Delta Record Management Object

Approved Version 1.0 – 21 May 2013

Open Mobile Alliance

OMA-ER-DeltaRecMO-V1_0-20130521-A

Contents

1. Scope

This document describes the Delta Record MO employed in Delta Record management that builds on and leverages the OMA DM v1.3 protocol [DMPRO]. It provides a standard DM Management Object, which is the interface between a DM Client and Server. This document also describes the associated client-side and server-side behaviour necessary to use the Delta Record feature.

2. References

2.1 Normative References

[DMARCH] "OMA Device Management Architecture, Version 1.3", Open Mobile Alliance™. OMA-AD-DM-V1_3,
URL: http://www.openmobilealliance.org/

[DMDICT] "OMA Device Management Dictionary, Version 1.3", Open Mobile Alliance™. OMA-SUP-DM_Dictionary-V1_0.
URL: http://www.openmobilealliance.org/

[DMPRO] "OMA Device Management Protocol, Version 1.3". Open Mobile Alliance™.
OMA-TS-DM-Protocol-V1.3,
URL: http://www.openmobilealliance.org/

[DMXSDDeltaRec] "OMA Device Management Delta Record XSD, Version 1.0". Open Mobile Alliance™. OMA-SUP-XSD_DeltaRec_DeltaRecord-V1_0.
URL: http://www.openmobilealliance.org/

[IOPPROC] "OMA Interoperability Policy and Process", Version 1.1, Open Mobile Alliance™, OMA-IOP-Process-V1_1,
URL: http://www.openmobilealliance.org/

[OMA-DM] *OMA Device Management, Version 1.2*,
Open Mobile Alliance™,
URL: http://www.openmobilealliance.org/

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997,
URL: http://www.ietf.org/rfc/rfc2119.txt

[RFC4234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. October 2005,
URL: http://www.ietf.org/rfc/rfc4234.txt

2.2 Informative References

[ACMO] "White Paper on Provisioning Objects", Open Mobile Alliance™, OMA-WP-AC_MO
URL: http://www.openmobilealliance.org

[DMSTDOBJ] "OMA Device Management Standardized Objects, Version 1.3". Open Mobile Alliance™. OMA-TS-DM-StdObj-V1_3.
URL: http://www.openmobilealliance.org

[RFC2616] "Hypertext Transfer Protocol – HTTP/1.1". Network Working group. June 1999.
URL: http://www.ietf.org/rfc/rfc2616.txt

[RFC3986] "Uniform Resource Identifier (URI): Generic Syntax". Network Working Group. January 2005.
URL: http://www.ietf.org/rfc/rfc3986.txt

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

Delta Record	Contains information showing a modification applied to a specific node in an OMA DM Client Management Object.
Device Management	Management of the Device configuration and Management Objects from the point of view of the various Management Authorities. Device Management includes - but is not restricted to - setting initial configuration information in Devices, subsequent updates of persistent information in Devices, retrieval of management information from Devices, and processing events and alarms generated by Devices.
Device Management Server	A network based entity that issues OMA DM commands to devices and correctly interprets responses sent from the devices.
DM Tree	A Device Management Tree is a collection of Management Objects [DMDICT]. See also Management Tree [DMDICT].

3.3 Abbreviations

DDF	Device Description Framework
DM	Device Management
MO	Management Object
OMA	Open Mobile Alliance
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

4. Introduction

As DM Trees contain more managed objects that represent characteristics of a mobile device software and hardware, retrieving the complete DM Tree MOs over-the-air will be a time-consuming and costly activity. It is also necessary from time to time for the DM Client and Server to synchronize their DM Trees for other management functions, such as software downloading and customer care support, to work effectively. Delta Records allow the DM Server to track changes applied to the Client DM Tree MOs. These Delta Records can be used to analyze the state of the Client configuration and synchronize the DM Tree information contained on the Client and Server.

When this feature is enabled, all modifications of the DM Tree, including structure and data values, MUST be recorded and stored on the device in the form of Delta Records, which can be retrieved by a DM Server. Delta Records provide a history of changes applied to a Client DM Tree, including information on the time and authority that enacted the change, as well as the change itself. Delta Records MUST contain a sequence number, timestamp, and authority identifier, which together create a unique identifier per Delta Record.

This feature records in Delta Records the adds, deletes, and changes applied to the Client DM Tree. For management of Delta Records, the ability to retrieve, purge, and asynchronously send Delta Records to a DM Server are defined via the DeltaRec MO.

The following diagram shows the flow of messages and events related to Delta Records:

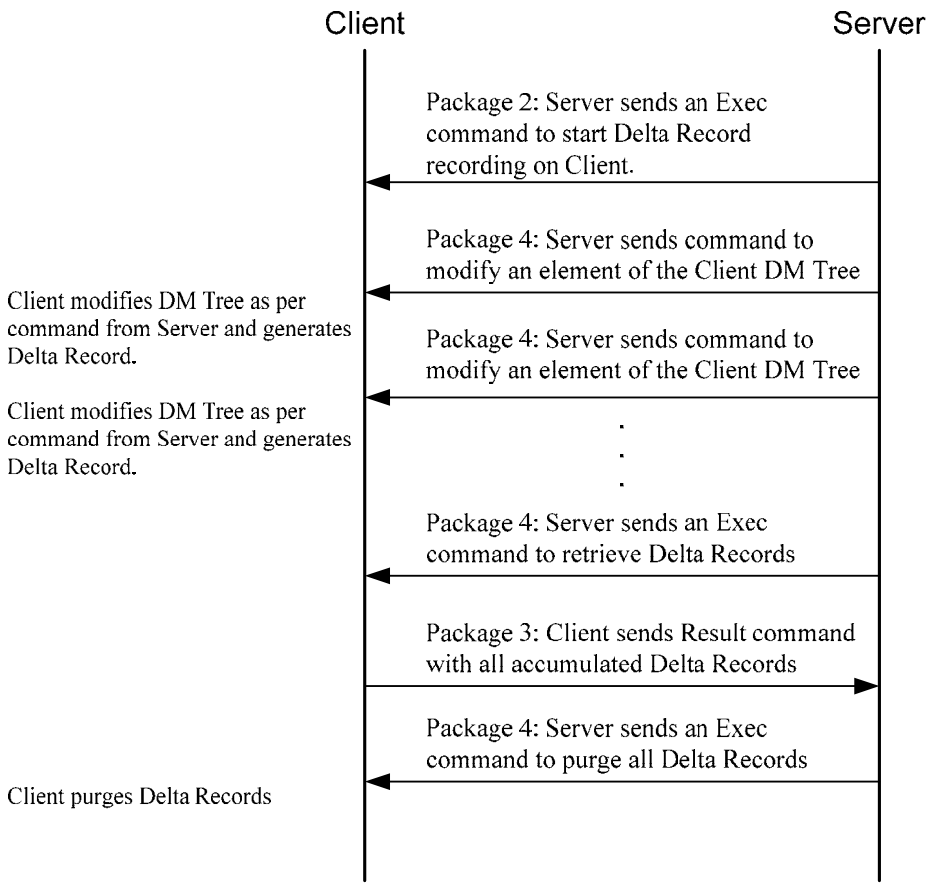


Figure 1: Message flows for Delta Record generation, retrieval, and purging

5. Requirements (Normative)

5.1 High-Level Functional Requirements

Label	Description	Release
DeltaRec-HLF-1	The Delta Record MO enabler SHALL record changes applied to a Client DM Tree.	1.0
DeltaRec-HLF-2	The Delta Record MO enabler SHALL provide the DM Client and DM Server with the ability to start and stop the recording of Delta Records on the device.	1.0
DeltaRec-HLF-3	The Delta Record MO enabler SHALL provide the ability for a DM Server to retrieve Delta Records from the device.	1.0
DeltaRec-HLF-4	The Delta Record MO enabler SHALL provide the DM Client and DM Server with the ability to purge Delta Records on the device.	1.0
DeltaRec-HLF-5	The Delta Record MO enabler SHALL provide the DM Client with the ability to send Delta Records asynchronously to the DM Server.	1.0
DeltaRec-HLF-6	A Delta Record SHALL include the time the recorded change was made to the Client DM Tree.	1.0
DeltaRec-HLF-7	A Delta Record SHALL include information identifying the authority that enacted the change on the Client DM Tree.	1.0
DeltaRec-HLF-8	A Delta Record SHALL include a sequence number from a sequence number series associated with each DM Server.	1.0
DeltaRec-HLF-9	A Delta Record MO Enabler SHALL provide the ability to specify a maximum number of Delta Records stored by the device.	1.0
DeltaRec-HLF-10	A Delta Record MO Enabler SHALL provide the ability to specify the maximum amount of device memory used to store Delta Records.	1.0
DeltaRec-HLF-11	A Delta Record MO Enabler SHALL purge Delta Records once the maximum number of Delta Records stored by the device is exceeded.	1.0
DeltaRec-HLF-12	A Delta Record MO Enabler SHALL purge Delta Records once the maximum amount of device memory used to store Delta Records is exceeded.	1.0
DeltaRec-HLF-13	A Delta Record MO Enabler SHALL only send a Delta Record to a DM Server(s) that has ACL permission to read the modified DM Tree element.	1.0
DeltaRec-HLF-14	A Delta Record MO Enabler SHALL report a change to the Delta Records recording status to all DM Servers managing the device via a Generic Alert.	1.0

6. Architectural Model

6.1 Dependencies

The DeltaRec architecture is compliant with the OMA DM architecture [DMARCH].

6.1.1 OMA Device Management [OMA-DM]

The described architecture is compliant with the OMA Device Management enabler [OMA-DM].

6.2 Architectural Diagram

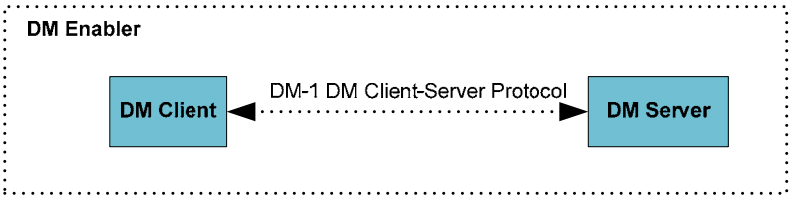


Figure 2: DeltaRec Architecture

6.3 Functional Components and Interfaces/Reference Points Definition

6.3.1 Management Object

6.3.1.1 DeltaRec Management Object

The Delta Record Management Object uses a standardized interface to provide management control of Delta Records on a device. Exposed through the {DM Client} for authorized access by {Management Authorities} utilizing {DM Servers} communicating over {DM-1} using the OMA Device Management Protocol [OMA-DM].

6.3.2 Protocol Endpoints

6.3.2.1 DM Client

The DM Client is the abstract software component that conforms to the requirements for DM Clients specified in the OMA Device Management Enabler.

6.3.2.2 DM Server

The DM Server is the abstract software component that conforms to the requirements for DM Servers specified in the OMA Device Management Enabler.

6.3.3 Interfaces

6.3.3.1 DM-1 Device Management Client-Server Protocol

This provides an interface over which DM Servers may send device management commands to DM Clients and DM Clients may return status and alerts to DM Servers. This is an interface that is bearer neutral and offers many standardized bindings, including HTTP and HTTPS.

6.4 Flows

The Delta Record Management Object described in this Enabler Release is schematic in nature and does not imply any particular protocol or data flow. The OMA Device Management Enabler [OMADM] specifies the protocol and data flows that are expected for client-server communication. These object schemas are utilized by and incorporated into the normal flow descriptions in that enabler; readers are referred to the OMA Device Management Enabler for further understanding.

7. Delta Records Structure

Delta Records are stored on the mobile device in a location that is separate from the DM Tree and implementation-specific. The information in a Delta Record is transmitted to a DM Server as an XML document adhering to the structure described in this section and MUST conform to the XML Schema provided in [DMXSDDeltaRec].

There are three Delta Record types defined showing adds, deletes, and changes to the Client DM Tree. Each record contains a timestamp indicating the time and date the event occurred; the DM Server identifier associated with the sequence number; a sequence number that is the number in the ongoing list of Delta Records accumulating on the device for a specific DM Server; and an authority identifier, which is the identifier of the entity that made the change; the change type; the node or leaf in the DM Tree that was modified; and if applicable, the modified data. These are defined as:

Timestamp: as per ISO 8601, format defined in TND section 7.7.6 TStamp.

DM Server: The DM Server identifier (name) associated with the Sequence Number. This is the DM Server that is to receive the Delta Record.

Sequence Number: 0 .. 65,535 (unsigned integer). The first Delta Record generated begins with sequence number zero.

Authority ID: Normally the DM Server identifier (name) as defined in [DMSTDOBJ] ./DMAcc/ServerID. A change made by the End User may be shown as the device ID from the DevInfo MO (./DevInfo/DevID node) defined in [DMSTDOBJ].

Change Type: Contains one of the strings “Change”, “Delete”, or “Add”.

Node URI – the URI of the node or leaf in the DM Tree that was modified.

Value – this is the current value of the leaf node if it was changed or added to the DM Tree. A deleted node does not have a Value element in the Delta Record.

The modified DM Tree node being referenced by the Delta Record will be identified using its unique URI as defined in the DM Protocol Specification section 5 Node Addressing. The referenced URI will be located in the NodeURI element of the Delta Record. Appendix E provides further information on the relationship between Delta Record data elements.

7.1 Transmitted Delta Records

Delta Records sent to a DM Server are only those the DM Server is authorized to receive according to the ACL for the changed DM Tree element. Other information such as other DM Servers allowed to receive the Delta Record MUST not be transmitted with the Delta Records for a specific DM Server.

7.1.1 Delta Change Record

When a node or leaf value in a DM Tree MO is changed this record will show the new value.

```
<dr:deltarec xmlns:dr=”urn:oma:xml:dm-deltarec:deltarecord:1.0”>
  <dr:timestamp>2011-06-28T10:48:00Z</dr:timestamp>
  <dr:serverid>DMS1958</dr:serverid>
  <dr:seqnumber>23499</dr:seqnumber>
  <dr:authority>DMS1958</dr:authority>
  <dr:changetype>change</dr:changetype>
  <dr:nodeuri>./DMTreeRoot/Node1MO/size</dr:nodeuri>
  <dr:content>
    <dr:contentint>19</dr:contentint>
  </dr:content>
</dr:deltarec>
```

Figure 3: Example of a Change Delta Record in XML form

7.1.2 Delta Delete Record

When a node or leaf in a DM Tree MO is deleted, this record will show the node or leaf that has been deleted. The value element is null in a Delete Delta Record.

7.1.3 Delta Add Record

When a node or leaf in a DM Tree MO is added, this record will show the new node or leaf and value. The DM Tree node URL in the record includes the new node or leaf.

8. Delta Record Management Object

The MO associated with Delta Record management under an internal node *x* (dynamically or statically created).

Management Object identifier: urn:oma:mo:oma-deltarec:1.0. This identifier SHALL be registered with OMNA (Open Mobile Naming Authority).

Protocol Compatibility: This object is compatible with OMA Device Management protocol specifications, version 1.3[DMPRO].

8.1 Figure of the Management Object (Informative)

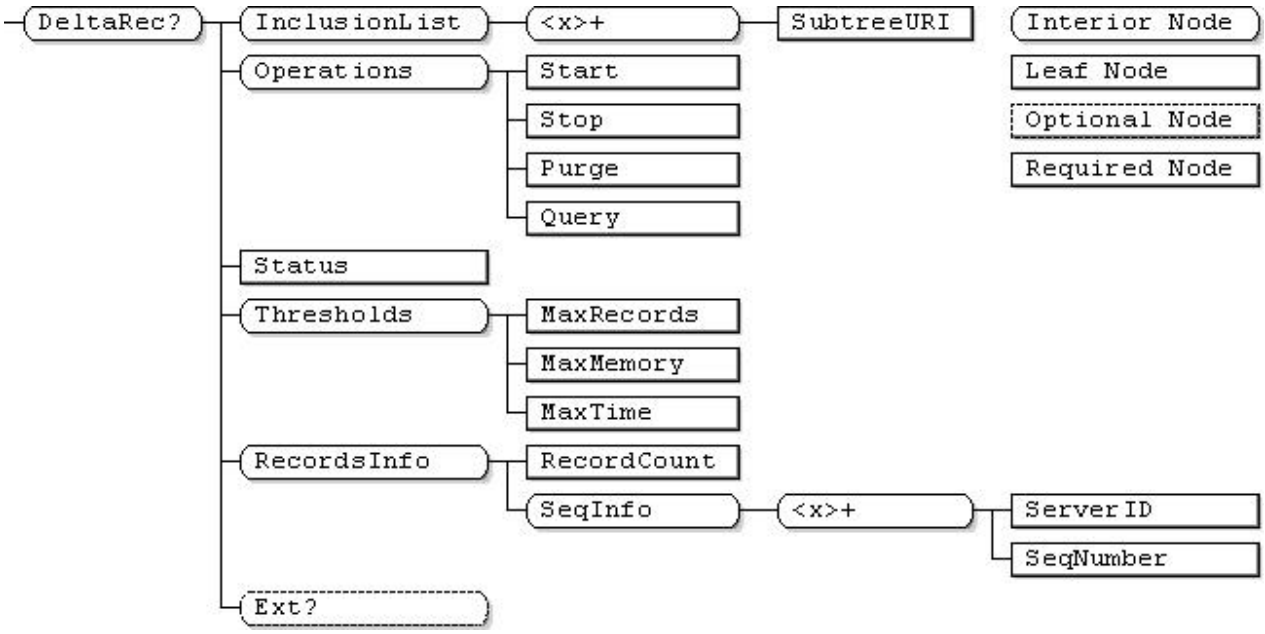


Figure 4: Delta Record Management Object

8.2 Delta Record Management Object Parameters

This section describes the properties for DeltaRecMO.

<x>/DeltaRec

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This interior node groups together the parameters and Delta Records defined as a Delta Record Management Object. The ancestor elements of this node define the position in the Management Tree of this Management Object. The location of a Management Object within the DM Tree is out of scope for this specification.

The type of this node MUST be the Delta Record Management Object ID “urn:oma:mo:oma-deltarec:1.0”.

InclusionList

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node groups the list of nodes or leafs in the DM Tree that will have Delta Records generated if a change is made to elements (nodes or leafs) at and under the URI listed in the <x>/DeltaRec/InclusionList/<x>/SubtreeURI node.

InclusionList/<x>

Status	Tree Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This placeholder node allows multiple parts of the DM Tree to be named for a generation of Delta Record records.

InclusionList/<x>/SubtreeURI

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get, Add, Delete, Replace

This leaf node contains the URI of a node or leaf in the DM Tree indicating all changes at and under this URI will be recorded as Delta Records. A change to the named node - specifically a delete from the DM Tree will generate a Delete Delta Record. Any Ext node occurring under the named URI will be excluded from Delta Record generation. To include a specific Ext node (and subtrees under it) for Delta Record generation, the URI to this Ext node must be explicitly named in a SubtreeURI leaf node.

Values
Leaf values are in URI format [RFC3986]

Operations

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node groups Delta Record operations.

Operations/Start

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Exec

This leaf node is the target of an `Exec` command to start the generation of Delta Records on the device.

If the current status of the <x>/DeltaRec/Status node is “Disabled”, a request to “Start” Delta Records operation will fail. In this case, the DM Client MUST report failure using Status Code 1400 given in Section 9.9 Results Reporting.

Operations/Stop

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Exec

This leaf node is the target of an Exec command to stop the generation of Delta Records on the device.

If the current status of the <x>/DeltaRec/Status node is "Disabled", a request to "Stop" Delta Records operation will fail. In this case, the DM Client MUST report failure using Status Code 1400 given in Section 9.9 Results Reporting.

Operations/Purge

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Exec

This leaf node is the target of an Exec command to purge Delta Records from a DM Client.

Operations/Query

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Exec

This leaf node is the target of an Exec command allowing a DM Server to retrieve all Delta Records for which it has read access. Access to a Delta Record is determined by the ACL of the modified DM Tree element shown in the Delta Record. The Delta Record contains the ACL of the node referenced by the NodeURI when the Delta Records was created. The ACL will be checked to ensure the requesting DM Server has access to the Delta Record.

Status

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get, No Replace

This leaf node specifies the operational state of the Delta Record function. The value of this node MUST be one of the following :

State	Meaning
Stopped	The Delta Record function is stopped.
Running	The Delta Record function is running.
Disabled	The Delta Record function has been disabled due to no resources available on the device for Delta Record generation and management. This state MUST only be set and reset by the Client and overrides the Start and Stop Exec requests.

Thresholds

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node groups the threshold parameters used to determine when to Purge Delta Records from the DM Client in accordance with section 9.6 Purging Delta Records.

Thresholds/MaxRecords

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get, Replace

This leaf node contains the maximum number of Delta Records that may be stored on a DM Client before a Purge is required. When the number of Delta Records exceeds the number specified in this leaf node, the DM Client MUST execute a Purge of all Delta Records in accordance with section 9.6 Purging Delta Records. See section 9.8 Threshold Value Settings for guidance on appropriate values. The maximum value for this parameter SHALL not exceed 65,535.

Thresholds/MaxMemory

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get, Replace

This leaf node indicates the maximum Client memory Delta Records may consume before a Purge is required. The value is expressed in kilobytes. When the amount of memory used to store Delta Records exceeds the specified number in this leaf node, the DM Client MUST execute a Purge of all Delta Records in accordance with section 9.6 Purging Delta Records. The amount of memory used by Delta Records is measured by the number of bytes consumed by the Delta Record described in section 5 Delta Record Structure. See section 9.8 Threshold Value Settings for guidance on appropriate values.

Thresholds/MaxTime

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This leaf node specifies the maximum amount of time, in minutes, the DM Client may wait before it deletes all untransmitted Delta Records. By default, the value SHALL be -1 indicating the DM Client will hold untransmitted Delta Record for a time determined by the Client. See section 9.6 Purging Delta Records for further details.

RecordsInfo

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node groups statistics and control parameters relating to Delta Records.

RecordsInfo/RecordsCount

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This leaf node contains the current number of Delta Records on the Client. The value must be greater than or equal to zero.

RecordsInfo/SeqInfo

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node groups information related to the Delta Record sequence numbering for all DM Servers.

RecordsInfo/SeqInfo/<x>

Status	Tree Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This placeholder groups a DM Server with its associated Delta Record Sequence Number count.

RecordsInfo/SeqInfo/<x>/ServerID

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies a server identifier associated with the <x>/DeltaRec/RecordsInfo/SeqInfo/<x>/SeqNumber node.

RecordsInfo/SeqInfo/<x>/SeqNumber

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This leaf node is an unsigned integer in the range of 0 to 65535. For each DM Server (<x>/DeltaRec/RecordsInfo/SeqInfo/<x>/ServerID), a sequential count of Delta Records associated with the DM Server will be maintained. The value of this leaf node is the sequence number of the last Delta Record generated for the DM Server in the <x>/DeltaRec/RecordsInfo/SeqInfo/<x>/ServerID node. See section 9.7 Sequence Number Management for further details. The first Delta Record generated begins with sequence number zero.

Ext

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This optional interior node designates a branch of the DeltaRec root where platform or vendor extensions MAY be added, permanently or dynamically. Ext sub trees, such as this one, are included at various places in the Delta Record Management Object to provide flexible points of extension for platform or implementation-specific parameters. However, vendor extensions MUST NOT be defined outside of one of these Ext sub-trees.

9. Delta Record Management

The DeltaRec MO allows a DM Server with sufficient authorization to start or stop the generation of Delta Records on a DM Client. The DeltaRec MO also allows a DM Server to retrieve all or some Delta Records and purge all or some Delta Records on the Client. The DM Client may send asynchronously all Delta Records to a DM Server and may purge all accumulated Delta Records.

9.1 Delta Record Data Confidentiality

The DM Tree may be shared among several DM Servers and there may be restrictions on who can view or change parts of the DM Tree, therefore a mechanism to ensure data confidentiality is required. To ensure data confidentiality among multiple entities accessing the DM Tree, Delta Records are stored separately from the DM Tree in a location that is implementation dependent. Each Delta Record will contain a copy of the ACL for the affected node referenced by the Delta Record NodeURI. This ACL will be the ACL at the time the Delta Record is created. The Client will ensure a DM Server that has read access to the node referenced in the NodeURI will be sent the Delta Record associated with that node. A DM Server that does not have read access will not be sent the Delta Record.

As an example, if in a DM Tree there is defined <x>/NodeOne and only DM Server A1 can read, add, change, or delete NodeOne. When the value of NodeOne is changed, the Delta Record will contain an entry in the ACL element that would show, Add=A1&Replace=A1&Delete=A1&Get=A1.

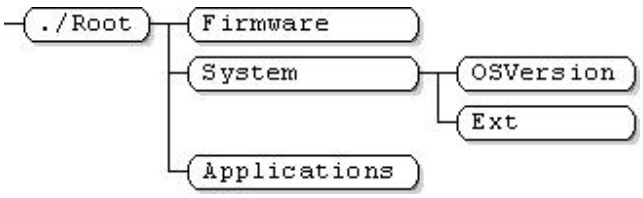
9.2 Delta Record Data Capture

The Delta Record function allows specified sections of the DM Tree to be monitored for changes. These sections are listed in the <x>/DeltaRec/InclusionList parameter of the DeltaRec MO. The Delta Record function requires at least one entry in the InclusionList to be specified.

When the Delta Record function is enabled the <x>/DeltaRec/InclusionList/<x>/ SubtreeURI parameter(s) indicates the part(s) of the DM Tree that -- when modified by a DM Server using the DM Protocol or through the DM Client API -- MUST be recorded in a Delta Record. In addition, any changes to the specified section of the DM Tree performed by the End User, regardless of by what means, MUST also be recorded as a Delta Record. Changes under ./Ext nodes throughout the DM Tree are excluded from the Delta Record function unless specifically listed in the <x>/DeltaRec/InclusionList.

The <x>/DeltaRec/InclusionList allows a DM Tree node or leaf URI to be identified for inclusion in the Delta Record function. The entry in the <x>/DeltaRec/InclusionList/<x>/SubtreeURI and all nodes and leafs at and under this entry in the DM Tree will have Delta Records generated if a change occurs, with the exception of the ./Ext nodes. A deletion of a node in the DM Tree that is named in SubtreeURI will generate a Delete Delta Record. The Delta Record function requires at least one entry in the <x>/DeltaRec/InclusionList to be specified; this may be the root node of the DM Tree.

As an example, a DM Tree with three nodes from the root node structured as:



Having entries in the SubtreeURI leaves as follows;

<x>/DeltaRec/InclusionList/1/SubtreeURI = ./Root/Applications

<x>/DeltaRec/InclusionList/2/SubtreeURI = ./Root/System

would not record changes to the ./Root/Firmware or ./Root/System/Ext subtrees of this example DM Tree.

Recording changes to the Ext (extension) part of the DM Tree is not required, although a manufacturer may agree to generate Delta Records for changes to this part of the DM Tree.

9.3 Starting and Stopping Delta Record Recording

The starting and stopping of Delta Record generation on a DM Client is controlled by the DM Client or DM Server through the DeltaRec MO node <x>/DeltaRec/Operations.

The DM Server or DM Client may perform an `Exec` command on the <x>/DeltaRec/Operations/Start node to turn on the generation of Delta Records, and similarly on the <x>/DeltaRec/Operations/Stop node to turn off Delta Record generation. Successful execution of the `Exec` command will set the value of the <x>/DeltaRec/Status node to either `Running` or `Stopped`. The act of changing the <x>/DeltaRec/Status value will itself generate a Delta Record, including the case where the Delta Record function is stopped. Thus, the first and last Delta Records in a set of Delta Records collected over a period of time between the `Exec` command to start recording and the `Exec` command to stop recoding will be the Delta Records indicating the changes to the <x>/DeltaRec/Status.

The DM Client may at anytime set the <x>/DeltaRec/Status value to `Disabled`. This would normally be used if the DM Client became low on resources or had reached the maximum number of Delta Records it could store based on the values for the <x>/DeltaRec/Thresholds/MaxMemory node or the <x>/DeltaRec/Thresholds/MaxRecords node and is unable to free storage space using the Purge function. The DM Client may resume the operation of the Delta Record function by first setting the Status value to `Stopped`, and then a DM Server may start the generation of Delta Records using the `Exec` command as described above.

Any change to the <x>/DeltaRec/Status node value will be reported to all DM Servers managing the device via a `Generic Alert`. The DM Client will use the `Generic Alert` mechanism defined in section 8.7 of [DMPRO] to asynchronously send to a DM Server the new value of the <x>/DeltaRec/Status. The `Generic Alert` message includes the following data:

- An alert type “urn:oma:at:oma-deltarec:1.0:Status”
- The URI <x>/DeltaRec/Status

The reporting of a change in the <x>/DeltaRec/Status value to all DM Servers managing the device is needed to ensure there is no conflict between DM Servers over the use of the Delta Record function. If a conflict arises, it will be quickly identified using this reporting mechanism. Resolution of the conflict is out of the scope of this specification and is left to the appropriate DM Server administrators to resolve.

Following is an example message:

```
<Alert>
   <CmdID>2</CmdID>
   <Data>1226</Data>    <!-- Alert Code for Generic Alert -->
   <Item>
    <Source>
      <LocURI>./DeltaRec/Status</LocURI>
    </Source>
   <Meta>
     <Type xmlns='syncml:metinf'> urn:oma:at:oma-deltarec:1.0:Status</Type>
     <Format xmlns='syncml:metinf'>text/plain</Format>
    </Meta>
    <Data>
      <!-- content of ./DeltaRec/Status one of “Stopped”, “Running” or
“Disabled”-->
    </Data>
   </Item>
</Alert>
```

9.4 DM Server Retrieving Delta Records

The DM Server with appropriate authority may retrieve all Delta Records from a client at any time using the Exec command applied to the <x>/DeltaRec/Operations/Query node.

An `Exec` command on the Query node will trigger the DM Client to retrieve all Delta Records for the DM Server invoking the `Exec,` and use the `Generic Alert` mechanism defined in [DMPRO] section 8.7 to asynchronously send Delta Records to the DM Server. The `Generic Alert` message includes the following data:

- An alert type “urn:oma:at:oma-deltarec:1.0:Records”,
- `Exec` command operation node URI, <x>/DeltaRec/Operations/Query
- All Delta Records on the device in the XML format as described in section 7 Delta Records Structure,
- An integer status code as defined in section 9.9 Results Reporting.

The following is an example message:

```
<Alert>
   <CmdID>2</CmdID>
   <Data>1226</Data>    <!-- Alert Code for Generic Alert -->
   <Item>
     <Source>
       <LocURI>./DeltaRec/Operations/Query</LocURI>
     </Source>
     <Meta>
       <Type xmlns='syncml:metinf'> urn:oma:at:oma-deltarec:1.0:Records</Type>
          <Format xmlns=' syncml:metinf'>xml</Format>
      </Meta>
      <Data>
         <!-- Delta Records in XML according to Section 7 -->
      </Data>
    <Data>1200</Data>     <!-- Result Code -->
   </Item>
</Alert>
```

Since multiple DM Servers may manage a single DM Client a DM Server will only be able to retrieve Delta Records where it is shown as having read access in the record ACL.

9.5 DM Client Sending Delta Records

To prevent Delta Records from accumulating on a Client indefinitely, the Client may asynchronously send all Delta Records to a DM Server(s). A set of parameters is defined under the <x>/DeltaRec/Thresholds node allowing a DM Server or DM Client to set the maximum number of records that may be stored on a DM Client, and/or the maximum amount of Client memory consumed by Delta Records. If these values are exceeded, the DM Client MUST Purge the Delta Records in accordance with section 9.6 Purging Delta Records. . The Client may also use other triggering events to send Delta Records, such as prior to or after a software download and installation, or after a reboot of the device.

Upon receiving the Delta Records the DM Server SHOULD apply the changes indicated in the Delta Records to its copy of the DM Tree associated with that Client, resulting in the synchronization of the Client and Server DM Trees. Once the DM Client has received a response from the DM Server indicating successful receipt of the Delta Records the DM Client MAY delete all the Delta Records that have been sent.

The DM Client will use the `Generic Alert` mechanism defined in section 8.7 of [DMPRO] to asynchronously send Delta Records to a DM Server. The `Generic Alert` message includes the following data:

- An alert type "urn:oma:at:deltarec:1.0:Records"
- All Delta Records on the device in the XML format as described in section 7 Delta Records Structure.

The Delta Records sent to a DM Server are only those the Server has authority to receive as indicated in the ACL of the node referenced by NodeURI. The DM Client MUST not send a `Generic Alert` to a DM Server if there are no Delta Records present applicable to that DM Server (for example, a null `Generic Alert` is not to be used to report that no Delta Records are stored for the DM Server). The DM Server may find this information by performing a `Get` on the <x>/DeltaRec/Records/AccumulatedRecords node.

Following is an example message:

```
<Alert>
   <CmdID>2</CmdID>
   <Data>1226</Data>    <!-- Alert Code for Generic Alert -->
   <Item>
     <Meta>
       <Type xmlns='syncml:metinf'> urn:oma:at:oma-deltarec:1.0:Records</Type>
          <Format xmlns=' syncml:metinf'>xml</Format>
      </Meta>
      <Data>
         <!-- Delta Records in XML according to Section 7 -->
      </Data>
   </Item>
</Alert>
```

9.6 Purging Delta Records

The DM Client SHOULD Purge Delta Records from time to time to ensure there is not a long-running accumulation of records. The Threshold values set in the <x>/DeltaRec/Thresholds node will ensure Delta Records do not accumulate to such an extent that they may consume resources needed by the Client. Threshold values SHOULD be set so that at maximum value, the resources on the device allow for successful transmission and Purging. In addition, the Client or Server may Purge records at anytime. The DM Server may choose to store Delta Records indefinitely and for purposes other than DM Tree synchronization, but this is out of the scope of this specification.

To Purge records from a DM Client the DM Server issues an `Exec` command on the DM Client's <x>/DeltaRec/Operations/Purge node. This will invoke a function to remove all Delta Records stored on the DM Client at the time the Exec command is received. The DM Client may start a Purge using internal means.

When a Purge is invoked, regardless of source, the DM Client MUST first send all Delta Records to their respective DM Servers in accordance with section 9.5 DM Client Sending Delta Records. When the DM Client receives an acknowledgement from the DM Server indicating the server has received the Delta Records and once all DM Servers have similarly acknowledged receipt of their Delta Records, the DM Client MUST delete the successfully transmitted Delta Records, and update the <x>/DeltaRec/RecordsInfo/RecordsCount node with the current number of Delta Records on the Client.

If a DM Server is unable to receive the transmitted Delta Records, those unsuccessfully transmitted records SHALL remain on the Client and the Client SHALL retransmit the Delta Records when it is next connected to that DM Server. All records that have been successfully transmitted to their appropriate DM Severs SHALL be deleted. A timer value in the <x>/DeltaRec/Thresholds/MaxTime node MAY be defined to set a limit on how long Delta Records are held on the Client. Once this time has expired, the DM Client SHOULD delete all Delta Records created prior to the time the Purge was started.

If during a Purge a Delta Record is found that has an invalid DM Server entry in the ACL (for example, the DM Server is no longer actively managing the DM Client and has had its entry in the DM Account Managed Object deleted), the Delta

Records associated with the inactive DM Server MUST be deleted. A Delta Record with a valid DM Server and an invalid DM Server in the record ACL will have the record forwarded to the valid DM Server before deletion.

9.7 Sequence Number Management

Each Delta Record may be applicable to one or more DM Servers based on the ACL of the changed DM Tree element tracked by the Delta Record. Multiple DM Servers may be associated with a single Delta Record, but each Delta Record may have a different set of DM Servers associated with it, since some Delta Records will apply to some DM Servers and not to others. Therefore, the DM Client SHALL maintain separate sequence numbers for each DM Server so that the sequence number in the Delta Record sent to the DM Server is continuously incremented. The sequence number will be maintained in the DM Tree structure element <x>/DeltaRec/RecordsInfo/SeqInfo/<x>/SeqNumber. The reliability of transmission shall be guaranteed by DM protocol.

When a sequence number reaches its maximum value of 65,535 a Purge of Delta Records MUST take place. After the Purge, the sequence number that was at its maximum value will roll over to zero when the next Delta Record is generated.

Upon receiving a Delta Record, the DM Server shall check the sequence number of the record and apply the changes to its DM Tree respectively only if the sequence number of the received Delta Record is sequential. A Delta Record received by a DM Server with a duplicate sequence number SHALL be discarded. Delta Records received out of sequence by the DM Server may be held pending arrival of the missing Delta Record(s).

The <x>/DeltaRec/RecordsInfo/SeqInfo/<x>/SeqNumber leaf node may be read by the DM Server to determine the sequence number of the last Delta Record available or reported to the DM Server. For example, if the DM Server has received a Delta Record with sequence number 1436, and it then sends a request to the DM Client to retrieve the value of leaf node SeqNumber, and the value returned is 1440, the DM Server can calculate that four new Delta Records are on the DM Client related to itself and have not yet been delivered. If the value of SeqNumber was returned as 1436, the DM Server would know there are no new Delta Records on the Client related to itself.

If a given DM Server's account information is deleted from the DM Client, then any Delta Records that have not been uploaded to the deleted DM Server will be removed by the DM Client. Entries for the deleted DM Server occurring in <x>/DeltaRec/RecordsInfo/SeqInfo/<x> will be removed.

9.8 Threshold Value Settings

The setting of <x>/DeltaRec/Thresholds/MaxRecords and <x>/DeltaRec/Thresholds/MaxMemory values are specific to the configuration of the Client device hardware. The DM Server normally would have information about the Client device configuration by using the manufacturer name and model type to determine values for these parameters. But due to variations of hardware configurations, a DM Client may have the best information regarding available resources on a device and may also set these parameters.

The purpose of setting these threshold values is to limit the amount of resources (primarily RAM), consumed by the presence of Delta Records. The Delta Record function is intended to run completely transparent to the user and have no perceivable impact on the users' operation of the device. With this in mind the threshold values should be set to allow for the accumulation of Delta Records to an amount that does not slow down the device operation.

The setting of threshold values that are too low will have the effect of sending Delta Records to DM Servers too frequently generating unnecessary over the air transmissions. Setting the values to zero will cause a Delta Record to be sent each time it is created, leaving no records stored on the device. These value settings should be avoided.

The threshold parameters MaxRecords and MaxMemory can work together and also independently. The size of a Delta Record is normally under 100 bytes of data, although some exceptions may exist. Therefore setting MaxMemory to be the product of 100 × MaxRecords would have both parameters reserving approximately the same amount of memory. To set only one threshold parameter and not use the other parameter the unused parameter may be set to -1. For example MaxMemory is set to 100 kilobytes (approximately 1000 Delta Records) and the MaxRecords parameter is set to -1; when 100 kilobytes is exceeded the DM Client will Purge Delta Records from the device.

It is also possible to leave the setting of these thresholds to the management of the DM Client. By placing a -1 in the threshold value for both MaxMemory and MaxRecords the DM Client will police the resources used by Delta Records. In this case the amount of resources consumed by Delta Records is implementation dependent.

The following table summarizes the above discussion:

Threshold Value	MaxMemory	MaxRecords	Notes
-1	Do not use this parameter in resource calculations	Do not use this parameter in resource calculations	When both parameters are set to -1 the DM Client determines resource usage levels
0	Delta Record sent to DM Server upon creation	Delta Record sent to DM Server upon creation	Not recommended due to high bandwidth usage
N > 0	Purge Delta Records when amount of memory consumed exceeds N kilobytes	Purge Delta Records when number of records exceeds N	N should be a number allowing accumulation of Delta Records on the Client

9.9 Results Reporting

An `Exec` command may be invoked by a DM Server on a DM Client supporting the Delta Record Enabler to execute one of the operations; Start, Stop, Purge and Query. The status of the `Exec` command invocation is reported back to the DM Server as a Response Status Code in a `Status` message associated with the `Exec` command and MUST be sent to the invoking DM Server.

The Delta Record Enabler uses the `Generic Alert` mechanism defined in [DMPRO] section 8.7 Generic Alert to asynchronously send Delta Records to the DM Server. Each `Generic Alert` message MUST contain a Response Status Code sent as an integer value in the Item/Data element of the `Generic Alert` message.

The Response Status Codes for the Delta Record Enabler MUST be one of the values defined below sent as an integer value.

Status Code	Meaning	Usage
1200	Successful	Successful - The Request has Succeeded
1201-1299	Success – Reserved for Delta Record	Reserved for reporting Success of a requested Delta Record operation.
1400	Operation Failed	The request to execute a Delta Record operation failed.
1401	Not authorized	The server is not authorized to conduct this operation.
1402	Not Implemented	The device does not support the requested operation.
1403-1499	Failure - Reserved for Delta Record	Reserved for reporting Failure of requested Delta Record operation.

Table 1 – Delta Record Response Status Codes

9.10 Alert Types

The following alert types are defined for use by the Delta Record Enabler:

- urn:oma:at:oma-deltarec:1.0:Records. This alert type is used to report Delta Records to a DM Server in a `Generic Alert`;

- urn:oma:at:oma-deltarec:1.0:Status. This alert type is used in a `Generic Alert` to report the operational status of the Delta Record Enabler as it is indicated in the <x>/DeltaRec/Status leaf node, and;
- urn:oma:at:oma-deltarec:1.0:OperationStatus. This alert type is used in a `Generic Alert` to report the status of an `Exec` command invoked by a DM Server on a DM Client.

10.Release Information

10.1 Supporting File Document Listing

Doc Ref	Permanent Document Reference	Description
Supporting Files		
[DMDRXSL]	OMA-SUP-XSD_DeltaRec_DeltaRecord-V1_0-20110927-C	XML Schema describing the Delta Record XML structure
[DMDRMODDF]	OMA-SUP-MO_DeltaRec-V1_0-20110927-C	Delta Record Device Description File. Working file in DM_MO directory: http://www.openmobilealliance.org/tech/omna/dm_mo/DeltaRecMO-V1_0.ddf

Table 2: Listing of Supporting Documents in DeltaRecMO Release

10.2 OMNA Considerations

The OMNA registry maintains the following in the MO registry:

MO Identifier	Description	Owner	Version	MO DDF	MO Spec
urn:oma:mo:oma-deltarec:1.0	Delta Record MO	DM WG	1.0	DeltaRec-V1_0.ddf	OMA-ER-DeltaRec-V1_0-20110927-C

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-ER-DeltaRecMO-V1_0-20130521-A	21 May 2013	Status changed to Approved by TP TP Ref # OMA-TP-2013-0137-INP_DeltaRecMO_V1_0_ERP_for_Final_Approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [IOPPROC].

B.1 SCR for DeltaRec Tree Structure

Item	Function	Reference	Requirement
DeltaRecMO-T-001-M	Use of appropriate management object identifier for the DeltaRec root node	Section 8	
DeltaRecMO-T-002-M	Support for Required nodes under root node	Section 8.2	
DeltaRecMO-T-003-O	Support for Optional nodes	Section 8.2	

B.2 SCR for DeltaRec Client

Item	Function	Reference	Requirement
DeltaRecMO-C-001-M	The ability to start and stop the recording of Delta Records on the device.	Section 9.3	
DeltaRecMO-C-002-M	The ability to purge Delta Records on the device.	Section 9.6	
DeltaRecMO-C-003-M	The ability to send Delta Records asynchronously to the DM Server.	Section 9.5	
DeltaRecMO-C-004-M	A Delta Record SHALL include the time the change was made to the Client DM Tree.	Section 7	
DeltaRecMO-C-005-M	A Delta Record SHALL include information identifying the authority that enacted the change on the Client DM Tree.	Section 7	
DeltaRecMO-C-006-M	A Delta Record SHALL include a sequence number from a sequence number series associated with each DM Server managing the device.	Section 7	
DeltaRecMO-C-007-M	The ability to specify the maximum number of Delta Records stored by the device.	Sections 8.2, 9.8	
DeltaRecMO-C-008-M	The ability to specify the maximum amount of device memory used to store Delta Records.	Sections 8.2, 9.8	

DeltaRecMO-C-009-M	Delta Records SHALL be purged once the maximum number of allowed Delta Records (as specified in <x>/DeltaRec/Thresholds/MaxRecords) stored by the device is exceeded.	Section 9.6	
DeltaRecMO-C-010-M	Delta Records SHALL be purged once the maximum amount of device memory used to store Delta Records (as specified in <x>/DeltaRec/Thresholds/MaxMemory) is exceeded.	Section 9.6	
DeltaRecMO-C-011-M	Each Delta Record is only sent to those DM Server(s) that have ACL permission to read the modified target DM Tree element.	Section 9.1	
DeltaRecMO-C-012-M	Report a change to the Delta Records recording status (as indicated in <x>/DeltaRec/Recording/Status) to all DM Servers using a Generic Alert.	Section 9.3	
DeltaRecMO-C-011-M	The ability to retransmit purged Delta Records to a DM Server if an initial transmission is unsuccessful.	Section 9.6	
DeltaRecMO-C-012-M	The ability to retransmit purged Delta Records to a DM Server if an initial transmission is unsuccessful until the expiry of timer <x>/DeltaRec/Thresholds/MaxTime.	Section 9.6	
DeltaRecMO-C-13-M	The ability to generate Delta Records only for the parts of the DM Tree identified in the <x>/DeltaRec/InclusionList/<x>/ SubtreeURI entries.	Section 9.2	

B.3 SCR for DeltaRec Server

Item	Function	Reference	Requirement
DeltaRecMO-S-001-O	Start and stop the recording of Delta Records on the device.	Section 9.3	
DeltaRecMO-S-002-O	The ability to purge Delta Records on the device.	Section 9.6	
DeltaRecMO-S-003-O	Ability to retrieve Delta Records from the device.	Section 9.4	
DeltaRecMO-S-004-O	Use sequence numbers contained in the DM Client <x>/DeltaRec/Records/DMSRecordsSequence/SeqNumber, to determine whether all Delta Records have been received.	Section 9.7	

B.4 SCR for DeltaRec System

Item	Function	Reference	Requirement
DeltaRecMO-SYS-001-M	Record changes applied to a Client DM Tree.	Sections 4, 9.2	

Appendix C. Best Practices for Delta Record Functions (Informative)

To prevent Delta Records from accumulating on a mobile device indefinitely, it is recommended that DM Servers periodically retrieve Delta Records from a DM Client and use those records to update (synchronize) their DM Tree. Once the update is successful the DM Server should send a command to the DM Client to purge all records.

The DM Client should send all Delta Records to their appropriate DM Servers before purging Delta Records.

The purpose of the InclusionList is to allow for nodes and leafs that are of particular interest to the DM Server to be identified. The InclusionList can also be used to exclude parts of the DM Tree that may have frequent changes (such as those used to collect statistics in the Diagnostic and Monitoring MO), since it is not the purpose of the Delta Record MO to generate Delta Records for parts of the DM Tree that change frequently, such as statistics counters.

Appendix D. Delta Record Sequence Numbers (Informative)

Sequence Numbers are associated with a specific DM Server and will increment continuously from 0 to 65,535 then reset to 0 after 65,535. Sequence Numbers for different DM Servers will be maintained; Sequence Numbers for individual DM Servers may not be the same, since some Delta Records will apply to some DM Servers and not to others due to different ACL configurations. When the Sequence Number for a specific DM Server reaches its maximum value of 65,535, the DM Client must perform a Purge of all Delta Records. The Sequence Number for DM Servers that have not reached the maximum value will not be set to zero until they reach the maximum value. The following table illustrates the above:

Time	Sequence Numbers			Number of Delta Records Stored on the Client
	DM Server A	DM Server B	DM Server C	
T0	65532	20000	65533	0 (a Purge had just been done and all Delta Records have been deleted from the Client)
T1	65533	20001	65533	1 (a new Delta Record is added to the Client which applies to DMS A & B)
T2	65534	20002	65533	2 (a new Delta Record is added to the Client for DMS A & B)
T3	65535	20003	65534	3 (a new Delta Record is added to the Client for DMS A, B & C)
T4 *Purge all Delta Records*				0 (All Delta Records are sent to the appropriate DM Servers, three to A & B and one for C)
T5	0	20004	65534	1 (a new Delta Record is added for DMS A & B)
T6	1	20005	65535	2 (a new Delta Record is added for DMS A, B & C)
T7 *Purge all Delta Records*				0 (All Delta Records sent to the appropriate DM Servers, two for A & B and one for C)
T8	2	20006	0	1 (a new Delta Record added for DMS A, B & C)
T9	3	20007	1	2 (a new Delta Record added for DMS, A, B, & C)

Appendix E. Delta Record Data Elements (Informative)

The following figure is an example data structure showing the relationship of Delta Record data elements.

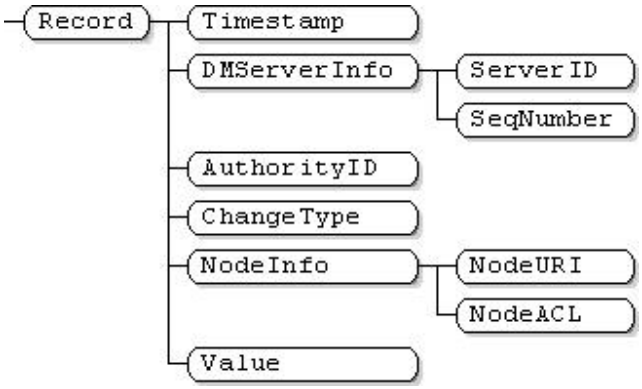


Figure E.1 Example Delta Record Data Structure

The above diagram is a graphical representation of a data structure to store Delta Record data on a Client and is only for illustrative purposes. Delta Records are not part of the DM Tree and are not stored as a managed object. Delta Record data sent to the DM Server will be sent in the Data part of the Generic Alert in CDATA escaped XML format.

Each change made to a Client DM Tree that is to be sent as a Delta Record will need to have information stored on the Client which would include;

- the time of the change,
- the identifiers for DM Server(s) that have authority to see the changed data in the DM Tree indicated by the ACL of the change node or leaf in the DM Tree,
- a Delta Record sequence number associated to each DM Server,
- the identity of the entity that made the change,
- the element in the DM Tree that was changed,
- the access rights to that DM Tree node that was changed. This is used to determine which DM Servers may receive the Delta Record, and
- the new value of the changed element, if applicable.