Bringing convenience to life.

NAMA VDI Compliant Standard Server-to-Server DEX Message Transfer Standard

VDI S2S-DEX

Version 1.1

National Automatic Merchandising Association
20 N. Wacker Drive, Chicago IL USA
www.vending.org

Initial Posting → November 2011

Table of Contents

Chapter 1 DEX Transfer Webservices

1.1 Background

Open system webservices are typically required to transmit DEX payloads from one system, be it a device or application or server, to another. A common use of these webservices is the push transmission of a DEX payload from a ***deployed telemetry unit, or server, or application*** to a backend enterprise route management system using an UploadDex Webservice. The payload document can also be used as a standalone file and be traded using FTP or any other transmission method. Another common use would be an enterprise route management system contacting another system to "pull" DEX files from another server or application using a GetDex Webservice.
In these Webservices, an attempt has been made to keep the payload XML as similar as possible. Other features include:

- Arguments that include the provider identification (NAMA identifier)
- Ability to include the customer identifier as a string
- Identification of the device or application transmitting the data
- Versioning
- Optional Compression, with ability to transmit compression meta data (UploadDex only)
- Metadata information including time stamp, result code, and GMT offset
- Transmission of a single record, multiple records in one transmission, or a large compressed payload of multiple records pulled over time, each with their own meta data
- Error Code upload allows transmission of metadata when DEX reads fail
- DexType element to define the type of service initiating the DEX upload (refill, cash-out, archive, current, etc.)

1.2 Authentication

The webservice provider will authenticate the webservice consumer against a username/password entered into SOAP header information (header element: authorization, basic base64 encoded username:password). The username/password is part of basic HTTP basic authentication. The SOAP header information is the same as standard HTTP username/password credentials passed as part of the transfer header

Example (within the HTTP header):

```
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
```

For example, the base64 above decodes to "username:password".

Refer to **http://www.w3.org/Protocols/HTTP/1.0/spec.html#BasicAA** for additional details.

Chapter 2 GetDex message

2.1 Purpose

A webservice that allows receiving of DEX payloads from Server or application system (webservice provider) by another Server or application system (webservice consumer).

VDIxml GetDex(parameters)

Note: VDIxml here and later refers to artificial data type, which is XML String according to VDI standard.

2.2 Parameters list

This call takes 12 parameters:

Name	**Type**	**Required**	**Options**	**Description**
TransactionID	String(16)	Yes		Unique request identifier from the device. Used to prevent 'echo' data if a retry is sent from the device to the Server. Server returns an error and rejects the message if a duplicate TransactionID is detected.
ProviderID	String(32)	Yes		Name of the provider of the source data, for example. inOne or MEI sending DEX.
CustomerID	String(32)	Yes		Name of the bottler or operator in which the data belongs, like BestFamilyVending.
ApplicationID	String(32)	Yes		Identifies the provider application which called UploadDex Web Method .
ApplicationVersion	String(8)	Yes		Version of the application which called UploadDex Web Method
DeviceList	Array of strings, each element String(16), enclosed in <Item> </Item> tags	No		A list of telemetry device IDs. If present the call will only return DEX collected by telemetry devices in the list. .
OutletList	Array of strings, each element String(16), enclosed in <Item> </Item> tags	No		A list of machine identifiers. If present the call will only return DEX collected at machine in the list.
OnOrAfter	Date/time	No		If provided, only DEX collected at or after the specified time can be returned.

OnOrBefore	Date/time	No		If provided, only DEX collected at or before the specified time can be returned.
ReturnSet	String(16)	Yes	FIRST <qty>, LAST <qty>, ALL	Determines how many DEX records should be returned for each device or machine. Valid values are: FIRST <qty>, LAST <qty>, ALL. The quantity parameter is optional and is assumed to be 1 if omitted
UserData	String	No		User defined data, if any.
VDIXMLVersion	String(8)	Yes		Version of VDIXML to send back.

2.3 Example Message

Following is an example SOAP 1.2 message body:

```
<ws:GetDex xmlns:ws="urn:ExampleVDIService">
        <TransactionID>LIYaSha0nEqhSBLy</TransactionID>
        <ProviderID>ExampleProvider</ProviderID>
        <CustomerID>ExampleCustomerID</CustomerID>
        <ApplicationID>ExampleApp</ApplicationID>
        <ApplicationVersion>1.2.33.44</ApplicationVersion>
        <VDIXMLVersion>1.1</VDIXMLVersion>
        <ReturnSet>LAST</ReturnSet>
        <DeviceList enc:id="DeviceList"
            xmlns:enc="http://www.w3.org/2003/05/soap-encoding"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            enc:itemType="xsd:string" enc:arraySize="2">
                <Item>DeviceID1</Item>
                <Item>DeviceID2</Item>
        </DeviceList>
</ws:GetDex>
```

2.4 Return

GetDex message returns an XML string that contains a VDIReturn block and a collection of DEX records according to the values of the GetDex parameters. If neither DeviceList nor OutletList are specified, the server will return DEX records for all data collection points (points of sale) that correspond to the specified CustomerID and ProviderID. If no time range is specified (that is, OnOrBefore and OnOrAfter are both omitted), the server will return only DEX records for the last 48 hours.

Example return XML:

Following is an example GetDex return message:

```
<?xml version="1.0" encoding="utf-8"?>
<VDITransaction VDIXMLVersion="1.1"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   TransactionReason="GetDEX"
   TransactionID="2byBm1f1MkC7oZyz"
   TransactionTime="2011-07-28T12:34:56"
   ProviderID="ExampleProvider"
   CustomerID="ExampleCustomerID"
   ApplicationID="ExampleApp"
   ApplicationVersion="1.2.33.4">
      <VDIReturn>
            <Code>0</Code>
            <Message>Success</Message>
      </VDIReturn>
      <DEXList DEXEncoding="0"
         DEXCompressionType="" DEXCompressionParam="">
            <DexTransmission DeviceID="DeviceID1"
               TransmitTime="2011-07-28T11:17:04" GMTOffSet="-5">
                  <DexCollection>
                      <DEX ReadDateTime="2011-07-28T16:08:00"
                         GMTOffSet="0" DexReason="0" DexType="0"
                         ResponseCode="OK">
                            <RawDEX>
                                DXS*9259630001*VA*V1/1*1**100
                                DXE*1*1
                            </RawDEX>
                      </DEX>
                  </DexCollection>
            </DexTransmission>
            <DexTransmission DeviceID="DeviceID2"
               TransmitTime="2011-07-28T11:18:36" GMTOffSet="-5">
                  <DexCollection>
                      <DEX ReadDateTime="2011-07-28T16:08:00"
                         GMTOffSet="0" DexReason="0" DexType="0"
                         ResponseCode="OK">
                            <RawDEX>
                                DXS*9253420001*VA*V1/1*1**100
                                DXE*1*1
                            </RawDEX>
                      </DEX>
                  </DexCollection>
            </DexTransmission>
      </DEXList>
</VDITransaction>
```

Chapter 3 UploadDex message

3.1 Purpose

A webservice that allows the transmission of DEX payloads from device or application system (webservice consumer) to another application system serving as the host for the webservice (webservice provider).

XML string UploadDex(parameters)

3.2 Parameters list

This call takes 11 parameters.

Name	Type	Required	Options	Description
TransactionID	String(16)	Yes		Unique request identifier from the device. Used to prevent 'echo' data if a retry is sent from the device to the Server. Server returns an error and rejects the message if a duplicate TransactionID is detected.
ProviderID	String(32)	Yes		Name of the provider of the source data, for example inOne or MEI sending DEX.
CustomerID	String(32)	Yes		Name of the bottler or operator in which the data belongs, like BestFamilyVending.
ApplicationID	String(32)	Yes		Identifies the provider application which called UploadDex Web Method.
ApplicationVersion	String(8)	Yes		Version of the application which called UploaDex Web Method
DEXEncoding	Integer	Yes	0 – 'None' 1 – 'ISO8859-1' 2 – 'UTF-8'	Type of encoding used to encode DEX files. Can be only ISO8859-1 or UTF-8.
DEXCompressionType	String(32)	No	0 – 'None', or implementation-specific code	Type of Compression used. Can be omitted or 'NONE' to flag that no compression was used.
DEXCompressionParam	String	No		Additional compression information if anything required for decompression.
UserData	String	No		User defined data, if any.
VDIXMLVersion	String(8)	Yes		Version of VDIXML sent as VDIXML parameter (1.0 initially)
VDIXML	XML String	Yes		XML String representing collection of DEX Files with additional attributes. See format below.

3.3 UploadDEX XML

VDIXMLVersion – This is a version of VDIXML. If VDIXML format will ever change VDIXML parsers would need to be changed also. It's unrealistic to think that this can be or will be done by whole industry in the same time. To prevent this situation VDIXMLVersion would identify what parser (version of VDIXML) to use.

VDIXML - XML String representing collection of DEX Files with additional attributes as in the following SOAP 1.2 example:

```
<?xml version="1.0" encoding="utf-8"?>
<VDITransaction VDIXMLVersion="1.1"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   TransactionReason="UploadDEX"
   TransactionID="2bC7oZyzyBm1f1Mk"
   TransactionTime="2011-07-28T12:42:42"
   ProviderID="ExampleProvider"
   CustomerID="ExampleCustomerID"
   ApplicationID="ExampleApp"
   ApplicationVersion="1.2.33.4">
      <DEXList DEXEncoding="0"
         DEXCompressionType="" DEXCompressionParam="">
            <DexTransmission DeviceID="DeviceID1"
               TransmitTime="2011-07-28T11:17:42" GMTOffSet="-5">
                  <DexCollection>
                      <DEX ReadDateTime="2011-07-28T16:08:00"
                         GMTOffSet="0" DexReason="0" DexType="0"
                         ResponseCode="OK">
                            <RawDEX>
                                DXS*9259630001*VA*V1/1*1**100
                                DXE*1*1
                            </RawDEX>
                      </DEX>
                  </DexCollection>
            </DexTransmission>
            <DexTransmission DeviceID="DeviceID2"
               TransmitTime="2011-07-28T11:18:42" GMTOffSet="-5">
                  <DexCollection>
                      <DEX ReadDateTime="2011-07-28T16:08:00"
                         GMTOffSet="0" DexReason="0" DexType="0"
                         ResponseCode="OK">
                            <RawDEX>
                                DXS*9253420001*VA*V1/1*1**100
                                DXE*1*1
                            </RawDEX>
                      </DEX>
                  </DexCollection>
            </DexTransmission>
      </DEXList>
</VDITransaction>
```

3.4 Return

UploadDEX message returns an XML string in the standard VDI return structure, for example:

```
<?xml version="1.0" encoding="utf-8"?>
<VDITransaction VDIXMLVersion="1.1">
      <VDIReturn>
            <Code>0</Code>
            <Message>Success</Message>
     </VDIReturn>
</VDITransaction>
```

Chapter 4 XML Tags and Attributes Descriptions

4.1 Tags Descriptions

4.1.1 Common Tags

These tags apply to all VDI messages:

Name	Type	Required	Options	Description
<VDITransaction>		Yes		This is the Root Element for VDI transaction. All other elements are nested within it
<UserData>	String	No		This is an optional element within different levels of VDIXML and represents provider specific data.
<OtherCollectionsOrLists>	Collection	No		Just a place holder to show that other elements (like alarms, etc…) can go here.

4.1.2 DEX Tags

These tags apply to both GetDEX and UploadDEX messages.

Name	Type	Required	Options	Description
<DEXList>	Collection	No		This element is start for DEX related data transmission.
<DexTransmission>	Collection	No		This is an element within <DEXList>. Dex Transmission combines Device information and collection of DEX reads for this Device transmitted from the same device in the same time. In some cases Device may read DEX several times and not being able to send it individually. All DEX reads not sent yet are being combined into DEX transmission. <DexList> may have multiple <DexTransmissions> collections for the same or different devices. Context

			of each individual <DexTransmission> will be compressed based on DEXEncoding and DEXCompressionParam parameters of UploadDex method.
<DexCollection>	Collection	No	This is an element within <DexTransmission> and represents combination of dex records and their specific information for the same device, read in the different time, but being transmitted in the same time. Each <DEXCollection> may combine information about multiple DEX files.
<Dex>	Collection	No	This is an element within <DexCollection>. Each <DEX> collection combines information about single DEX file and has DEX read itself.
<RawDEX>	String	No	This is an element within <DEX> and represents **Encoded** Raw Dex. (for simplicity DEX is not encoded in XML sample above). RAWDEX context is going to be encoded based on DEXEncoding parameter of UploadDEX method or DEXEncoding attribute of DEXList element.

4.2 Attributes Description

4.2.1 Common Attributes

These tags apply to all VDI messages.

Tag Name	**Attribute Name**	**Type**	**Required**	**Description**
VDITransaction	VDIXMLVersion	String(8)	Yes	Version of VDIXML used in XML
VDITransaction	TransactionReason	String(32)	Yes	Reason for transaction. It can be a message type, e.g. “GetDEX”, “UploadDEX”, or any other reason for this particular transaction
VDITransaction	TransactionID	String(16)	Yes	Unique identifier for transaction. Used to prevent transmitting or processing of the same transaction.
VDITransaction	TransactionTime	Date/Time	Yes	Time when this transaction was transmitted
VDITransaction	ProviderID	String(32)	Yes	Name of the provider of the source data, for example inOne or MEI sending DEX.
VDITransaction	CustomerID	String(32)	Yes	Name of the bottler or operator in which the data belongs, like BestFamilyVending.
VDITransaction	ApplicationID	String(32)	Yes	Identifies the provider application which created transaction
VDITransaction	ApplicationVersion	String(8)	Yes	Version of the application which created transaction

4.2.2 DEX Attributes

These attributes apply to both GetDEX and UploadDEX messages.

Tag Name	Attribute Name	Type	Required	Description
DEXList	RecordsCount	Integer	No	RecordsCount represents a number of transmissions within the list.
DEXList	DEXEncoding	Integer	Yes	Type of encoding used to encode DEX files. Can be only ISO8859-1 or UTF-8.
DEXList	DEXCompressionType	String(32)	No	Type of Compression used. Can be omitted or 'NONE' to flag that no compression was used. For Now only one type is supported: 0 - 'None'
DEXList	DEXCompressionParam	String	No	Additional compression information if anything required for decompression.
DexTransmission	DeviceID	String(32)	Yes	ID of the device that this <DEXTransmission> is for.
DexTransmission	TransmitTime	Date/Time	Yes	Date/time of this particular transmission. Value is local date/time of transmission.
DexTransmission	GMTOffSet	Integer	Yes	offset between local time of transmission place and GMT (LocalTime – GMTTime)
DexCollection	RecordsCount	Integer	No	RecordsCount attribute represents a number of DEX files within the DexCollection.
DEX	ReadDateTime	Date/Time	Yes	Date/time when Dex was read. This is device local Date/Time.
DEX	GMTOffSet	Integer	Yes	Offset between device local time and GMT (LocalTime – GMTTime)
DEX	FileSize	Long	No	Size of DEX file in bytes
DEX	DexReason	Integer	Yes	During what user activity (if any) DEX were taken. Possible values currently defined as: Refill, Service, Cash Out, Archive. List of values can be extended with new types. 0 – Scheduled DEX Read 1 – Service Button Pressed 2 – Door Open 3 – Door Closed 4 – Fulfills Specific Request 5 – VMC initiated 6 – Dex from external Device (Driver HH or similar)

				99 – Other
DEX	DexType	Integer	Yes	‘Full’ dex was sent/taken or only specific portion of DEX stream. 0 – Full Dex 99 – Other
DEX	ResponseCode	String(128)	Yes	Status of DEX read. If DEX read is successful then value should be OK. In case of unsuccessful read, value will be text of the error.

Alpha Listing -- VDI TASK FORCE MEMBERSHIP --- November 2011

Chair – Chris Lilly, Best Vendors Technology Group
Coordinator – Michael Kasavana, Michigan State University/NAMA
NAMA Board Liaison – Mark Stein, Mark Vend

1-Anant Agrawal (Cantaloupe Systems)
2-Mandeep Arora (Cantaloupe Systems)
3-Tammy Baker (Cantaloupe Systems)
4-Louis Beaudoin (Cantaloupe Systems)
5-Rebecca Boyle (Apriva)
6-Glenn Butler (CTO Services)
7-Jim Canter (Crane)
8-Jeff Doerr (Flextronics)
9-Don Finley (MEI Group)
10-Justin Grant (Cantaloupe Systems)
11-Doug Haddon (MEI Group)
12-Dennis Hammer (Seaga Manufacturing)
13-Ron Hoormann (CoinCo)
14-Tom Howell (Coca-Cola)
15-Tenoah Hunt (Coca-Cola)
16-Glen Johnson (Vendors Exchange)
17-Michael Kasavana (Michigan State University)
18-Mark Kronenberg (CompuVend)
19-Chris Lilly (Best Vendors Technology Group)
20-Dan Mathews (NAMA)
21-Jeff Mayoros (Wittern)
22-Steve Merwarth (Coca-Cola)
23-Bud Nixon (Compass-USA)
24-Chris Norris (Validata)
25-Gene Ostendorf (InOne Technology)
26-Paresh Patel (Courtesy Vending)
27-Warren Philips (Validata)
28-Anton Rakushkin (Crane-Streamware)
29-Mary Rampe (MEI Group)
30-Bill Robertson (Apriva)
31-Cary Sagady (USA Technologies)
32-Mark Stein (Mark Vend)
33-David Saracini (Sprout Retail)
34-Leandro Valdez (Apriva)
35-Bob Williams (Compass-USA)