETAILS>
</TXN_PAGE>
<TXN_PAGE page_seq="3">
<TXN_PAGE_PERFORMANCE start_msec="7240" first_byte_msec="10"
system_delta="2" connect_delta="77" first_packet_delta="100" request_delta="0"
remain_packets_delta="217" delta_msec="435" delta_user_msec="450"/>
<TXN_PAGE_OBJECT element_count="9" page_bytes="16226"/>
<TXN_PAGE_STATUS content_errors="0"/>
</TXN_PAGE>
<TXN_PAGE page_seq="4">
<TXN_PAGE_PERFORMANCE start_msec="10249" dns_lookup_msec="1002"
first_byte_msec="20" system_delta="3" connect_delta="77" first_packet_delta="208"
request_delta="0" remain_packets_delta="424" delta_msec="2068"
delta_user_msec="2193"/>
<TXN_PAGE_OBJECT element_count="13" page_bytes="100910"/>
<TXN_PAGE_STATUS content_errors="0"/>
</TXN_PAGE>
<TXN_PAGE page_seq="5">
<TXN_PAGE_PERFORMANCE start_msec="15270" dns_lookup_msec="63"
first_byte_msec="20" system_delta="10" connect_delta="76" first_packet_delta="49"
request_delta="0" remain_packets_delta="215" delta_msec="912"
delta_user_msec="942"/>
<TXN_PAGE_OBJECT element_count="23" redir_delta="115" redir_count="1"
page_bytes="95459"/>
<TXN_PAGE_STATUS content_errors="0"/>
<TXN_PAGE_DETAILS>
<TXN_REDIRECT record_seq="1" record_subseq="1">
<TXN_DETAIL_PERFORMANCE start_msec="0" system_delta="1" connect_delta="76"
dns_delta="63" element_delta="218" first_packet_delta="78" request_delta="0"
remain_packets_delta="0"/>
<TXN_DETAIL_OBJECT conn_string_text="http://rd.yahoo.com"
content_type="UNKNOWN" content_bytes="0" header_bytes="667" request_bytes="786"
ip_address="64.58.77.87"
object_text="/O=1/search/iy/shop_cat/router/*http://advanced.search.shopping.yahoo.com
/shop/crouter?y=mf&amp;amp;dports_from=&amp;amp;dports_to=&amp;amp;aports_from
=&amp;amp;aports_to=&amp;amp;minprice_from=&amp;amp;minprice_to=&amp;amp;t=b
rowse"/>
<TXN_DETAIL_STATUS status_code="399"/>
</TXN_REDIRECT>
<TXN_BASE_PAGE record_seq="1">
<TXN_DETAIL_PERFORMANCE start_msec="225" system_delta="10"
connect_delta="10" dns_delta="26" element_delta="528" first_packet_delta="49"
```

request_delta="0" remain_packets_delta="215"/>
 <TXN_DETAIL_OBJECT conn_string_text="http://advanced.search.shopping.yahoo.com"
content_type="text/html" content_bytes="62986" header_bytes="351"
request_bytes="781" ip_address="66.218.75.137"
object_text="/shop/crouter?y=mf&dports_from=&dports_to=&aports_from=
&aports_to=&minprice_from=&minprice_to=&t=browse"/>
 <TXN_DETAIL_STATUS status_code="200"/>
 </TXN_BASE_PAGE>
 </TXN_PAGE_DETAILS>
 </TXN_PAGE>
 </TXN_MEASUREMENT>

End ofFirst Measurement

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

<TXN_MEASUREMENT slot="428213" agent="36943" agent_inst="36944"
datetime="07/12/2002 23:45:18" target="467277">
 <TXN_SUMMARY delta_user_msec="1602" delta_msec="1263" content_errors="0"/>
 <TXN_PAGE page_seq="1">
 <TXN_PAGE_PERFORMANCE start_msec="0" dns_lookup_msec="6"
first_byte_msec="230" system_delta="2" connect_delta="57" first_packet_delta="61"
request_delta="0" remain_packets_delta="176" delta_msec="1263"
delta_user_msec="1602"/>
 <TXN_PAGE_OBJECT element_count="25" page_bytes="116967"/>
 <TXN_PAGE_STATUS content_errors="0"/>
 </TXN_PAGE>
 </TXN_MEASUREMENT>
 <TXN_MEASUREMENT slot="428214" agent="36939" agent_inst="36940"
datetime="07/12/2002 23:37:26" target="467278">
 <TXN_SUMMARY delta_user_msec="2133" delta_msec="2040" content_errors="0"/>
 <TXN_PAGE page_seq="1">
 <TXN_PAGE_PERFORMANCE start_msec="0" dns_lookup_msec="5"
first_byte_msec="480" system_delta="2" connect_delta="95" first_packet_delta="249"
request_delta="0" remain_packets_delta="1644" delta_msec="2040"
delta_user_msec="2133"/>
 <TXN_PAGE_OBJECT element_count="25" page_bytes="114599"/>
 <TXN_PAGE_STATUS content_errors="0"/>
 </TXN_PAGE>
 </TXN_MEASUREMENT>
 <TXN_MEASUREMENT slot="428214" agent="36948" agent_inst="36949"
datetime="07/12/2002 23:45:02" target="467278">
 <TXN_SUMMARY delta_user_msec="1372" delta_msec="1304" content_errors="0"/>
 <TXN_PAGE page_seq="1">
 <TXN_PAGE_PERFORMANCE start_msec="0" dns_lookup_msec="6"

```
              first_byte_msec="321" system_delta="3" connect_delta="33" first_packet_delta="120"
              request_delta="0" remain_packets_delta="1015" delta_msec="1304"
              delta_user_msec="1372"/>
               <TXN_PAGE_OBJECT element_count="25" page_bytes="108989"/>
               <TXN_PAGE_STATUS content_errors="0"/>
               </TXN_PAGE>
               </TXN_MEASUREMENT>
               <TXN_MEASUREMENT slot="428214" agent="36943" agent_inst="36944"
              datetime="07/12/2002 23:45:27" target="467278">
               <TXN_SUMMARY delta_user_msec="1942" delta_msec="1872" content_errors="0"/>
               <TXN_PAGE page_seq="1">
               <TXN_PAGE_PERFORMANCE start_msec="0" dns_lookup_msec="5"
              first_byte_msec="380" system_delta="64" connect_delta="69" first_packet_delta="176"
              request_delta="0" remain_packets_delta="1492" delta_msec="1872"
              delta_user_msec="1942"/>
               <TXN_PAGE_OBJECT element_count="26" page_bytes="111681"/>
               <TXN_PAGE_STATUS content_errors="0"/>
               </TXN_PAGE>
               </TXN_MEASUREMENT>
               </DP_TXN_MEASUREMENTS>
          </DP_RESPONSE>
```

Response Declarations

The list below shows the possible status conditions that can be returned in the response payload.

Note: the NAME of the response is not returned in the XML payload. Responses in the response payload are identified by code (CODE column below).

MESSAGE_NAME	Short Text	Code	Severity	Description	Properties
AGREEMENT_ID_INVALID	Invalid agreement ID	20	ERROR	Returned if agreement ID is malformed, non-numeric or negative	AGREEMENT_ID LOGIN_NAME
SERVICE_TYPE_UNKNOWN	Unknown service type	21	ERROR	Returned if the service type is unrecognized. Legal values are	SERVICE_TYPE AGREEMENT_ID

				TRANSACTION or [<<Get MWP]	
DELTA_TIME_MALFORMED	Malformed Delta Time	22	ERROR	Returned if delta-time is malformed, non-numeric or negative.	DELTA_TIME AGREEMENT _ID
DELTA_TIME_OUT_OF_RANGE	Delta time out of range	23	ERROR	Returned if delta-time is outside the legal range (e.g.: [30..900] secs)	AGREEMENT _ID, DELTA_TIME, DELTA_MIN, DELTA_MAX
LOGIN_UNKNOWN_USER	Unknown Login	40	ERROR	Returned when there is no login with the specified name	AGREEMENT _ID, LOGIN_NAME
LOGIN_INVALID_PASSWORD	Invalid password	41	ERROR	Returned when the specified password is incorrect	AGREEMENT _ID, LOGIN_NAME
LOGIN_ACCESS_DENIED	Login Access Denied	42	ERROR	Returned when the login does not have the proper security flag enabled	AGREEMENT _ID, LOGIN_NAME
LOGIN_TOO_MANY_REQUESTS	Too many login requests	43	ERROR	Returned when the login has made too many requests per minute	AGREEMENT _ID, LOGIN_NAME , SERVICE_TY PE, MAX_PER_MI NUTE
AGREEMENT_NOT_FOUND	Agreement not found	60	ERROR	Returned when the specified agreement was not found	AGREEMENT _ID
AGREEMENT_ACCESS_VIOLATION	Agreement Access Denied	61	ERROR	Returned if the specified agreement does	AGREEMENT _ID, LOGIN_NAME

				not belong to the same company as the authenticated login	
DATA_NOT_READY	Data not ready	80	WARNING	Returned when data caches have less data then the specified time range (occurs during restart)	DELTA_TIME, CURRENT_UPTIME_MAX
DATA_NOT_AVAILABLE	Data not found	81	WARNING	Returned when no data was found for the specified agreement and service type in the specified delta time range.	AGREEMENT_ID, DELTA_TIME, SERVICE_TYPE
SERVICE_NOT_AVAILABLE	Service is unavailable	82	ERROR	Returned if the backend servers are not responding or unavailable	AGREEMENT_ID

Measurement Reference

This chapter describes the measurement elements and attributes returned by Keynote Data Pulse.

Transaction Perspective and Application Perspective Measurements

This table describes the measurement data elements that are included in the Data Pulse response payload for Transaction Perspective and Application Perspective measurements.

ELEMENT: DP_TXN_MEASUREMENTS Consists of 0 or more TXN_MEASUREMENT elements.	
ELEMENT: TXN_MEASUREMENT Each instance represents a specific transaction measurement.	
ATTRIBUTES	
slot	Keynote identification for the URL being measured.
agent	Keynote identification for which agent took the measurement.
agent_inst	An internal Keynote identifier for the specific Agent computer taking the measurement.
datetime	This field identifies when the measurement was taken, the timezone is GMT.
target	The target group that the slot is part of.
ELEMENT: TXN_SUMMARY	

ATTRIBUTES	
delta_user_msec	The user experience full page time as reported by Internet Explorer.
delta_msec	The time to complete the full page, measured from the time the navigation started until the time the last resource was fetched. Because Internet Explorer uses multiple connections to fetch the resources, the sum of the times for download of all resources could be larger than the full-page download. This component does not include resources that are used by the page, but cached locally and thus not downloaded in the current session.
content_errors	The total number of content errors in the transaction. Content errors are errors while retrieving page elements and are non-fatal to the transaction.

ELEMENT: **TXN_ERROR**	
ATTRIBUTES	
code	Contains an error code for the first page in the transaction that has a page level error. The page's number is shown in Error_Page.
page	In the case of a page level error, this field contains the erroneous page's number. The error code is in Error_Code.

ELEMENT: **TXN_PAGE**	

ATTRIBUTES	
page_seq	Identifies which page of the transaction the data is associated with.

ELEMENT: **TXN_PAGE_PERFORMANCE**	
ATTRIBUTES	
delta_msec	The time to complete the full page, measured from the time the navigation started until the time the last resource was fetched. Because Internet Explorer uses multiple connections to fetch the resources, the sum of the times for download of all resources could be larger than the full-page download. This component does not include resources that are used by the page, but cached locally and thus not downloaded in the current session.
delta_user_msec	The user experience full page time as reported by Internet Explorer.
start_msec	The start time offset of a page from the beginning of download of the transaction. The first page off set is always 0.
dns_lookup_msec	The time spent resolving the DNS name to an IP address.
first_byte_msec	Once the connection is set up, the agent will request data from the server with an HTTP Get command. This Time to First Byte measurement is the amount of time from when the agent starts its initial connection setup until it sees the first byte

	back from the server.
system_delta	The total time spent fetching the resource, excluding the time spent for interaction with the network.
connect_delta	The connection time of the base page resource, if presented.
first_packet_delta	The time to first packet of the base page resource.
ssl_handshake_delta	The base page SSL time.
request_delta	The base page request time.
remain_packets_delta	Time to download the remaining packets after the first packet.

ELEMENT: **TXN_PAGE_OBJECT**

ATTRIBUTES

element_count	The number of elements in the page, including base page.
redir_delta	The total redirect time of the base page, if presented.
redir_count	The number or redirects of the base page, if presented.
page_bytes	Byte size of the content.

ELEMENT: **TXN_PAGE_STATUS**

ATTRIBUTES

content_errors	The total number of content errors in the transaction. Content errors are errors while retrieving page elements and are non-fatal to the transaction.
error_code	Contains an error code for the first page in the transaction that has a page level error.

ELEMENT**: TXN_DETAIL_PERFORMANCE**	
ATTRIBUTES	
start_msec	The start time offset of a page element from the beginning of download of the transaction. The first page element off set is always 0.
system_delta	The total time spent fetching the page element, excluding the time spent for interaction with the network.
connect_delta	The connection time of the element, if presented.
dns_delta	DNS time for the element.
element_delta	Total download time for a page element (much like measurement_network).
first_packet_delta	The time to first packet of the page element.
request_delta	The page element request time.
remain_packets_delta	The time to download the rest of the packets for the page element.
ssl_handshake_delta	The page element SSL time.

ELEMENT: **TXN_DETAIL_OBJECT**	
ATTRIBUTES	
conn_string_text	The page element's protocol, plus host name, plus port number.
content_type	The type of content (for example, Gif, Java, etc.) returned by the agent either as a type code or as a string as reported by the content.
content_bytes	Byte size of the content.
header_bytes	Byte size of the header.
header_code	Provides additional information about the HTML request/response, most frequently about whether the body is compressed.
request_bytes	Byte size of the requested call.
ip_address	IP address of the page or page element presented in the case of an error.
object_text	The page element's path and document name.
ELEMENT**: TXN_DETAIL_STATUS**	
ATTRIBUTES	
error_code	Contains an error code for the first page in the transaction that has a page level error. The page's number is shown in Error_Page.

status_code	Non-error information reported by the agent.

ELEMENT: **TXN_PAGE_DETAILS**	

ATTRIBUTES	

page	In the case of a page level error, this field contains the erroneous page's number. The error code is in ErrorCode.

ELEMENT: **TXN_DETAIL_ERROR**
Represents a transaction playback error record. This element is sent when there was an error that caused the playback engine to abort the transaction. This would usually be caused by changes to the page after recording the script.

ATTRIBUTES	

record_seq	The error's sequence id.
url	Location of page with error.
title	Title of the page.
html_element_text	Outer HTML text (For example: <input size=30 name=p>).
error_text	Page Error String (For example: Page Not found).

ELEMENT: **TXN_REDIRECT**
Represents a redirect detail record. Page-level redirects are sent whenever a redirect occurs. There can be zero or more redirects associated with each page of the transaction. There can also be redirect elements associated with page content (elements), however, these redirects are only sent when there is a content error detected on the page.

ATTRIBUTES	

record_seq*)*	The redirect's sequence id.
record_subseq	The redirection sequence of specific element. If the there is no redirection the value is 0.

ELEMENT: **TXN_BASE_PAGE** Represents a base page detail record. There will be exactly one of these elements present (per page). However, this element is only sent if a Content Error occurred on the page.

ATTRIBUTES

record_seq	The base page's sequence id.

ELEMENT: **TXN_PAGE_ELEMENT** Represents a page element detail record. There can be zero or more of these elements representing content on the page. However, these elements are only sent if there was at least one content error on the page.

ATTRIBUTES

record_seq	The page element's sequence id.