



Mississippi Department of Mental Health (DMH)

**Technical Assistance Note:
Deleting Root Entries**

Introduction

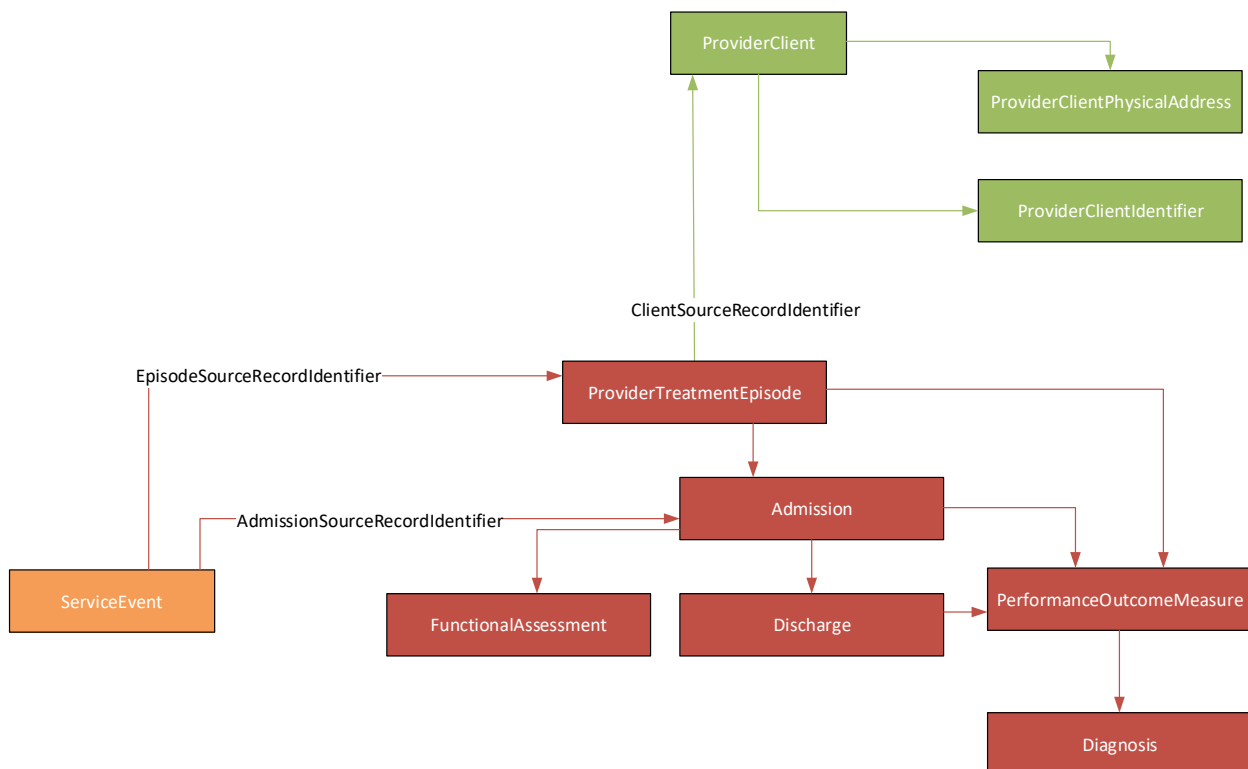
A small change is in development now that affects how “root entries” – Service Events, Treatment Episodes, and Provider Client records – can be deleted by Providers. This document describes the new functionality and how Providers will need to adjust what they send to the Data Warehouse. It also describes how deletions function in the Data Warehouse.

NOTE: XML EXAMPLES IN THIS DOCUMENT MAY BE TRUNCATED FOR ILLUSTRATION PURPOSES AND MAY NOT INCLUDE NECESSARY DATA.

Background

In the XML files that a Provider sends to the Data Warehouse, the Provider can specify an action in an object tag. If no action is specified, the default action is “create/update”. If you need a review on the actions you can supply in an XML file, see the “XML Submission Review” section at the end of this document. This document specifically describes how to use the “delete” action to delete a top-level object. A top-level object is the root of the XML. An easy way to think of it is: If the object is the name of the file, it’s a top-level object. If there’s a file called TreatmentEpisodeDataSet and you’re deleting a Treatment Episode, that’s a top-level object. Deleting sub-entities (like a Performance Outcome Measure or a ProviderClientPhysicalAddress) happens as it always had, by adding `action="delete"` in the entity tag. Remember that deleting any entity will “cascade” the delete to any sub-entities (so deleting a Performance Outcome Measure will delete all Diagnoses under it). Deletes only cascade within a data set; so deleting a Client record will not automatically delete Treatment Episode records related to that client.

Another way to think of this is to review the Overall Submission Data Model:



Each Data Set is a different color – Client in green, Treatment Episode in Red, and Service Event in orange. If a top level record in a data set is deleted – for example, a Provider Treatment Episode is deleted – the delete will cascade down, deleting all of the (red) related objects for that Treatment Episode.

Requirements and Repercussions for deleting top-level objects

1. Top-level object deletions must be sent in a **separate XML file**. This file must contain the Data Set Name and the word “Delete”

Examples:

TreatmentEpisodeDataSet_Delete_040220201423.xml

ClientDataSet_Delete_033120200940.xml

2. If objects are being deleted from multiple data sets, deletion files must be sent in the following order.
 1. ServiceEventDataSet_Delete
 2. TreatmentEpisodeDataSet_Delete
 3. ClientDataSet_Delete

The order of the files is to prevent “orphaning” records. Rules are enforced within a data set, but not across data sets. To prevent a Client record from being deleted where that Client has active Treatment Episode records, the Treatment Episode Delete is sent before the Client is deleted.

3. If a top-level object is deleted, **all objects within that object will be deleted within the data set**. This action cannot be undone. Note that an undelete can be performed on the top-level object, but that will not cascade to all sub-entities (see “Deletes and Undeletes”, below). Cascading deletes for top-level objects will work as follows:
 - If a **Provider Client** record is deleted
 - All associated Provider Client Phone records will be deleted.
 - All associated Provider Client Physical Address records will be deleted.
 - If a **Provider Treatment Episode** record is deleted
 - All associated Admission records will be deleted. Each time an Admission is deleted:
 - All Discharge records for that Admission will be deleted.
 - All Functional Assessment records for that Admission will be deleted.
 - All Performance Outcome Measures for that Treatment Episode will be deleted, including any Performance Outcome Measures that were associated with an Admission or Discharge within the Treatment Episode. Each time a Performance Outcome Measure is deleted:
 - All Diagnosis records for that Performance Outcome Measure will be deleted.
 - All UsedSubstance records for that Performance Outcome Measure will be deleted.

Deletes and Undeletes

Deleting something in the Data Warehouse performs a “soft delete” on it. A “soft delete” means

- The delete flag for that record is set
- That record will not appear in any searches or reports.
- Sending an “undelete” action for that record will clear the delete flag, effectively “restoring” the object.
- Deletes cascade; undeletes do not.
- All rules are checked on both deletes and undeletes. If deleting an entity would cause a rule to be violated (e.g. Every Discharge must have a Performance Outcome Measure; a delete is sent for the Performance Outcome Measure for a Discharge), XML processing will fail with the relevant error message. Similarly, if undeleting an entity will cause a rule to be violated (e.g. Every Discharge must have a Performance Outcome Measure and a Discharge is undeleted but the Performance Outcome Measure is not), the XML processing will fail with the relevant error message.
- Provider Client Example:

In this example we have three clients, listed in the ProviderClient table. The colors (green, orange, yellow) are used to show which of the sub-entities are associated with each client. For example, “Bugs Bunny” has two addresses (NC and VA); “Daffy Duck” has one address (CT). At this point, no records have been deleted and all are visible.

ProviderClient

| Delete Status | First Name | Last Name |
|---------------|------------|-----------|
| 0 | Bugs | Bunny |
| 0 | Daffy | Duck |
| 0 | Elmer | Fudd |

ProviderClientPhysicalAddress

| Delete Status | Type | Address 1 | City | State | Zip | Client |
|---------------|------|-----------------|-----------|-------|-------|--------|
| 0 | Home | 123 Anywhere St | Charlotte | NC | 28262 | Bugs |
| 0 | Work | 222 Main | Vienna | VA | 22222 | Bugs |
| 0 | Home | 123 Broad St | Windsor | CT | 6095 | Daffy |
| 0 | Home | 551 Western Ave | Alamosa | CO | 81101 | Elmer |

ProviderClientIdentifier

| Delete Status | Type | Identifier | Client |
|---------------|------|-------------|--------|
| 0 | SSN | 111-11-1111 | Bugs |
| 0 | SSN | 222-22-2222 | Daffy |
| 0 | SSN | 333-33-3333 | Elmer |

If we delete the “Work” address for Bugs Bunny by changing “Delete Status” to 1, that address disappears from the Data Warehouse. It is still in the database, but hidden from view (note the gray text):

ProviderClient

| Delete Status | First Name | Last Name |
|---------------|------------|-----------|
| 0 | Bugs | Bunny |
| 0 | Daffy | Duck |
| 0 | Elmer | Fudd |

ProviderClientPhysicalAddress

| Delete Status | Type | Address 1 | City | State | Zip | Client |
|---------------|------|-----------------|-----------|-------|-------|--------|
| 0 | Home | 123 Anywhere St | Charlotte | NC | 28262 | Bugs |
| 1 | Work | 222 Main | Vienna | VA | 22222 | Bugs |
| 0 | Home | 123 Broad St | Windsor | CT | 6095 | Daffy |
| 0 | Home | 551 Western Ave | Alamosa | CO | 81101 | Elmer |

ProviderClientIdentifier

| Delete Status | Type | Identifier | Client |
|---------------|------|-------------|--------|
| 0 | SSN | 111-11-1111 | Bugs |
| 0 | SSN | 222-22-2222 | Daffy |
| 0 | SSN | 333-33-3333 | Elmer |

If we then delete the top-level client record for Bugs Bunny, all of the Physical Address and Identifier records for that client are deleted (Note – the Work Address record was already deleted and is unchanged). This is what that looks like:

ProviderClient

| Delete Status | First Name | Last Name |
|---------------|------------|-----------|
| 1 | Bugs | Bunny |
| 0 | Daffy | Duck |
| 0 | Elmer | Fudd |

ProviderClientPhysicalAddress

| Delete Status | Type | Address 1 | City | State | Zip | Client |
|---------------|------|-----------------|-----------|-------|-------|--------|
| 1 | Home | 123 Anywhere St | Charlotte | NC | 28262 | Bugs |
| 1 | Work | 222 Main | Vienna | VA | 22222 | Bugs |
| 0 | Home | 123 Broad St | Windsor | CT | 6095 | Daffy |
| 0 | Home | 551 Western Ave | Alamosa | CO | 81101 | Elmer |

ProviderClientIdentifier

| Delete Status | Type | Identifier | Client |
|---------------|------|-------------|--------|
| 1 | SSN | 111-11-1111 | Bugs |
| 0 | SSN | 222-22-2222 | Daffy |
| 0 | SSN | 333-33-3333 | Elmer |

At this point, the entire Client Record for “Bugs Bunny” has been deleted from the Data Warehouse.

Finally, if we undelete the top-level client record, **only that record is undeleted** (no Physical Address or Identifier records are undeleted):

ProviderClient

| Delete Status | First Name | Last Name |
|---------------|------------|-----------|
| 0 | Bugs | Bunny |
| 0 | Daffy | Duck |
| 0 | Elmer | Fudd |

ProviderClientPhysicalAddress

| Delete Status | Type | Address 1 | City | State | Zip | Client |
|---------------|------|-----------------|-----------|-------|-------|--------|
| 1 | Home | 123 Anywhere St | Charlotte | NC | 28262 | Bugs |
| 1 | Work | 222 Main | Vienna | VA | 22222 | Bugs |
| 0 | Home | 123 Broad St | Windsor | CT | 6095 | Daffy |
| 0 | Home | 551 Western Ave | Alamosa | CO | 81101 | Elmer |

ProviderClientIdentifier

| Delete Status | Type | Identifier | Client |
|---------------|------|-------------|--------|
| 1 | SSN | 111-11-1111 | Bugs |
| 0 | SSN | 222-22-2222 | Daffy |
| 0 | SSN | 333-33-3333 | Elmer |

This action will fail because there is a rule that requires an Address and Identifier for each client. If you are sending an Undelete for the top-level Provider Client record, the validation rules will run for Provider Client. To comply with the Submission Guidelines (see the Provider Client Submission Guide), you can either create new sub-entities or undelete the appropriate sub-entities in the same XML file.

XML Submission Review

As an example, an XML file to create a Client record might look like this:

```
<ClientDataSet>
  <ProviderClients>
    <ProviderClient>
      <SourceRecordIdentifier>9012345</SourceRecordIdentifier>
      <ProviderSourceRecordIdentifier>12345</ProviderSourceRecordIdentifier>
      <BirthDate>2/5/1990</BirthDate>
      <FirstName>Cindy</FirstName>
      <MiddleName>Lou</MiddleName>
      <LastName>Client</LastName>
      <SuffixName></SuffixName>
      <MaidenName>Who</MaidenName>
      <SexCode>8300.2</SexCode>
      <SexualOrientationCode>11300.1</SexualOrientationCode>
      <VeteranStatusCode>1400.2</VeteranStatusCode>
```

```

    <RaceCode>6800.1</RaceCode>
    <EthnicityCode>6900.5</EthnicityCode>
    <ProviderClientPhysicalAddresses>
      <ProviderClientPhysicalAddress>
        <TypeCode>8600.1</TypeCode>
        <FirstStreetAddress>100 Airpark Dr.</FirstStreetAddress>
        <SecondStreetAddress>Suite 401</SecondStreetAddress>
        <CityName>Jackson</CityName>
        <StateCode>1600.MD</StateCode>
        <PostalCode>39056</PostalCode>
        <CountyCode>1700.25</CountyCode>
      </ProviderClientPhysicalAddress>
      <ProviderClientPhysicalAddress>
        <TypeCode>8600.2</TypeCode>
        <FirstStreetAddress>200 Gulf Dr.</FirstStreetAddress>
        <SecondStreetAddress>Apt. 4</SecondStreetAddress>
        <CityName>Jackson</CityName>
        <StateCode>1600.MD</StateCode>
        <PostalCode>39056</PostalCode>
        <CountyCode>1700.25</CountyCode>
      </ProviderClientPhysicalAddress>
    </ProviderClientPhysicalAddresses>
  </ProviderClient>
</ProviderClients>
</ClientDataSet>

```

Note that there is no "Action". Similarly, if the above XML file had already been sent, the following could be sent as an update:

```

<ClientDataSet>
  <ProviderClients>
    <ProviderClient>
      <SourceRecordIdentifier>901234</SourceRecordIdentifier>
      <ProviderSourceRecordIdentifier>1234</ProviderSourceRecordIdentifier>
      <LastName>Smith</LastName>
    </ProviderClient>
  </ProviderClients>
</ClientDataSet>

```

Again, no action is provided. Because the Source Record Identifier and Provider Source Record Identifier match an existing record (green highlighting), the DWH interprets this as an Update. The updated information (yellow highlighting) is applied to the existing record. Cindy Lou Client becomes Cindy Lou Smith.

Actions are sent in the XML tag of an object. For example, if we wanted to delete the Physical Address for this Client record, we could send

```

<ClientDataSet>
  <ProviderClients>
    <ProviderClient>
      <SourceRecordIdentifier>901234</SourceRecordIdentifier>
      <ProviderSourceRecordIdentifier>1234</ProviderSourceRecordIdentifier>
      <ProviderClientPhysicalAddresses>
        <ProviderClientPhysicalAddress action="delete">
          <TypeCode>8600.2</TypeCode>
        </ProviderClientPhysicalAddress>
      </ProviderClientPhysicalAddresses>
    </ProviderClient>
  </ProviderClients>
</ClientDataSet>

```

It's important to note that to delete a sub-entity like this (in this case, a ProviderClientPhysicalAddress), you must identify it. In the above XML, we do that by first identifying the Client (using Source Record Identifier and Provider Source Record Identifier) and then by which Address record to delete. Reference the relevant Submission Guide to see what information needs to be provided to delete a record. In this case, in the Client Submission Guide, under 1.3.1.6 "ProviderClientPhysicalAddress" it reads

"The fields in this entity that will be used to uniquely identify a record, to determine whether to create or update an existing record, and to **delete an existing record** are ... Type Code"

This means that we need to identify the ProviderClient in which we are deleting the Address and the Type Code that we are deleting. We identify the ProviderClient as we did above – by SRI and Provider SRI – and then by providing the Type Code, the Physical Address record with Type Code 8600.2 is deleted.

Finally, we can delete an entire Client record, which would be done in a separate file as described above:

```
<ClientDataSet>
  <ProviderClients>
    <ProviderClient action=delete>
      <SourceRecordIdentifier>9012345</SourceRecordIdentifier>
      <ProviderSourceRecordIdentifier>12346</ProviderSourceRecordIdentifier>
    </ProviderClient>
  </ProviderClients>
</ClientDataSet>
```

Note that the delete action is in the ProviderClient tag and the SRI and Provider SRI are provided. Also note that this would delete any Identifiers or Addresses associated with this client.