# Table of Contents

## List of Figures



## List of Tables

# Chapter 10: Advanced Topics on Select Data Concepts

In This Chapter

- Understand the requirements related to
  - Route and Timetable Header
  - Transfers: Transfer Cluster and Event Connection
  - Block (from Service Branch)
  - Route Grouping
- Learn how to apply these data concepts in the SDP

Advanced Topics

Advanced Topics describes SDP Data Concepts that are optional and designed for specific downstream applications. Topics include the following data concepts:

- Route and Timetable Header
- Transfer Cluster and Event Connection
- Block (for buses)
- Route Grouping

## Why Advanced Topics on Select Data Concepts

The topics in this chapter are designated as *advanced* because the data concepts support key downstream applications that are not central to the initial scope of the Schedule Data Profile. They include topics such as:

- Timetable Header—captures the heading row of a public timetable. This is typically a manually derived set of data.
- Transfer Cluster and Event Connection—the places and times when connections may be made. The places refer to the Transfer Cluster, and the times refer to the Event Connection. These concepts differ from geographically and temporally calculated trip connections in that they are ***recommended*** transfer locations and times.
- Block (for buses only)—the daily work of a transit bus in revenue service. This is just a different way of grouping trip information.
- Route Grouping—a special designation for a route or group of routes. Again, a different way of grouping trip and pattern information.

Although these concepts may provide significant benefit to key downstream applications, most of these data concepts have not yet been validated in a demonstration environment or using real-world examples.

The data concepts may be found in different branches of the SDP XML hierarchy. Figure 10-1 highlights the position of these concepts within the SDP Schema. Under the Agency Registration branch may be found both the Route Grouping and the Timetable Header as a child element of the Route element. The Service branch includes both the Block (and Block Time) and Event Connection elements. The Transit Gazetteer includes the Transfer Cluster (and ConnectionSeg) element.

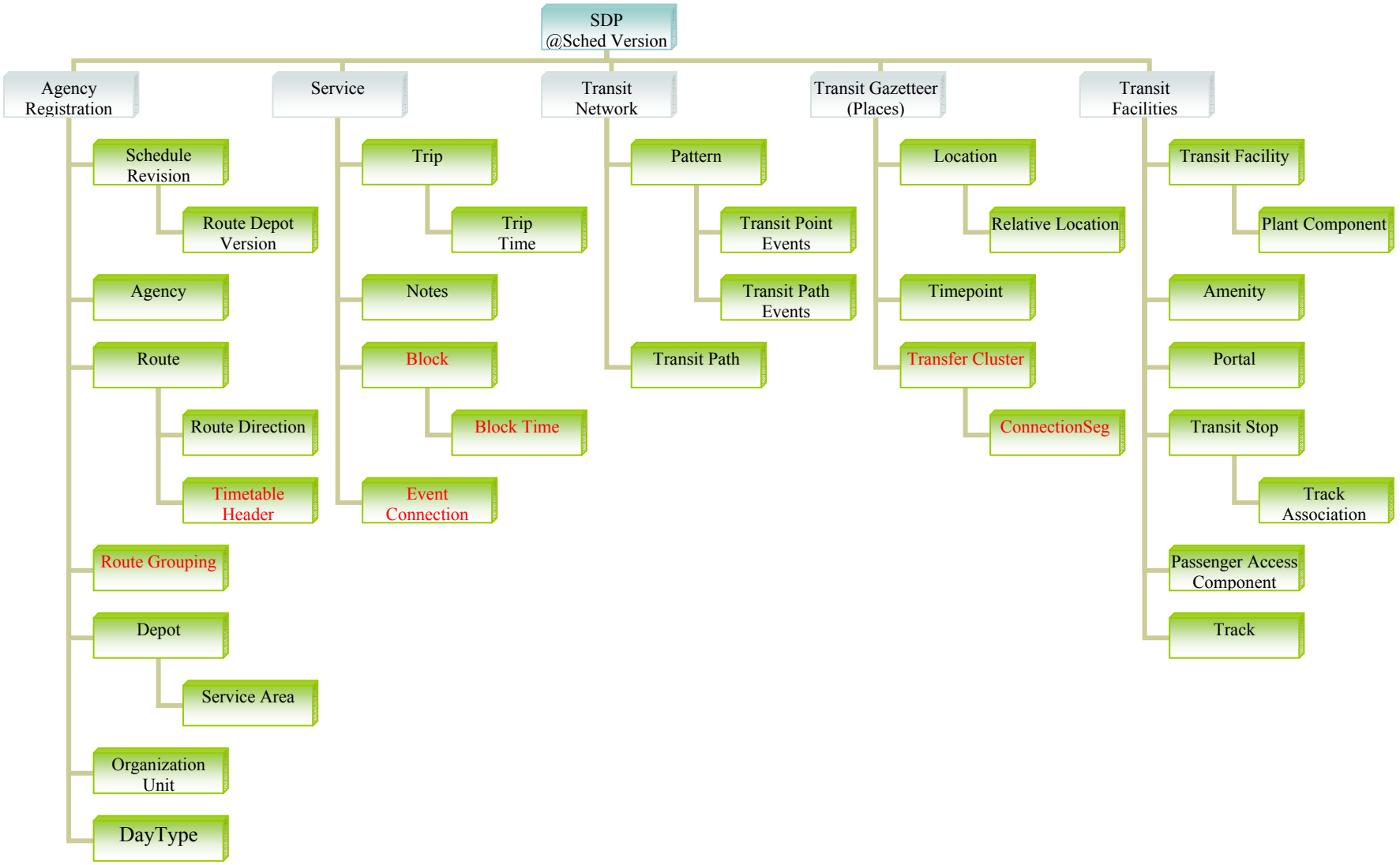



**Figure 10-1: SDP XML Schema Organization with Advanced Topics highlighted in "red"**

# Section 10.1: Route and Timetable Header

In This Section

- Learn how to apply the Timetable Header Data Concept.
- Learn how to apply the Timetable Header element of the Route data concept.

**Timetable Header Definition**
The Timetable Header defines the header row for a table that contains public schedule information for a specific route. There should be one or two Timetable Header records (one for each direction) associated

## What is the Timetable Header Data Concept?

A route timetable that is used by the public is an edited version of the full schedule. The public timetable summarizes each trip of all the patterns that are oriented in the same direction for a particular service type. An agency with three types of services—weekday, Saturday and Sunday—would generate three public timetables in each direction for each route. The Timetable Header data concept stores a header row of stop places for each route direction. The consolidation of the patterns of the same route direction is not necessarily a trivial exercise, neither is the reduction of timepoints or stops along each pattern. This exercise is typically a manual process performed by transit staff at each service change. The TimetableHeader data concept is a place to store summary header information with the schedule data so that when the pattern's events change, the header information may be updated simultaneously.

## Timetable Header XML Schema Element Description

The Timetable Header concept is stored as a table in the CDRM. It is related directly to the Route it represents. This relationship is also passed on to the SDP XML Schema. The timetableHdr element is nested in the Route element (see Figure 10-2). Although optional in the Route structure, the timetableHdrList may contain many collections of headers. The rules related to applying the TimetableHeader Structure are as follows:

- Zero or more TimetableHeaderStructures may be included in the Route.
- The routeDirection should correspond to a routeDirection element in the routeDirectionList contained in the same Route.
- The timepointList is a locationID and its corresponding publicLocationDescription.

A number of assumptions may be inferred by downstream users or applications. For example:

- When present, it may be assumed that two timetableHdrList elements are included, one for each direction.
- When an agency prefers to display information about the stop, a stopID should be added to the timepointList.
- publicLocationDescription, although also contained in the Location table, is included in the TimetableHeader for the efficient processing of a downstream application that generates public timetables.
- This information may be derived (and stored at a later time) by an editing tool such as the Timetable Publisher application.

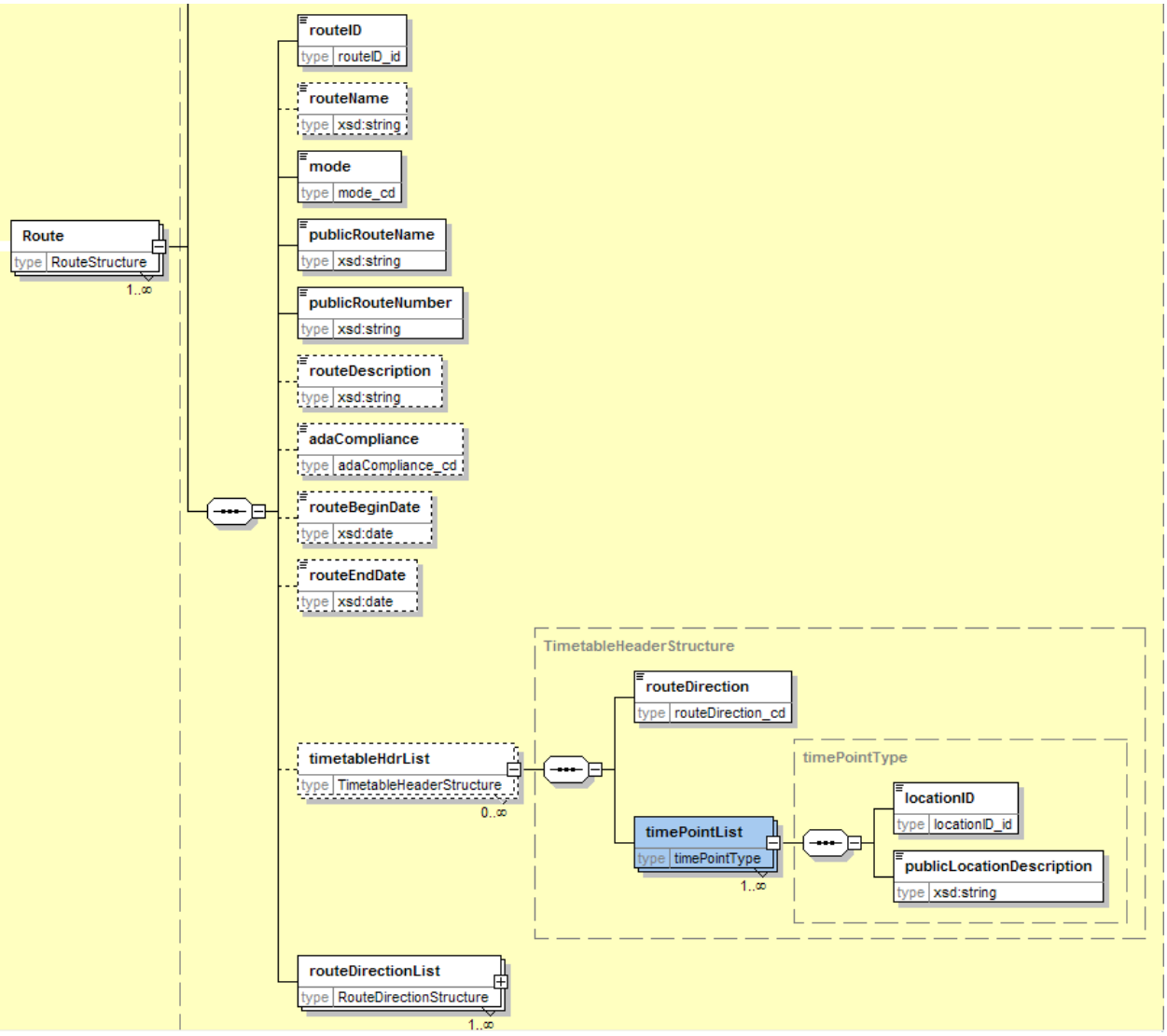



**Figure 10-2: SDP XML Schema Fragment for Timetable Header**

## Detailed Data Descriptions and Guidance for the Timetable Header Element

This section provides data type information and guidance associated with TimetableHeader element described above. The guidance for each element is consolidated into a table with the following column headings: Requirement Status (M for mandatory and O for optional), the element name, the data type and guidance related to the element. The guidance attempts to bring additional clarity to the data definition. The first column of each table identifies the baseline requirements as driven by the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet the application's data needs. The XML Schema element name corresponds to the related CDRM entities and attributes descriptions (although the capitalization and spacing may differ slightly). The type may refer to a native XML type, or a declared type in the XML schema. The Guidance column is called "Questions to Ask." These questions direct the analyst to a similar or equivalent concept in their

own schedule data set. In addition, some comments describe the impact of the data structures on the XML document deployment.

Table 10.1-1 incorporates TimetableHeader Guidance and Table 10.1-2 incorporates timePointType Guidance.

**Table 10.1-1: Timetable Header Guidance**

| | Element Name | Type | Questions to Ask |
|---|---|---|---|
| M | routeDirection | routeDirection_cd | Use the routeDirection elements included in the routeDirectionList of the Route element. This element is used to select the trip/pattern that is oriented in the same direction. |
| M | timePointList | List of timePointType | As part of the data script, a timetable header may be generated. Generation requires review of each route's route direction, an aggregation of all events along all the patterns in the same direction, selection of the places along the pattern that are designated as timetable header transit places (locationID), and selecting the publicLocationDescription that matches the locationID (in the Location table). |

**Table 10.1-2 TimePointType Guidance**

| | Element Name | Type | Questions to Ask |
|---|---|---|---|
| M | locationID | locationID_id | A place that matches the Trip Time in order for a match with the Trip Time locationID to be made. |
| M | publicLocationDescription | string | A public name for the location that may be displayed on the timetable. This location description may be the same as a similarly named element in the Location table. |

## Examples of Timetable Header from a SDP XML Document

The Timetable Header documents the column headings for a route timetable. One Header record is associated with each direction. This example describes how the XML Schema will incorporate the Timetable Header record. The example shows how Suffolk County Transit (SCT) may define their Timetable Header for Route S29 Babylon serving the Walt Whitman Mall.

SCT may describe route-related information in the Route element and nest the TimetableHeader element under timetableHdrList (see SDP XML Schema description in Figure 10-2). The values for route are defined in Table 10.1-3.

**Table 10.1-3: Example of Route Table**

| Route | Value |
|---|---|
| routeID | S29 |
| routeName | S29 Babylon |
| publicRouteName | S29 Babylon—Walt Whitman Mall |
| publicRouteNumber | S29 |
| routeDescription | S29 Babylon-Walt Whitman Mall |
| routeBeginDate | 20041214 <!—yyyymmdd—> |
| routeEndDate | <!—null—> |
| timetableHdrList | <!—see below; only one direction is included in this example—> |
| routeDirection | [first, North], [second, South] |

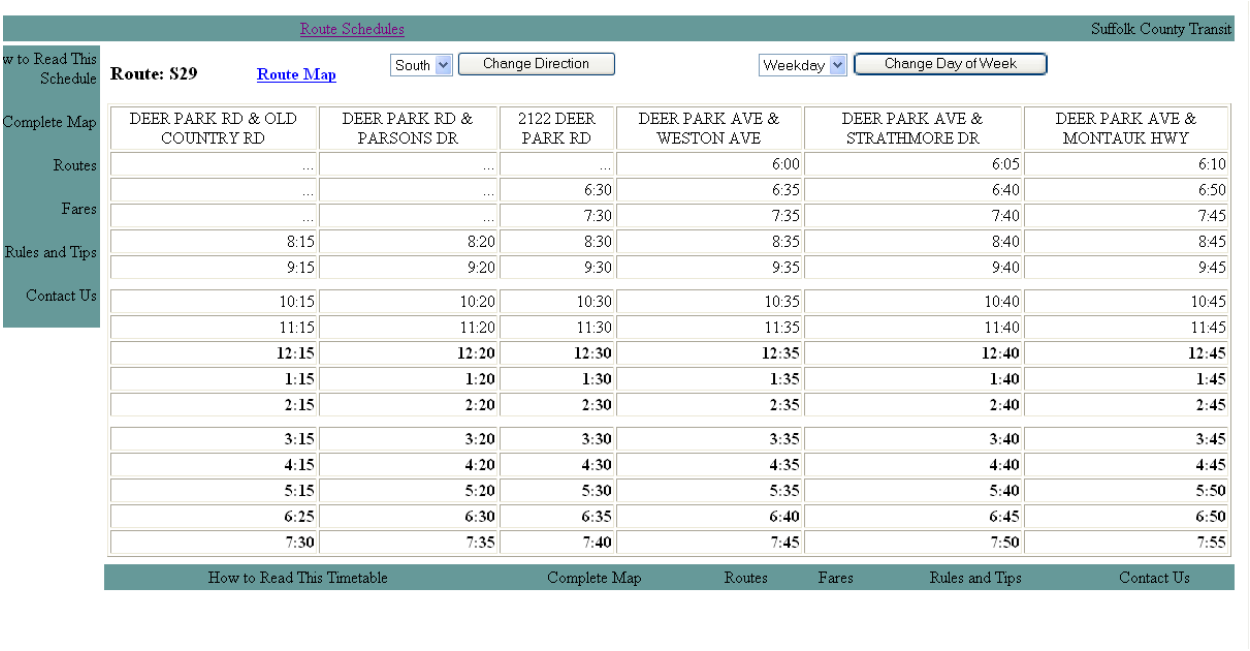
The Timetable Header derives its values from other elements in the schedule. For example, routeDirection should match the value in the Route/Route Direction, and the timepointList's locationID should match a locationID in a Route/Pattern/TransitPointEvent of a Pattern designated with the same route direction. For example, Route S29 may include five patterns, two patterns oriented in the first direction and three patterns in the second direction. Only the transitPointEvent elements that are associated with a Pattern specifying the second direction are considered for the header.

Given the requirements, SCT might specify a Timetable Header element that is described as an SDP XML Document fragment below.

```
<timetableHdrList>
    <routeDirection>second</routeDirection>
    <timePointList>
        <locationID>1204</locationID>
        <publicLocationDescription> Deer Park Rd and Old Country Rd </publicLocationDescription>
    </timePointList>
    <timePointList>
        <locationID>1210</locationID>
        <publicLocationDescription> Deer Park Rd and Parsons Dr </publicLocationDescription>
    </timePointList>
    <timePointList>
        <locationID>1217</locationID>
        <publicLocationDescription>2122 Deer Park Rd </publicLocationDescription>
    </timePointList>
    <timePointList>
        <locationID>1223</locationID>
        <publicLocationDescription> Deer Park Ave and Weston Ave </publicLocationDescription>
    </timePointList>
    <timePointList>
        <locationID>156</locationID>
        <publicLocationDescription> Deer Park Ave and Strathmore Dr </publicLocationDescription>
    </timePointList>
    <timePointList>
        <locationID>984</locationID>
        <publicLocationDescription> Deer Park Ave and Montauk Hwy </publicLocationDescription>
```

```
    </timePointList>
</timetableHdrList>
```

The Timetable Header for Route S29 in the second direction (i.e., South) for weekday schedules is illustrated in Figure 10-3. Note that the Dynamic Timetable Generator—DTG (the tool that generated the timetable illustrated in Figure 10-3) reads the header information, accesses the Trip information, and then matches the Trip's tripTime locationID to the header's locationID. A location that is included in the Trip but does not match the timetable header will be passed over. In addition, the application separated the service information by Route Direction (i.e., "Change Direction") and Day Type (i.e., "Change Day of Week").



Route Schedules — Suffolk County Transit

w to Read This Schedule | Complete Map | Routes | Fares | Rules and Tips | Contact Us

**Route: S29** Route Map — South — Change Direction — Weekday — Change Day of Week

| DEER PARK RD & OLD COUNTRY RD | DEER PARK RD & PARSONS DR | 2122 DEER PARK RD | DEER PARK AVE & WESTON AVE | DEER PARK AVE & STRATHMORE DR | DEER PARK AVE & MONTAUK HWY |
|---|---|---|---|---|---|
| ... | ... | ... | 6:00 | 6:05 | 6:10 |
| ... | ... | 6:30 | 6:35 | 6:40 | 6:50 |
| ... | ... | 7:30 | 7:35 | 7:40 | 7:45 |
| 8:15 | 8:20 | 8:30 | 8:35 | 8:40 | 8:45 |
| 9:15 | 9:20 | 9:30 | 9:35 | 9:40 | 9:45 |
| 10:15 | 10:20 | 10:30 | 10:35 | 10:40 | 10:45 |
| 11:15 | 11:20 | 11:30 | 11:35 | 11:40 | 11:45 |
| **12:15** | **12:20** | **12:30** | **12:35** | **12:40** | **12:45** |
| **1:15** | **1:20** | **1:30** | **1:35** | **1:40** | **1:45** |
| **2:15** | **2:20** | **2:30** | **2:35** | **2:40** | **2:45** |
| **3:15** | **3:20** | **3:30** | **3:35** | **3:40** | **3:45** |
| **4:15** | **4:20** | **4:30** | **4:35** | **4:40** | **4:45** |
| **5:15** | **5:20** | **5:30** | **5:35** | **5:40** | **5:50** |
| **6:25** | **6:30** | **6:35** | **6:40** | **6:45** | **6:50** |
| **7:30** | **7:35** | **7:40** | **7:45** | **7:50** | **7:55** |

How to Read This Timetable | Complete Map | Routes | Fares | Rules and Tips | Contact Us

**Figure 10-3: Example of Dynamic Timetable Generator Timetable for Suffolk County Transit**

# Section 10.2: Transfer Cluster and Event Connection Data Concepts

In This Section

- Learn about Transfer data concept requirements and issues. Learn about how the SDP supports Transfers. The key data concepts include
  - Transfer Cluster
  - Connection Segment, and
  - Event Connection
- Learn how to apply Transfer Cluster, Connection Segment and Event Connection to describe a Transfer.

**Transfer Cluster Definition**
A transfer cluster is a collection of transit stops where transfer between stops is convenient and scheduled.

**Connection Segment Definition**
A Connection Segment is a linear path allowing transit riders to move from one Transit Stop to another. The segment may be defined as a walking path, bike path, escalator or other modal connection. Attributes include distance, fromStop, toStop and connection instructions. Accessibility information in the form of obstacleTypes may optionally be provided for Connection Segments.

**Event Connection Definition**
An Event Connection is the provision for a connection between two route/trips at a trip time event. Connection types include [protected, recommended, scheduled].

## Issues Related to Transfers

Transfer opportunities are easily derived by downstream applications by finding stops that are in close proximity and times where trips appear to allow for convenient connections. The requirements manifested in the SDP provide information on scheduled and recommended transfer locations and opportunities. A derived trip plan may not always generate a transfer that is convenient or viable, particularly if the path between trips is enabled by temporal events (e.g., gate open during business hours only, use of train in middle track as "platform bridge"). As such, the SDP data concepts allow for an agency to set recommendations for preferred transfer locations and opportunities. The SDP transfer provision includes a data concept in which a walking path (see Connection Segment Definition) may be defined that specifies access preferences.

## Requirements for Transfer Cluster Data Concept

The requirements that drive the Transfer Cluster Data Concept are described in Table 10.2-1

**Table 10.2-1 Transfer Cluster Data Concept Requirements**

| # | Category | Requirements |
|---|---|---|
| 1 | Unique identification, naming conventions, and references | • Transfer Clusters may contain both internal and public names and numbers. They are typically known through their Public Location Description (as described as publicLocationDescription in the Location Table). |
| 2 | Geometry and spatial characteristics | • A Transfer Cluster may share a location with one of the stops which is "clustered."<br>• The location usually refers to a "centroid" or generalized location reference. |

**Table 10.2-1 Transfer Cluster Data Concept Requirements**

| # | Category | Requirements |
|---|---|---|
| 3 | Policy driven transfers | • A Transfer Cluster is defined for a set of stops where one or more agencies may coordinate schedule transfers between trips. The transfer may be based on a set of policies such as safety of boarding areas, connectivity, timed transfers, etc.<br>• As a policy driven transfer, there may be special directions described for traversing from one boarding area to another. Several Connection Segments may be described between each stop pair. The connection may be accessible via elevator, ramp, etc. These are defined as a directed connection (from stop and to stop). |
| 4 | Associations | • A Transfer Cluster is associated with more than one Transit Stops (boarding area).<br>• A Transfer Cluster may be indicated by the Trip Time timeEventType as a coordinated transfer location. |

## XML Schema Descriptions for TransferCluster and ConnectionSeg

The Transfer Cluster data concept includes a child element that describes one or more segments that connect the transfer locations (i.e., ConnectionSeg). Figure 10-4 depicts the SDP XML Schema fragment for TransferCluster and connectionList (instance of a ConnectionSeg).

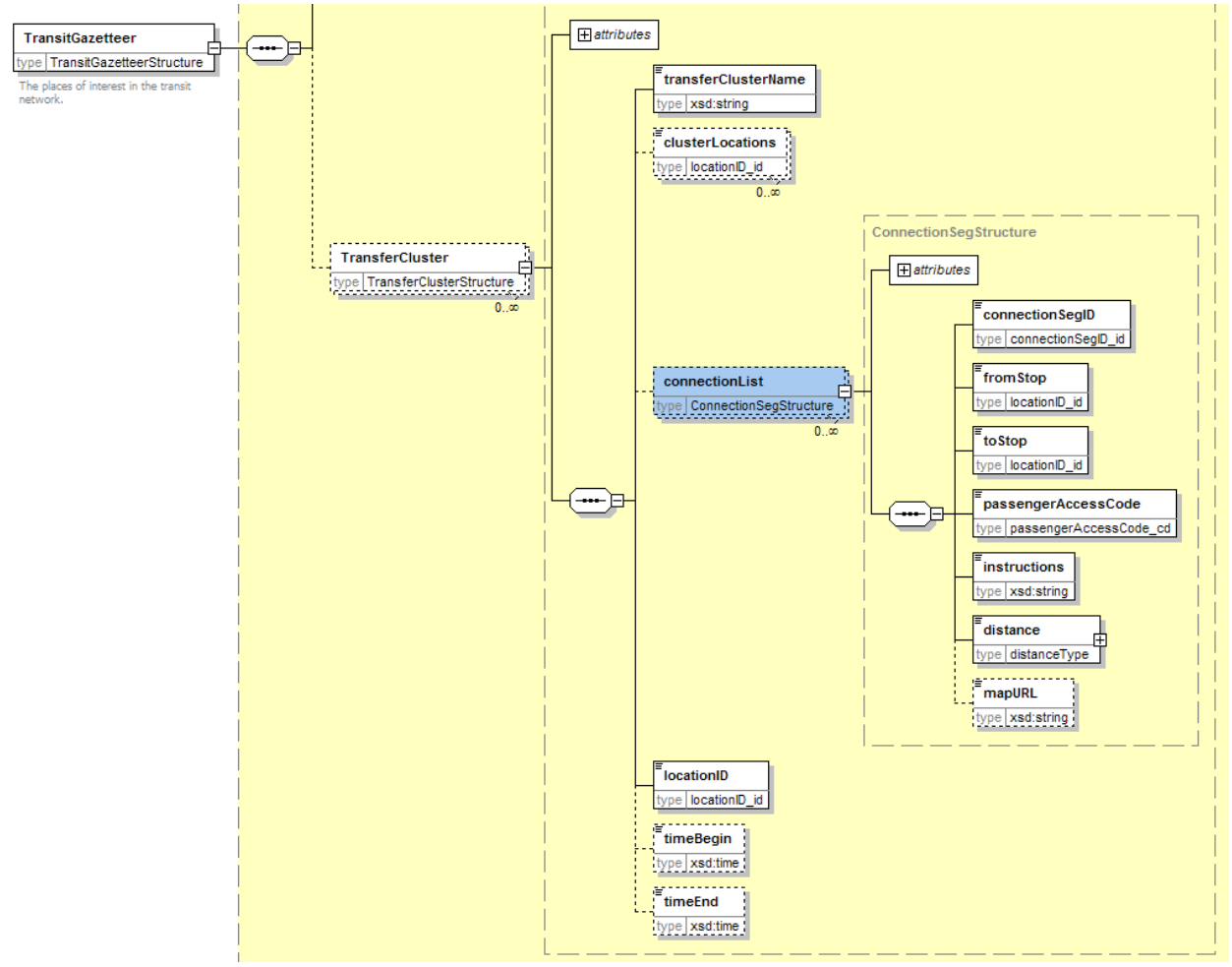



**Figure 10-4: TransferCluster and ConnectionSeg Elements from SDP XML Schema**

Specifically, the following rules were applied to build the XML Schema:

- The transferClusterName should be unique, and consistently spelled and capitalized. It is used by the EventConnection to reference the TransferCluster record.
- Transfer stops (referenced by locationID) are included by a series of clusterLocations.
- locationID describes the TransferCluster "centroid"; alternatively, the locationID may be a reference to one of the stops or a facility location.
- The TransferCluster may be temporal, and as such it is initiated and closed at specified begin (timeBegin) and end (timeEnd) times.
- The connectionList is a set of Connection Segments. Each Connection Segment is unique in each Transfer Cluster, as such, it may include an unique identifier (connectionSegID).
- The Connection Segment is a path between two Transit Stops (fromStop to toStop).
- The path may be described as an accessible or non-accessible path (passengerAccessCode)
- The instructions element contains human readable directions for the path from its origin to destination.
- The distance units may be specified as meters or feet.
- The mapURL is a link to a map or graphic of the connection path.

## Conceptual Data Reference Model of the Event_Connection

The Event_Connection entity links two trips at a scheduled trip time pair. The transfer opportunity may be scheduled, coordinated or guaranteed (i.e., the connectionType code values). They may be from the same route, different routes or even different agencies.[1] The Event_Connection references the Transfer Cluster by its name (transferClusterName). The relationship among the Event_Connection entities is illustrated in Figure 10-5.

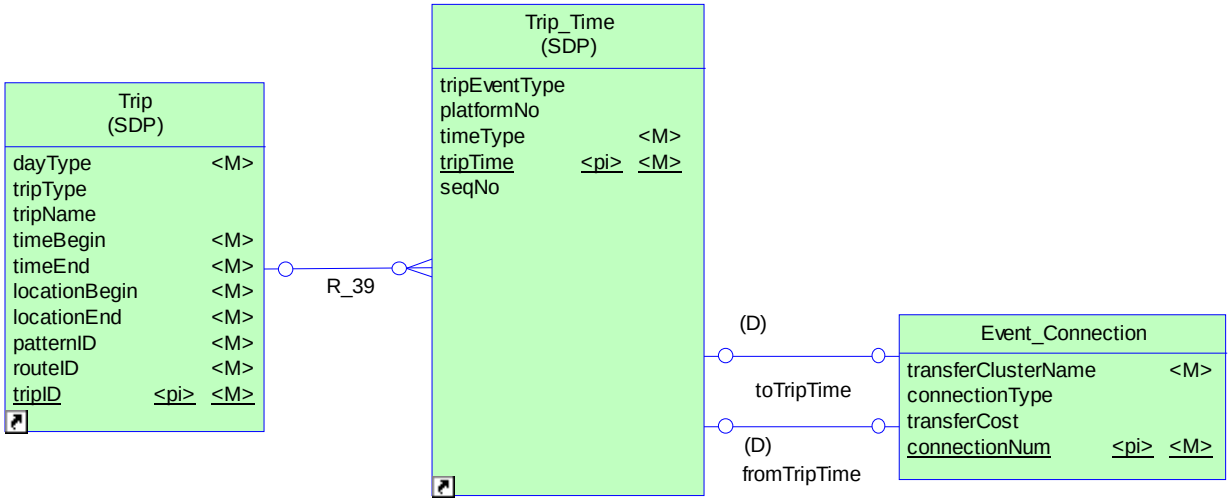



**Figure 10-5: Event_Connection Conceptual Data Reference Model**

Specifically, the Event_Connection CDRM represents:

A coordinated transfer (Event_Connection) may be designated between a pair of trips at a trip time from each trip. The time to transfer (e.g., 2 minute, 5 minute) is assumed from the fromTripTime to the toTripTime tripTimes. The location of each point and the path connecting them are described by the Transfer Cluster and referenced by the transferClusterName. The transferCost value is a description of the value or fare media that is required for this transfer. The connectionType values indicate if the trip is scheduled, coordinated or guaranteed.

The Event Connection table may include (from and to) agencyID when a multi-agency transfer is described.

## XML Schema Description for EventConnection

The CDRM associates the Event_Connection with the Trip_Time. Because best practices in developing an XML Schema recommends that the schema limit the nesting elements (and flatten the hierarchy), the Event Connection is described as a child directly from the Service Branch. The rules applied to implement the EventConnection in the SDP XML Schema from the CDRM are as follows:

- Although the CDRM did not include foreign keys, the schema fragment does through the use of the XML Schema constraint -- KEYREF. The EventConnection must reference the

[1] A logical model that supports interagency connections should provide additional identifying keys into the Event_Connection entity to support connections between different agency trips.

Trip and Route from which the selected tripTimes derive. These are designated as "from" (fromRouteID, fromTripID) and "to" (toRouteID, toTripID).

- tripName (from and to) are included for readability and for use by rail transit to store the primary train number.
- agencyID (from and to) should be included if the transfer occurs between two agencies. However, the SDP XML Document assumes the route XML Document contains only a single Agency's schedule.
- connectionType is an enumerated type.
- If included, transferClusterName must match the precise spelling and capitalization of the TransferCluster element's transferClusterName.

The XML Schema fragment for EventConnection is depicted in Figure 10-6.

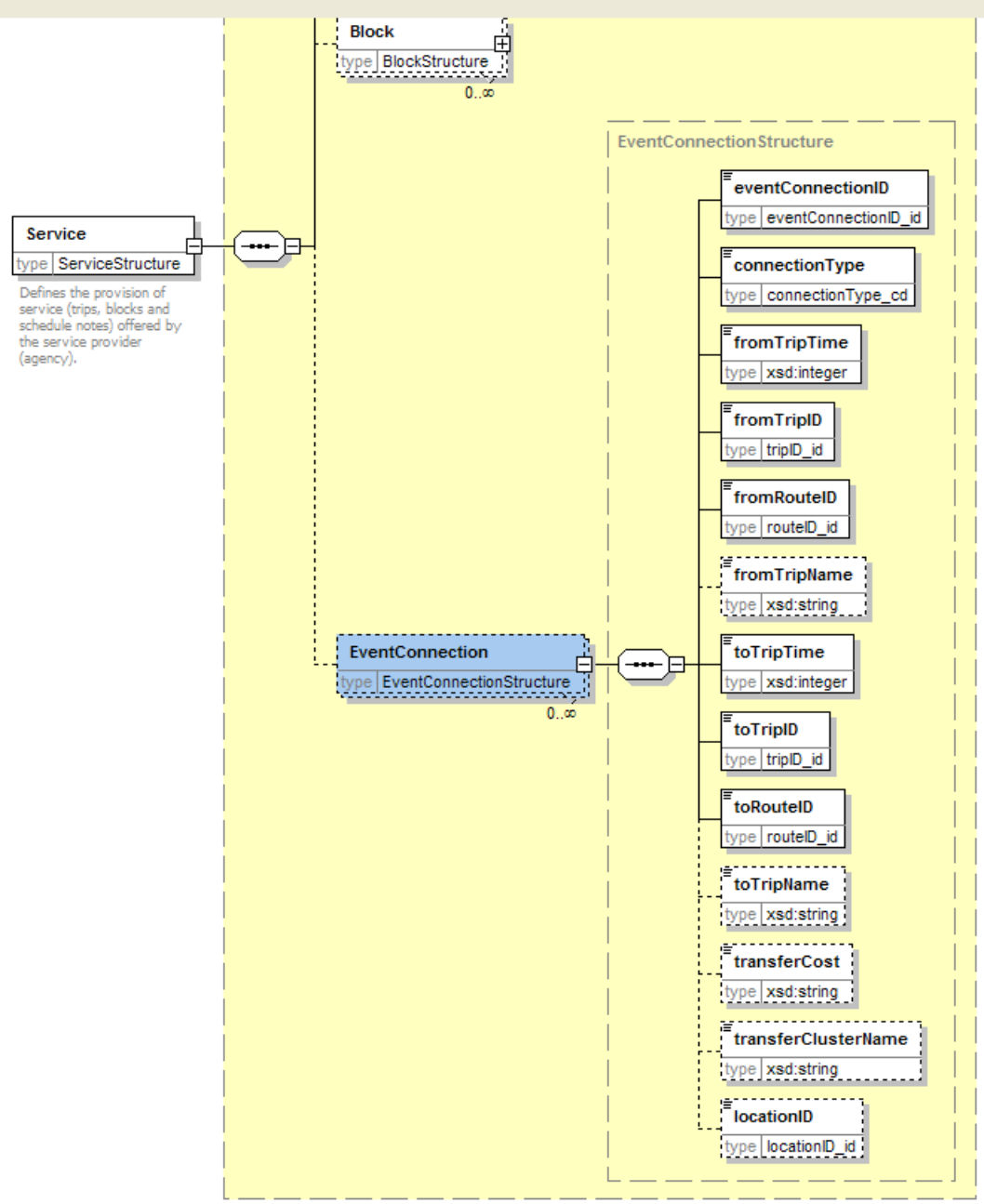



**Figure 10-6: EventConnection SDP XML Schema Fragment**

The schema fragment that depicts a guaranteed connection between two Metro North Railroad trains—Train 1530 (tripID 144711) on the New Heaven line and Train 1730 (tripID 144784) on the New Canaan line (at Stamford Station) is listed below. The transfer time between trips is 9 minutes (50220 sec – 49860 sec).

```
<EventConnection>
    <eventConnectionID>100</eventConnectionID>
    <connectionType>guaranteed</connectionType>
    <fromTripTime>49860</fromTripTime>
```

```
        <fromTripID>144711</fromTripID>
        <fromRouteID>3</fromRouteID>
        <toTripTime>50220</toTripTime>
        <toTripID>144784</toTripID>
        <toRouteID>4</toRouteID>
        <transferCost>0</transferCost>
        <locationID>124</locationID>
    </EventConnection>
```

## Detailed Data Descriptions and Guidance for the TransferCluster, ConnectionSeg and EventConnection

This section describes the format and guidance associated with TransferCluster, ConnectionSeg and EventConnection elements in the data concepts described above. The guidance for each element is consolidated into a table with the following column headings: Requirement Status (M for mandatory and O for optional), the element name, the data type and guidance related to the element. The guidance attempts to bring additional clarity to the data definition. The first column of each table identifies the baseline requirements as driven by the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet the application's data needs. The XML Schema element name corresponds to the related CDRM entities and attributes descriptions (although the capitalization and spacing may differ slightly). The type may refer to a native XML type, or a declared type in the XML schema. The Guidance column is called "Questions to Ask." These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the XML document deployment.

The following tables provide guidance on TransferCluster (see Table 10.2-2), Connection Segment (Table 10.2-3), and EventConnection (Table 10.2-4).

**Table 10.2-2: Transfer Cluster Guidance**

| Required | Element Name | Type | Questions to Ask |
|---|---|---|---|
| **Transfer Cluster** | | | |
| M | transferClusterName | string | A unique name by which the transfer cluster may be known. For example LIRR Jamaica Transfer |
| O | clusterLocations | list of locationID_id | These include the list of locations that are connected by this cluster. This may include multiple stops, facilities and landmarks. |
| O | connectionList | ConnectionSeg | Descriptions of the paths between each transfer pair may be described using the connection segment. |
| M | locationID | locationID_id | This is a location identifier that designates the location of the transfer cluster. It may be the location of a facility like LIRR Jamaica Station or a shared facility like Mineola Intermodal Terminal. |

**Table 10.2-2: Transfer Cluster Guidance**

| Required | Element Name | Type | Questions to Ask |
|---|---|---|---|
| O | timeBegin | time | Time transfer location is in service or accessible |
| O | timeEnd | time | Time transfer location is not in-service or is inaccessible. |
| O | @effectiveDate | date | [attribute] The date the record was inserted. |
| O | @endDate | date | [attribute] The date the record expires or becomes obsolete. |

**Table 10.2-3: Connection Segment Guidance**

| Required | Element Name | Type | Questions to Ask |
|---|---|---|---|
| **Connection Segment** | | | |
| M | connectionSegID | connectionSegID_id unique | An identifier that is stored by the SDP/TSDEA. |
| M | fromStop | locationID_id | The originating location. |
| M | toStop | locationID_id | The destination location. |
| M | passengerAccessCode | passengerAccessCode_cd | The type of access along the path. |
| M | instructions | string | A description of the directions along the connection path. |
| M | distance | distanceType | The distance between the origin and destination. An attribute, "units", may be designated for either feet or meters. The default units is ‘feet’. |
| O | mapURL | string | A link to a map showing the origin/destination; if this is generated, then it could be a web service that links the information from this element. |
| O | @effectiveDate | date | [attribute] The date the record was inserted. |
| O | @endDate | date | [attribute] The date the record expires or becomes obsolete. |

**Table 10.2-4: EventConnection Guidance**

| Required | Element Name | Type | Questions to Ask |
|---|---|---|---|
| **Event Connection** | | | |
| M | eventConnectionID | eventConnectionID_id | Insert a unique number or index in this field. Recommend a unique, sequential number be inserted for every event connection in the system throughout its lifecycle. |
| M | connectionType | connectionType_cd | This is a classification for connection type. These include: [scheduled, guaranteed, recommended]. |
| M | fromTripTime | integer | The time (in schedule time) of the first trip's trip time (arrival or departure) to the stop. |

**Table 10.2-4: EventConnection Guidance**

| Required | Element Name | Type | Questions to Ask |
|---|---|---|---|
| | | | This time should match the originating trip's Trip.TripTime.tripTime value. |
| M | fromTripID | tripID_id | The tripID of the originating trip. |
| M | fromRouteID | routeID_id | The routeID associated with the originating trip. |
| O | fromTripName | string | A name used by the originating trip. |
| M | toTripTime | integer | The time (in schedule time) of the connecting trip's trip time (arrival or departure) to the stop. This time should match the connecting trip's Trip.tripTimeList.tripTime value. |
| M | toTripID | tripID_id | The tripID of the connecting trip. |
| M | toRouteID | routeID_id | The routeID associated with the connecting trip. |
| O | toTripName | string | A name used by the connecting trip. |
| O | transferCost | string | A description of the transit fare cost or media needed to transfer from the first to second trip. |
| O | transferClusterName | string | The name of the transfer cluster as defined in TransferCluster element (see TransitGazetteer branch). |
| O | locationID | locationID_id | The location of the connection. This may be either the originating trip or the connecting trip. When convenient, the Transfer Cluster or Transit Facility locationID should be used. |

## Examples of TransferCluster and ConnectionSegment

Because the Transfer Cluster encapsulates recommended transfer points and the paths between them, the Transfer Cluster may describe specific paths associated between two transit stops in a Transit Facility. In the example below, the TransferCluster element shows the TransferCluster between the Long Island Railroad platforms and Long Island Bus Terminal in the Mineola Intermodal Center (MIC).

More fully described in Chapter 9 on Transit Facilities, the MIC contains two facilities and three passenger access components (see Tables 10.2-5 for TransitFacility and Table 10.2-6 for selected PlantComponent element). There are two key connection segments for transfer from the LIB terminal to the LIRR platforms, these include the transfer to the eastbound platform and transfer to the westbound platform.

**Mineola Intermodal Center**

The transfer cluster composed of the Mineola Intermodal Center is described in Table 10.2-5 and its' Plant Components are described in Table 10.2-6.

**Table 10.2-5: Mineola Intermodal Center TransitFacility Element with Two Nested TransitFacility Elements**

| *Element name/values* | LIB | LIRR | MIC |
|---|---|---|---|
| *effectiveDate* | 2006-10-16 | 2006-10-16 | 2006-10-16 |
| *endDate* | 9999-12-31 | 9999-12-31 | 9999-12-31 |
| *transitFacilityID* | MNLA | Mineola | MIC |
| *locationID* | MNLA | Mineola | MIC <master location> |
| *facilityName* | Mineola Bus Terminal | Mineola Station | Mineola Intermodal Center |
| *facilityDescription* | Mineola Bus Terminal | LIRR station serving Port Jefferson Branch | The Mineola Intermodal Center is on the south side of the LIRR track in the vicinity of Third Street between Third and Fourth Avenues and north of Old Country Road in Mineola. |
| *owner* | MTA | MTA | MTA |
| *isPartOf* | true | true | False |
| *partOf* | MIC | MIC | |

**Table 10.2-6: Selected List of Plant Component List in MIC TransitFacility**

| Element name/Values | | | | |
|---|---|---|---|---|
| *plantCompID* | *componentID* | *plantCompType* | *plantCompDescription* | *locationDescription* |
| 4 | passAccessID >2 | PassengerAccessComponent | North South Bridge with elevator linking Platform A, and Platform B, Bus Terminal and parking garage | N-S Bridge at west side of facility; entrance in Parking Garage. |
| From LIRR Transit Facility Plant Components | | | | |
| 1 | stopID > MineolaA | TransitStop | Platform A | Inbound |
| 2 | stopID> MineolaB | TransitStop | Platform B | Outbound |

The TransferCluster SDP XML Document fragment below includes the clusterLocations between the two facilities; these elements use location identifiers (locationID). The connectionList describes two paths between two stops (stopID), the first between the bus terminal (stop MNLA1) and Platform A, and the second between the bus terminal and Platform B. *The intention of the "instruction" element is that the description will be shown to a customer.* The map may be a compressed graphic or a link to a path and graphic reference, as suggested by the example. [The information in the SDP XML Document fragment is based on a photographs of the facility.]

```
<TransferCluster effectiveDate="2007-10-16" endDate="9999-12-31">
        <transferClusterName>Mineola Intermodal Center</transferClusterName>
        <clusterLocations> MNLA</clusterLocations>
```

```
        <clusterLocations>Mineola</clusterLocations>
        <connectionList effectiveDate="2007-10-16" endDate="9999-12-31">
                <connectionSegID>1</connectionSegID>
                <fromStop>MNLA1</fromStop>
                <toStop> MineolaA </toStop>
                <passengerAccessCode>elevator</passengerAccessCode>
                <instructions>”Take elevator on west side relative to bus berth 1 to
North-South Bridge.  Cross bridge, and take elevator on Rail Station side.”</instructions>
                <distance units="feet">100</distance>
                <mapURL>www.mineolagraphic.net/mic.jpg</mapURL>
        </connectionList>
        <connectionList effectiveDate="2007-10-16" endDate="9999-12031">
                <connectionSegID>2</connectionSegID>
                <fromStop>MNLA1</fromStop>
                <toStop>MineolaB</toStop>
                <passengerAccessCode>elevator</passengerAccessCode>
                <instructions> Take elevator on north side relative to berth 1 to
North-South Bridge.  Cross bridge, and take elevator on Rail Station side.”</instructions>
                <distance units="feet">140</distance>
                <mapURL> www.mineolagraphic.net/mic.jpg</mapURL>
        </connectionList>
        <locationID>MIC</locationID>
</TransferCluster>
```

### Example of EventConnection with TransferCluster

In this example, the EventConnection shows a coordinated connection between a LIRR train and LIB trip at the Mineola Intermodal Center. [The example is derived from the Roosevelt East scheduled connection table.]

| | TripTime locationID | tripTime | tripID | routeID |
|---|---|---|---|---|
| From | Mineola | 23340 [6:29 AM] | 1608 [604] | Port Jefferson |
| To | MNLA | 23460 [6:31 AM) | 2963 | 3484 [N22] |

```
<EventConnection>
        <eventConnectionID>1</eventConnectionID>
        <connectionType>coordinated</connectionType>
        <fromTripTime>23340</fromTripTime>
        <fromTripID>1608</fromTripID>
        <fromRouteID>Port Jefferson</fromRouteID>
        <toTripTime>23460</toTripTime>
        <toTripID>2963</toTripID>
        <toRouteID>3484</toRouteID>
        <transferCost>Pay both LIRR and LIB fares</transferCost>
        <transferClusterName>Mineola Intermodal Center</transferClusterName>
        <locationID>mic</locationID>
</EventConnection>
```

# Section 10.3: Block Data Concept

In This Section

- Learn about the Block Data Concept.
- Learn how to apply the elements in the Block data concept.
- Learn about the need for and application of the Block concept.

**Block Definition**
The daily sequence of revenue and nonrevenue trips assigned to a transit vehicle in revenue service from pull-out to pull-in.

**Block Trip Time Definition**
The path on which a transit vehicle in revenue service travels during the course of a day. Block Event Times (blockTimes) are part of an ordered set of events from beginLocation to endLocation. Each blockTime is associated with a specific event along the block, and is associated with one trip. The Block Event Time may be any type of transit feature type: timepoint, transit stop or other event (e.g., fare set change, headsign change) that is referenced by a Location (locationID).

## Issues Related to the Block Data Concept

Bus and rail use the term *block* to mean very different concepts. In rail, block refers to a section of track, while in bus it means the daily work of a transit vehicle in revenue service from garage pull-out to pull-in. Although rail has a similar concept assigning train "work", a train's journey is closely tied to coordinating switches (e.g., controls) and assigning tracks, and consequently is not related to the types of information included in the SDP. On the other hand, many bus transit applications use the block structure to organize and predict bus arrival time at a stop for bus riders.

## Requirements for Block Data Concept

The requirements associated with the Block Data Concept are listed in Table 10.3-1.

**Table 10.3-1: Block Requirements**

| # | Category | Requirements |
|---|---|---|
| 1 | Uniqueness and identity | • A block is referenced by a unique identifier. The block identifier may be a combination of other identifiers such as route/run.<br>• A block is assigned to a specific day or day type during a valid schedule version. |
| 2 | Temporal and Spatial Representation | • A block is a directed journey traversed by a transit vehicle while in revenue service. The journey is described by an ordered set of waypoints and times from the time a vehicle leaves its vehicle base until it returns to the vehicle base. [Note: during the course of a journey, a vehicle may be in revenue service, stopped, or traveling between revenue trips.]<br>• At a minimum, a block is composed of two (origin and destination) or more sequence of locations (corresponding to pattern descriptions) and associated passing times (corresponding to trip descriptions), and any additional locations that represent events along the journey.<br>• The "way points" consist of a unique, ordered sequence of transit point events such as timepoints, transit stops and events, and additional geo-located points that may be necessary to sufficiently describe the physical path.<br>• A block has an origin and destination that typically correspond to a Transit Facility such as a vehicle base, depot or garage. The first and |

**Table 10.3-1: Block Requirements**

| # | Category | Requirements |
|---|---|---|
| | | last (origin and destination) points constitute the termini of the block.<br>• A block is typically assigned to one or more service or day types. |
| 3 | Alternative Geo-spatial composition as a set of transit paths | • Alternatively, a Block may be represented by a topologically ordered set of transitPaths (time point intervals, route segments).<br>• The transitPath must include both tripTimes and transitPointEvents.<br>• Each transitPath should be unique (for that day or day type) and associated with a trip identifier.<br>• Note: the SDP transformation process will require that the points and times describing the transit path be reassigned to an ordered set of waypoints and times. |
| 4 | Sequential Waypoints and Passing Times | • Duplicate, sequential waypoints are allowed if they represent different types of passing points. For example, consecutive trips will share the same destination and origin, however, the destination will be described as an arrival time, and the origin of the next trip will be described as a departure time.<br>• Duplicate, sequential waypoints are mandatory to differentiate the completion and commencement of different trips. |
| 5 | PTV assignment | • Schedulers typically assign a block to a vehicle type based on several characteristics supported by its fleet, such as capacity, seating, wheelchair lifts.<br>• The actual assignment of the PTV is usually not made until the operator is dispatched and pulls out of the depot or vehicle base. |

## Conceptual Data Reference Model Description of Block

These requirements may be implemented in a CDRM depicted in Figure 10-7. The data model description follows.

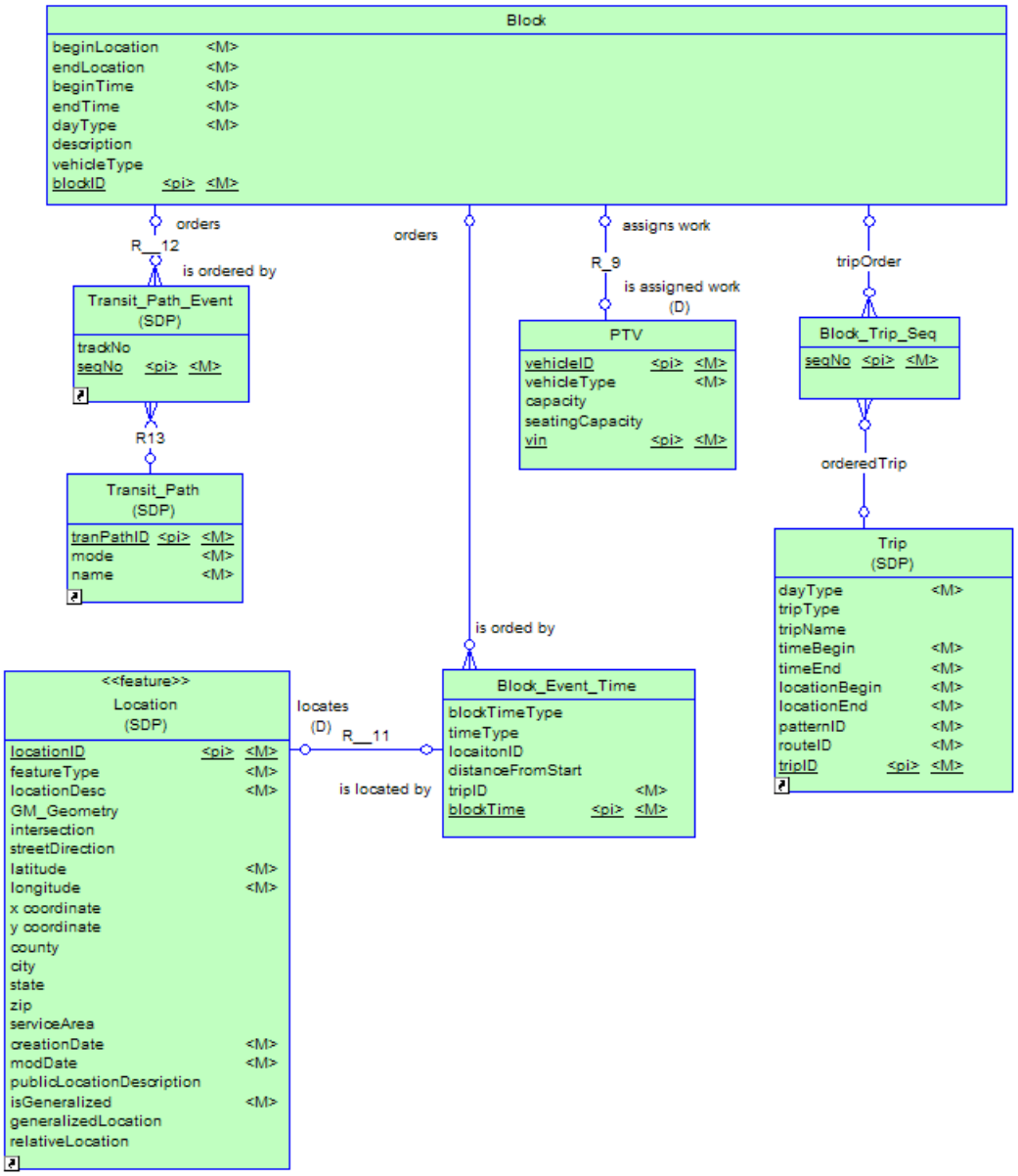



**Figure 10-7: Block Data Model**

A Block is described as the journey taken by a public transport vehicle in revenue service from when and where it leaves a transit facility until and where it returns to a transit facility. The actual PTV assignment may be made based on the required vehicleType. The path on which the vehicle travels is described by the BlockEventTimes, a series of coordinates, also linear references (distanceFromStart) from the beginLocation. The Block Event Times are ordered by blockTimes. Each blockTime is associated with a specific event along the block. Each Block Event Times is associated with one trip. The Block Event Time may be any type of transit feature type: timepoint, transit stop or other event (e.g., fare set change, headsign change) that is referenced by a Location.

## Block Fragment of XML Schema Model

A number of rules and assumptions were used to implement the CDRM Block Data Concept as the Block and nested blockTimeList elements in the SDP XML Schema fragments (as shown in Figures 10-8 and 10-9). These include the following:

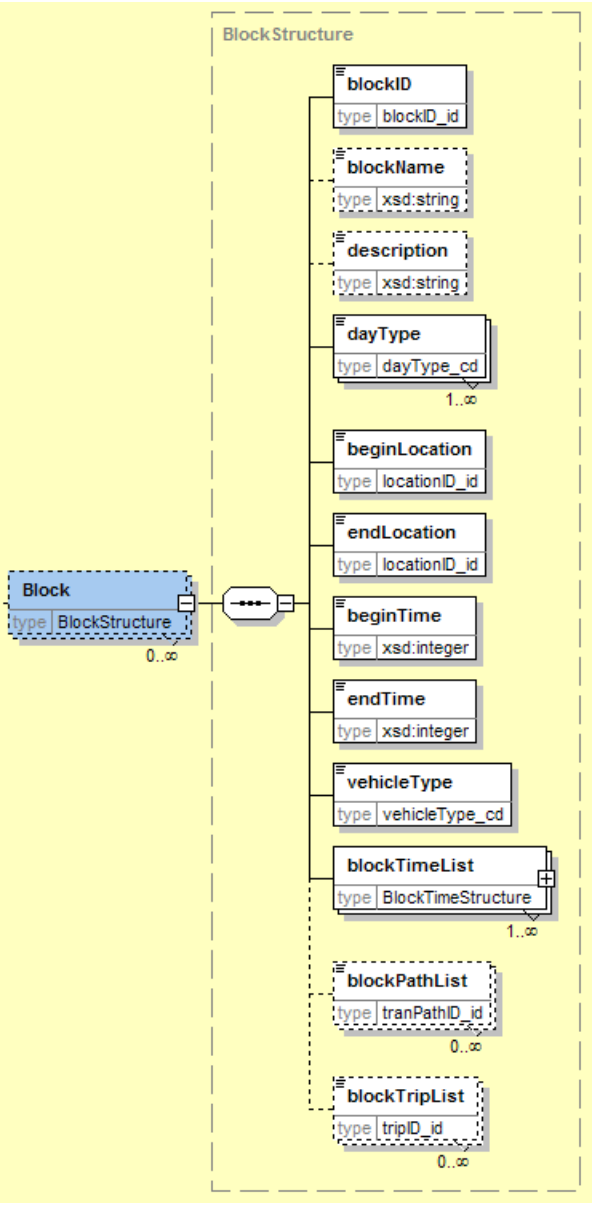



**Figure 10-8: Block Element Fragment in SDP XML Schema**

- blockID is unique across the data set;
- blockID, beginLocation, endLocation, beginTime, endTime, vehicleType and multiple blockTimeList nested elements must be included in Block structure.
- beginLocation and endLocation are coincident with the first and last blockTimeList (from blockTimeStructure) locationID element; beginTime and endTime are the same as the first and last blockTimeList blockTime element.
- A block is associated with the day type(s) on which it is assigned.

- blockPathList includes a sequential list of transitPathID.
- blockTripList includes a sequential list of tripID.

Note that the SDP does not include an actual transit vehicle assignment which is enabled by the CDRM Block data concept (see "PTV is assigned work to a Block"). The blockTimeList is equivalent to Block_Event_Time, the blockPathList is equivalent to Transit_Path_Seq, and the blockTripList is equivalent to Block_Trip_Seq.

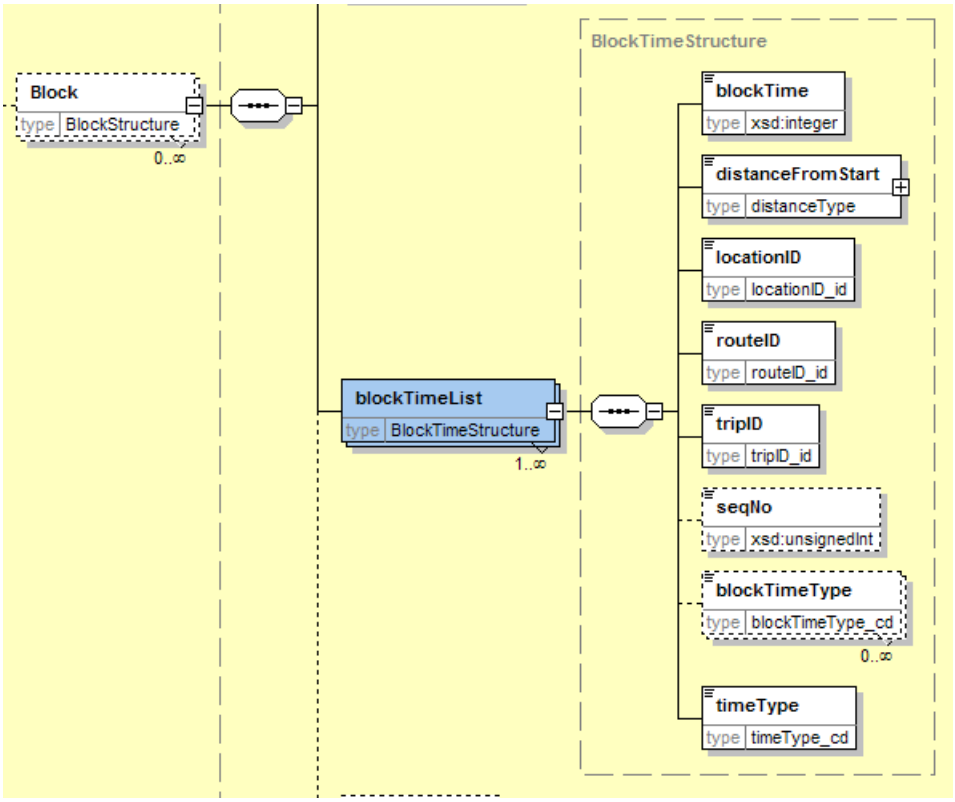



**Figure 10-9: Nested blockTimeList Element Fragment in SDP XML Schema**

## Detailed Data Descriptions and Guidance for Block

This section describes the format and guidance associated with Block and BlockTimes elements in the data concepts described above. The guidance for each element is consolidated into a table with the following column headings: Requirement Status (M for mandatory and O for optional), the element name, the data type and guidance related to the element. The guidance attempts to bring additional clarity to the data definition. The first column of each table identifies the baseline requirements as driven by the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet the application's data needs. The XML Schema element name corresponds to the related CDRM entities and attributes descriptions (although the capitalization and spacing may differ slightly). The type may refer to a native XML type, or a declared type in the XML schema. The Guidance column is called "Questions to Ask." These questions direct the analyst to a similar or

equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the XML document deployment.

**Table 10.3-2: Block Guidance**

| Required | Element Name | Type | Questions to Ask |
|---|---|---|---|
| **Block (for bus only)** | | | |
| M | blockID | blockID_id UNIQUE | Insert a unique number or character string for the blockID. Some agencies use a combination of the route and run numbers. |
| O | blockName | string | A block name is a character string that refers to the block. It is usually used for internal purposes. |
| O | description | string | A block description may be used to describe the vehicle journey and turn directions. It is typically used for internal purposes. |
| M | dayType | dayType_cd [1..∞] | The day type or day types to which it is assigned. Usually, the block assumes the day types of the service (trips) which compose its parts. |
| M | beginLocation | locationID_id | The beginLocation is typically the vehicle's pull-out location. It should correspond to the first timepoint of the first trip, (typically, the first trip is a non-revenue trip). |
| M | endLocation | locationID_id | The endLocation is typically the vehicle's pull-in location. It should correspond to the last timepoint of the last trip (typically, the last trip in a non-revenue trip). |
| M | beginTime | integer | The beginTime is typically the vehicle's pull-out time. It should correspond to the first timepoint of the first trip, (typically, the first trip is a non-revenue trip). |
| M | endTime | integer | The endTime is typically the vehicle's pull-in time. It should correspond to the last timepoint of the last trip (typically, the last trip in a non-revenue trip). |
| M | vehicleType | vehicleType_cd | The vehicleType is a general type of vehicle. This element supports internal fleet management strategies and in some cases customer-facing applications. The acceptable values mirror those in TCIP. They include additional values that may not be relevant for a block:<br>▪ twentyfiveRevenue<br>▪ thirtyRevenue<br>▪ fortyRevenue<br>▪ articulated<br>▪ cng<br>▪ lng<br>▪ supervisor |

**Table 10.3-2: Block Guidance**

| Required | Element Name | Type | Questions to Ask |
|---|---|---|---|
| | | | ▪ police<br>▪ towTruck<br>▪ shelterService<br>▪ van<br>▪ passengerVehicle<br>▪ lightRail<br>▪ commuterRail<br>▪ heavyRail<br>▪ aircraft<br>▪ ferry<br>▪ transitPolice<br>▪ otherPolice |
| M | blockTimeList | List of BlockTimes | These are the Block Times along the block. They may be derived from the sequence of trips by ordering the tripTimes into the blockTimes. |
| O | blockPathList | List of tranPathID_id | The blockPathList describes an ordered list of the physical path traversed by a transit vehicle in a block. |
| O | blockTripList | List of tripID_id | The blockTripList describes list of the vehicle trips passed in the order they occur during the block. |
| **Block Times (for buses only)** | | | |
| M | blockTime | integer | The blockTime is the schedule time, that is, a signed integer of seconds past midnight of a schedule day. |
| M | distanceFromStart | distanceType: float | The value of (driven) distance from the beginning of the Block (blockLocation) to this Block Time location (locationID). The element includes an attribute that describes the unit of measure for the distance value. The allowed attributes include "feet" or "meters." The default enumeration type is “feet.” |
| M | locationID | locationID_id | The location identifier associated with the Block Time. |
| M | routeID | routeID_id | The route identifier tow which the trip is associated. The routeID provides an identifying reference for the tripID. |
| M | tripID | tripID_id | The trip identifier (tripID) to which the blockTime is typically associated. If the Block Time terminates one trip and begins another, then the timeType value should differentiate between an arrival/departure or begin/end Trip point. |
| O | seqNo | unsignedInt | The sequence number is a unique unsigned integer that is used to order the Block Time elements within a Block. The blockTime is ordered, although not sequentially numbered. |

**Table 10.3-2: Block Guidance**

| Required | Element Name | Type | Questions to Ask |
|---|---|---|---|
| | | | This value is available for use to associate the points of multiple temporal or linear paths. |
| O | blockTimeType | blockTimeType_cd | The blockTimeType designates key points that occur in a Block. These include:<br>▪ revenue<br>▪ pullIn<br>▪ pullOut<br>▪ beginInterDead<br>▪ endInterDead |
| O | timeType | timeType_cd | The timeType enumerated type uses the same values used in the Trip Time element. |

## Example of Block

The block is typically used for operational purposes, the blocking process assigns revenue rolling stock to planned service. In addition to estimating the number of revenue vehicles needed to deploy scheduled service, the block is also used to track the vehicle throughout the day. To this end, the block is used by many Automatic Vehicle Location (AVL) and real-time tracking systems to collect schedule adherence performance measures and to provide customers with estimated time of arrival predictions. The SDP block data concept is focused on supporting planned service data for these two functions.

## Long Island Bus Block/Run 0101 for Weekday Service

This example shows the Block Structure for Block/Run 101 which covers Route 3210 (see Route example in Section 4.3).

**Table 10.3-3: Trip Set for Run/Block ID for 101 From LI Bus Data**

| RouteID | TripID |
|---|---|
| 3210 | 4575 |
| 3210 | 4576 |
| 3210 | 4577 |
| 3210 | 4578 |
| 3210 | 4579 |
| 3210 | 4580 |
| 3210 | 4581 |
| 3210 | 4582 |
| 3210 | 4583 |

```
<Block>
        <blockID>0101</blockID>
        <blockName>101</blockName>
        <dayType>weekday</dayType>
        <beginLocation>6</beginLocation>
        <endLocation>4918</endLocation>
        <beginTime>24180</beginTime>
```

```
        <endTime>51480</endTime>
        <blockTimeList>
                <blockTime>24180</blockTime>
                <distanceFromStart units="feet">0.00001</distanceFromStart>
                <locationID>6</locationID>
                <routeID>3210</routeID>
                <tripID>4575</tripID>
                <blockTimeType>pullOut</blockTimeType>
                <timeType>departure</timeType>
        </blockTimeList>
<!--more goes here -->
        <blockTimeList>
                <blockTime>51480</blockTime>
                <locationID>4918</locationID>
                <routeID>3210</routeID>
                <tripID>4583</tripID>
                <blockTimeType>pullIn</blockTimeType>
                <timeType>arrival</timeType>
        </blockTimeList>
        <blockTripList>4575</blockTripList>
        <blockTripList>4576</blockTripList>
        <blockTripList>4577</blockTripList>
        <blockTripList>4578</blockTripList>
        <blockTripList>4579</blockTripList>
        <blockTripList>4580</blockTripList>
        <blockTripList>4581</blockTripList>
        <blockTripList>4582</blockTripList>
        <blockTripList>4583</blockTripList>
</Block>
```

# Section 10.4: Route Grouping Data Concept

In This Section

- Explore the Route Grouping Data Concepts.
- Learn when and how to apply the elements in the Organization Unit and Depot data concepts.

**Route Grouping Definition**
A realignment of a collection of patterns and/or trips known by a common name or number. The Route Grouping may be a "curtain" route, public timetable route, scheduler's route, or other collection of revenue service.

## Purpose of Route Grouping in the SDP

Similar to a Route, a Route Grouping is a collection of patterns, trips or both. A route grouping is assembled for a specific purpose other then a typical Route definition. For example, NYCT publishes for the public Routes S61 and S91, although their schedulers generate a Route 6191, and the bus carry a headsign displaying the "curtain" Route S61/S91. In this case, the routeID may be 6191, the public route number may be S61/S91. A Route Grouping record may aggregate all Route S61 patterns and trips and another may aggregate all S91 patterns and trips. Alternatively, the Route element may be indexed by 61 and 91, and the Route Grouping may aggregate the two routes into the S61/S91. Typically, the SDP recommends that the public-facing Route number is used in the Route element, and the ones used for internal and operational purposes use the RouteGrouping element designation.

## Requirements for Route Grouping Data Concept

The requirements associated with the Route Grouping Data Concept are listed in Tables 10.4-1.

**Table 10.4-1: Route Grouping Requirements**

| | | |
|---|---|---|
| 1 | Definition | • A line or route grouping is a collection of patterns or trips that are grouped together for a particular purpose.<br>- A set of patterns or trips may be grouped together operationally in order to closely coordinate their scheduled headways along a common alignment or carriageway.<br>- Public timetables group routes together to communicate the frequency of service of routes that share a common corridor before they branch.<br>o There are other types of groups and some of these are listed in Route Type |
| 2 | Uniqueness and identity | • A route grouping is referenced by a unique identifier which is distinct from a Route. Each route grouping identifier should be uniquely defined within the schedule version, revision and activation/deactivation dates. The route grouping identifier may be a combination of other identifiers such as two route identifiers or a partition of a single route identifier.<br>• A route grouping may be identified by its duration during a schedule version such as its activation and deactivation dates. |

## Conceptual Data Reference Model Description for Route Grouping

The Route Grouping requirements from Table 10.4-1 may be implemented in the CDRM as depicted in Figure 10-10. The data model description (as excerpted from the *Functional Requirements* document) follows.

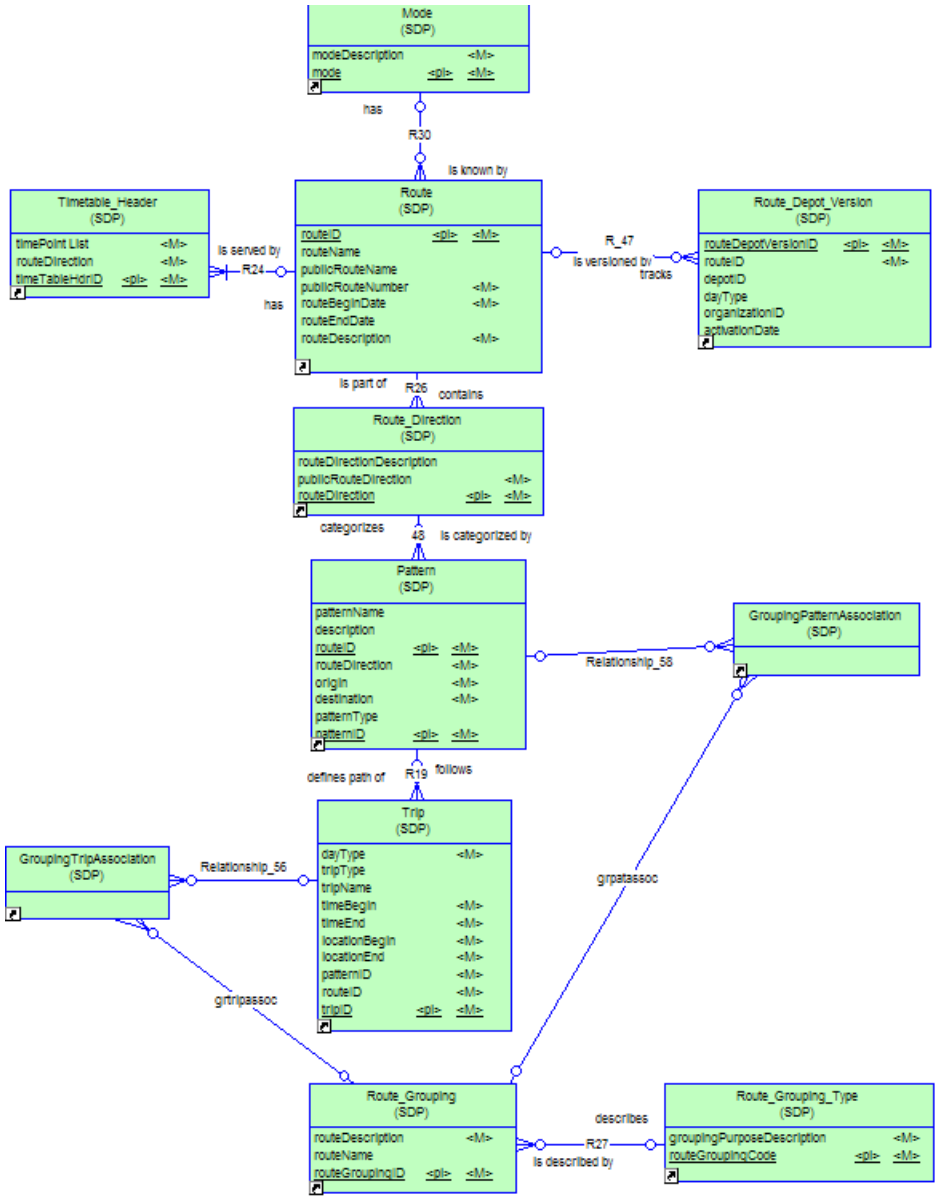



**Figure 10-10: Route Grouping Data Model**

"A Route is a collection of patterns classified by their route direction in scheduled service with a common identifier. As shown, Route is an entity associated with patterns and related trips. One or more directed patterns may optionally be associated with a Route, and one Route categorizes one or more patterns through a route direction. One or more trips may optionally be associated with each pattern...

"A Route is valid during its designated period (routeBeginDate to routeEndDate) that falls within a valid schedule version.

"Similar to a Route, a Route Grouping is a collection of patterns, trips or both. A route grouping is assembled for a specific purpose other then a typical Route definition."

### Route Grouping Excerpt of XML Schema Model

The following rules and assumptions were used to implement the CDRM Route Grouping Data Concept as the RouteGrouping element in the SDP XML Schema:

- A unique identifier, routeGroupingID, is associated with the RouteGrouping element.
- The element includes XML attributes to designate the placement and deactivation dates of the element.
- The element is associated with a routeName and sometimes with a routeDescription (either for operational purposes or for the public).
- The routeGroupingCode describes the purpose for the route grouping. The code values include: curtain, public, line. Other codes may be designated by code values above an integer value of 5.
- The groupingSetList builds the associations between Route_Grouping and its related Patterns (GroupingPatternAssocation) and Trips (GropuingTripAssociation).
  - Through the groupingSetList, a list of applicable route-pattern (patternSet) and/or related route-trip identifiers (tripSet) are associated with each RouteGrouping element.

Figure 10-11 depicts the RouteGrouping element portion of the SDP XML Schema as it was derived from the CDRM.

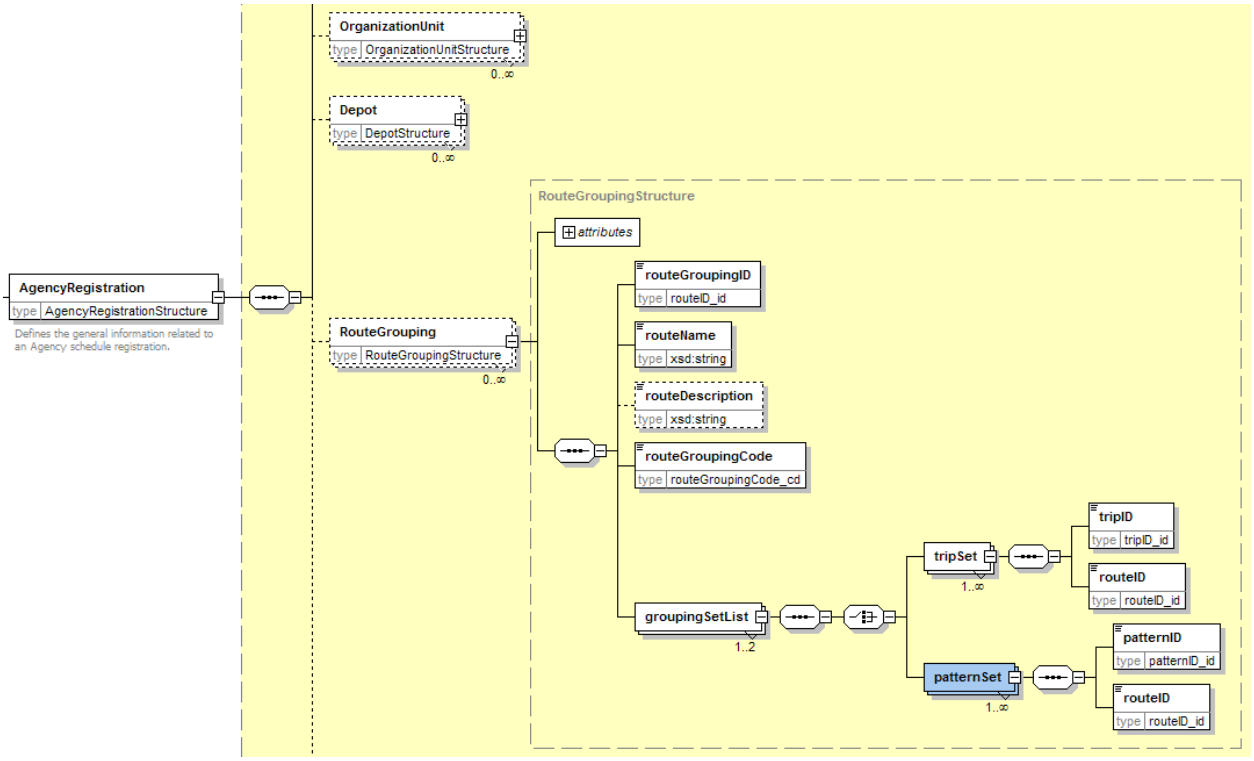



**Figure 10-11: SDP XML Schema Fragment of Route Grouping**

## Detailed Data Formats and Guidance for the RouteGrouping Element

This section describes the description and guidance associated with RouteGrouping element in the data concepts described above. The guidance for each element is consolidated into Table 10.4-2 with the following column headings: Requirement Status (M for mandatory and O for optional), the element name, the data type and guidance related to the element. The guidance attempts to bring additional clarity to the data definition. The first column of each table identifies the baseline requirements as driven by the SDP XML Schema version 1.0. A downstream application may further restrict these requirements in order for the data set to meet the application's data needs. The XML Schema element name corresponds to the related CDRM entities and attributes descriptions (although the capitalization and spacing may differ slightly). The type may refer to a native XML type, or a declared type in the XML schema. The Guidance column is called "Questions to Ask." These questions direct the analyst to a similar or equivalent concept in their own schedule data set. In addition, some comments describe the impact of the data structures on the XML document deployment.

**Table 10.4-2: Route Grouping Guidance**

| Required | Element Name | Type | Questions to Ask |
|---|---|---|---|
| RouteGrouping structure | | | |
| M | routeGroupingID | routeID_id UNIQUE | The routeGroupingID is unique in the schedule version for both the Route Grouping and Route elements. |
| M | routeName | string | If the route grouping element is used for the public, then routeName may also be used for as a public name like the Route element – publicRouteName. Otherwise, it is used by schedulers or for other internal and operational purposes. The name should be formatted for the purpose of the route grouping. |
| O | routeDescription | string | Similar to the routeName, the routeDescription is typically used for internal and operational purposes. The description should be formatted for the purpose of the route grouping. |
| M | routeGroupingCode | routeGroupingCode_cd | The routeGroupingCode is used to describe the purpose of the route. Curtain and Public are public-facing route identifiers and Line is a scheduler's artifact. |
| M | groupingSetList | [1] tripSet; and<br>[2] patternSet | Both elements (and no more then the two elements) should be included in the groupSetList. |
| tripSet element | | | |
| M | tripID | tripID_id | The tripID is associated with an existing route designated in the Trip element. The trips listed in this set should correspond to the patterns identified in the patternSet |
| M | routeID | routeID_id | The routeID is needed to distinguish the trip identified above. |
| patternSet element | | | |
| M | patternID | patternID_id | The patternID is associated with an existing route designated in the Pattern element. |
| M | routeID | routeID_id | The routeID is needed to distinguish the pattern identified above. |

## Examples Using Route Grouping

There are several types of route groupings. The major categories that will be described in this section are two scenarios that exist in the downstate NY region. The first scenario is when a single route is separated into two routes for a singular purpose like branding an express versus local service in the same scheduled route. The second alternative is when two routes are merged into a single route perhaps for operational purposes or because they share a common trunk. New York City Transit and New Jersey Transit have several examples that cover these scenarios.

Scenario #1: Merging two routes into a single route

New Jersey Transit Route 78 timetable carries two designated scheduled routes: #78 and #378. #78 is the Newark/Penn Station to Secaucus-Harmon Cove via Plaza Drive and Meadowland Parkway, and #378 goes directly from Newark/Penn Station to Secaucus-Harmon Cove (also called express service). In the scheduling system, the holiday service may be scheduled differently, for example, Route 378 is not operated on the Friday after Thanksgiving, MLK Day and President's Day. This Route Grouping scenario is described in Table 10.4-3 below.

**Table 10.4-3: Route Grouping Example for NJ TRANSIT Route 78 Timetable**

| **Route Grouping** | Record 1 |
|---|---|
| routeGroupingID | 78 |
| routeDescription | Serving Newark Secaucus Harmon Meadow Harmon Cove |
| RouteName | 78 Newark Secaucus |

| **Route Grouping Type** | Type associated with routeGrouping ID = 78 |
|---|---|
| routeGroupingCode | 1 |
| groupingPurposeDescription | Similar origin and destination (local and express service) subject to different holiday schedules; may be used to generate timetable. |

| **Partial Set of Trips Associated with routeGroupingID=78** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| tripID | tripType | dayType | timeBegin | timeEnd | locBegin | locEnd | routeID* | pattID* |
| 7801 | 1 | 1 | 5:57 | 6:54 | 1001 | 2156 | 78 | 7811 |
| 7802 | 1 | 1 | 6:35 | 7:32 | 1001 | 2156 | 78 | 7811 |
| 7803 | 1 | 1 | 7:00 | 7:57 | 1001 | 2156 | 78 | 7811 |
| 7804 | 1 | 1 | 7:20 | 8:17 | 1001 | 2156 | 78 | 7811 |
| 7805 | 1 | 1 | 7:50 | 8:47 | 1001 | 2156 | 78 | 7811 |
| 7806 | 1 | 1 | 8:20 | 9:24 | 1001 | 2156 | 78 | 7811 |
| 7807 | 1 | 1 | 8:50 | 9:47 | 1001 | 2156 | 78 | 7811 |
| 7808 | 1 | 1 | 9:20 | 10:23 | 1001 | 2156 | 78 | 7811 |
| 7809 | 1 | 1 | 11:00 | 12:02 | 1001 | 2156 | 78 | 7811 |
| 7810 | 1 | 1 | 13:00 | 13:57 | 1001 | 2156 | 78 | 7811 |
| 7811 | 1 | 1 | 15:00 | 15:56 | 1001 | 2156 | 78 | 7812 |
| 37801 | 2 | 1 | 16:07 | 16:39 | 1001 | 2156 | 378 | 37811 |
| 37802 | 2 | 1 | 16:35 | 17:07 | 1001 | 2156 | 378 | 37811 |

*needed for physical database; patternIDs, locationBegin and locationEnd are arbitrary numbers.
Where tripType 1=local and tripType 2=express
Where dayType 1=weekday

Scenario #2: Composite Route Separated in Route Grouping

In the second scenario, NYCT may separate a single route into two route grouping for the purpose of branding local and limited stop services. In the example below (see Table 10.4-4) the composite route 6292 is divided into two routes in order to brand the limited stop route with its own public route number and name.

**Table 10.4-4: Route Grouping Example for NYCT Composite Route S62/S92**

| **Route Grouping** | Record 1 | Record 2 |
|---|---|---|
| routeGroupingID | S62 | S92 |
| routeName | Local for Route S62/S92 | Limited Stop Service for S62/S92 |
| routeDescription | S62 Local Travis-St. George Ferry Terminal | S92 Limited Stop Travis St.-George Ferry Terminal |

| **Route Grouping Type** | routeGroupingCode | groupingPurposeDescription |
|---|---|---|
| *Type associated with routeGrouping ID = S92* | curtain | Express Service |
| Type associated with routeGrouping ID = S62 | curtain | Local Service |

| **Partial Set of Trips Associated with routeGroupingID=S62** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| tripID | tripType | dayType | timeBegin | timeEnd | locBegin | locEnd | routeID* | pattID* |
| 6201 | 1 | 1 | 00:20 | 12:51 | 2345 | 5657 | 6292 | 6202 |
| 6202 | 1 | 1 | 1:20 | 1:51 | 2345 | 5657 | 6292 | 6202 |
| 6203 | 1 | 1 | 2:20 | 2:51 | 2345 | 5657 | 6292 | 6202 |
| 6204 | 1 | 1 | 3:20 | 3:51 | 2345 | 5657 | 6292 | 6202 |
| 6215 | 1 | 1 | 3:50 | 4:21 | 2345 | 5657 | 6292 | 6202 |
| 6206 | 1 | 1 | 4:20 | 4:51 | 2345 | 5657 | 6292 | 6202 |
| 6207 | 1 | 1 | 4:50 | 5:21 | 2345 | 5657 | 6292 | 6202 |
| 6209 | 1 | 1 | 5:10 | 5:41 | 2345 | 5657 | 6292 | 6202 |
| 6208 | 1 | 1 | 5:25 | 5:56 | 2345 | 5657 | 6292 | 6202 |
| 6210 | 1 | 1 | 5:40 | 6:12 | 2345 | 5657 | 6292 | 6202 |
| 6211 | 1 | 1 | 5:55 | 6:29 | 2345 | 5657 | 6292 | 6202 |
| 6212 | 1 | 1 | 6:21 | 6:39 | 4556 | 5657 | 6292 | 6204 |
| 6213 | 1 | 1 | 6:41 | 6:59 | 4556 | 5657 | 6292 | 6204 |
| 6214 | 1 | 1 | 6:56 | 7:16 | 4556 | 5657 | 6292 | 6204 |
| 6205 | 1 | 1 | 7:11 | 7:32 | 4556 | 5657 | 6292 | 6204 |
| 6221 | 1 | 1 | 7:26 | 7:47 | 4556 | 5657 | 6292 | 6204 |
| 6216 | 1 | 1 | 7:41 | 8:02 | 4556 | 5657 | 6292 | 6204 |
| 6217 | 1 | 1 | 7:56 | 8:17 | 4556 | 5657 | 6292 | 6204 |
| 6218 | 1 | 1 | 8:11 | 8:32 | 4556 | 5657 | 6292 | 6204 |
| 6219 | 1 | 1 | 8:20 | 8:50 | 5766 | 5657 | 6292 | 6207 |
| 6220 | 1 | 1 | 8:21 | 9:00 | 2345 | 5657 | 6292 | 6202 |

| **Partial Set of Trips Associated with routeGroupingID=S92** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| tripID | tripType | dayType | timeBegin | timeEnd | locBegin | locEnd | routeID* | pattID* |
| 9201 | 2 | 1 | 6:10 | 6:47 | 2345 | 5657 | 6292 | 6203 |
| 9211 | 2 | 1 | 6:28 | 7:05 | 2345 | 5657 | 6292 | 6203 |
| 9202 | 2 | 1 | 6:43 | 7:21 | 2345 | 5657 | 6292 | 6203 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 9203 | 2 | 1 | 6:58 | 7:36 | 2345 | 5657 | 6292 | 6203 |
| 9204 | 2 | 1 | 7:12 | 7:51 | 2345 | 5657 | 6292 | 6203 |
| 9212 | 2 | 1 | 7:27 | 8:06 | 2345 | 5657 | 6292 | 6203 |
| 9205 | 2 | 1 | 7:42 | 8:21 | 2345 | 5657 | 6292 | 6203 |
| 9206 | 2 | 1 | 7:57 | 8:36 | 2345 | 5657 | 6292 | 6203 |

*needed for physical database;
patternIDs, locationBegin and locationEnd are arbitrary numbers.
Where tripType 1=local and tripType 2=express.
Where dayType 1=weekday

### Example of the RouteGrouping XML Excerpt

This is a XML fragment from Scenario 2 Composite Route Separated in Route Grouping, Record 1 from Table 10.4-4.

```
<RouteGrouping endDate="2008-01-02" effectiveDate="2008-06-24">
        <routeGroupingID>S62</routeGroupingID>
        <routeName> Local for Route S62/S92</routeName>
        <routeDescription> S62 Local Travis-St. George Ferry Terminal</routeDescription>
        <routeGroupingCode>curtain</routeGroupingCode>
        <groupingSetList>
                <tripSet>
                        <tripID>9201</tripID>
                        <routeID>6292</routeID>
                </tripSet>
        </groupingSetList>
        <groupingSetList>
                <patternSet>
                        <patternID>6203</patternID>
                        <routeID>6292</routeID>
                </patternSet>
        </groupingSetList>
</RouteGrouping>
```

The SDP document fragment may also be documented with either the tripSet or the patternSet, as follows:

```
<RouteGrouping endDate="2008-01-02" effectiveDate="2008-06-24">
        <routeGroupingID>S62</routeGroupingID>
        <routeName> Local for Route S62/S92</routeName>
        <routeDescription> S62 Local Travis-St. George Ferry Terminal</routeDescription>
        <routeGroupingCode>curtain</routeGroupingCode>
        <groupingSetList>
                <tripSet>
                        <tripID>9201</tripID>
                        <routeID>6292</routeID>
                </tripSet>
```

```
        </groupingSetList>
</RouteGrouping>
```

Or

```
<RouteGrouping endDate="2008-01-02" effectiveDate="2008-06-24">
        <routeGroupingID>S62</routeGroupingID>
        <routeName> Local for Route S62/S92</routeName>
        <routeDescription> S62 Local Travis-St. George Ferry Terminal</routeDescription>
        <routeGroupingCode>curtain</routeGroupingCode>
        <groupingSetList>
                <patternSet>
                        <patternID>6203</patternID>
                        <routeID>6292</routeID>
                </patternSet>
        </groupingSetList>
</RouteGrouping>
```