

## **PermanentNet Data Model**

Munich January 2012

## Network model

The network is build within Administrative areas to afford updating by responsible local authorities.

### Administrative areas

Polygon feature class							
<b>AdministrativeArea</b>							
Field name	Data type	Al-lows nulls	De-fault Value	Domain	Pre-cision	Scal-e	Length
OBJECTID	Object ID						
Shape	Geometry	Yes					
AdminAreaID	ShortInteger	No			0		
ISO	String	Yes		ISOCode			3
NameEnglish	String	Yes					80
NameLocal	String	Yes					80
AreaType	ShortInteger	Yes		AreaCode	0		
RecordDate	Date	Yes					
RecordStatus	Short Integer	Yes	1	RecordCodes	0		
EntityStatus	Short Integer	Yes	1	EntityCodes	0		
FromDate	Date	Yes					
ToDate	Date	Yes					
DataOrigin	Short Integer	Yes		DataOrig-inCodes	0		
GlobalID	GUID	No			0		
Shape_length	Double	Yes			0	0	
Shape_area	Double	Yes			0	0	

Identifier of the Administrative Area  
ISO code of the Administrative Area  
English name of the Administrative Area  
Local name of the Administrative Area  
One-digit area type code  
Date record was created  
Status of record  
Status of the link  
Starting date of validity  
Ending date of validity  
Origin of the data  
Global Unique Identifier

Coded value domain	
<b>AreaCode</b>	
Description	<i>AreaType</i>
Field type	<i>Short Integer</i>
Split policy	<i>Default value</i>
Merge policy	<i>Default value</i>
Code	Description
1	Country
2	Region
3	District
4	Commune

Coded value domain	
<b>ISOCode</b>	
Description	<i>AreaType</i>
Field type	<i>Short Integer</i>
Split policy	<i>Default value</i>
Merge policy	<i>Default value</i>
Code	Description
1	NUTS0
2	NUTS1
3	NUTS2
4	NUTS3

Transport areas

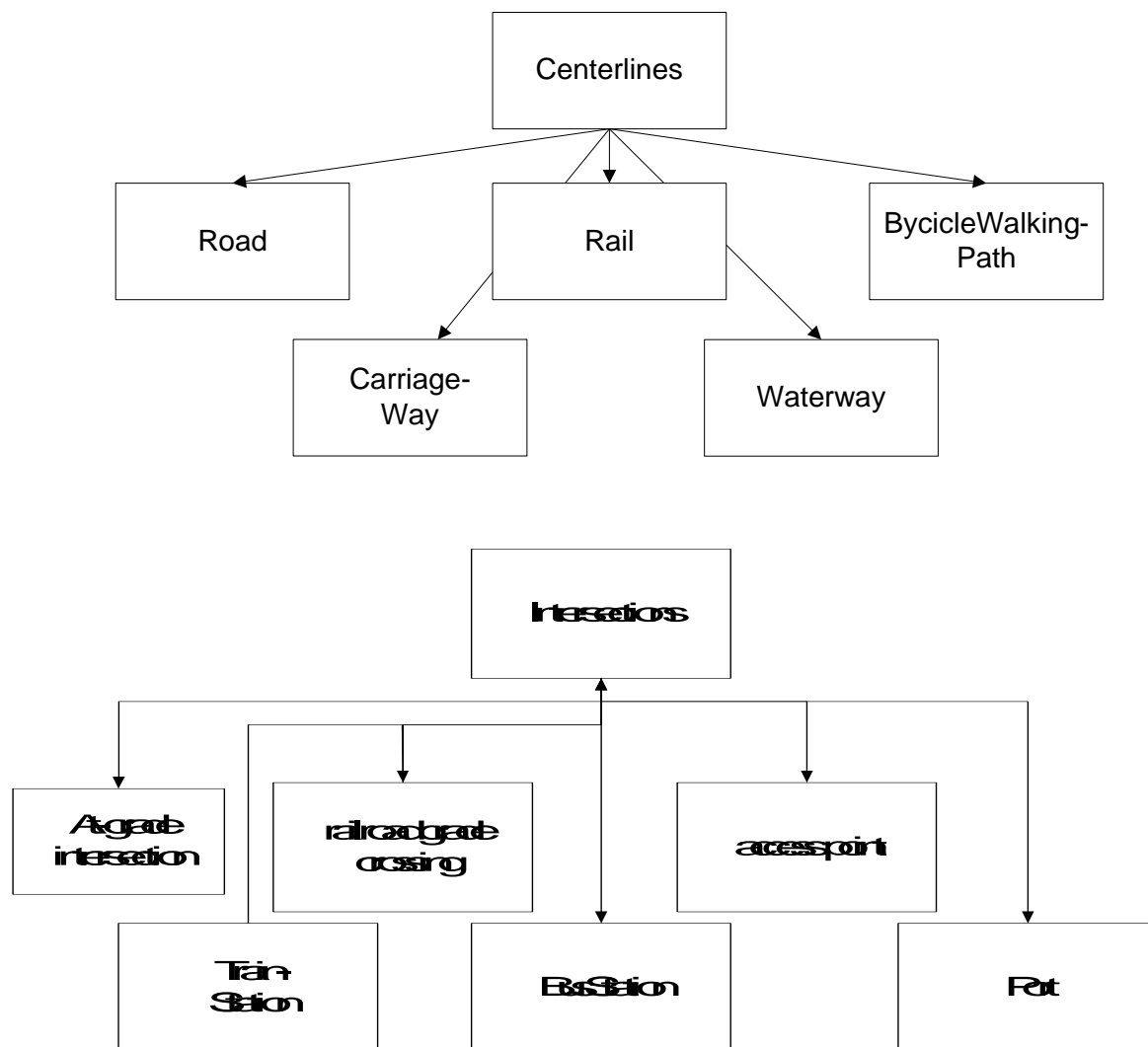
The Transport Network is part of Transport areas.

Polygon feature class							
<b>TransportArea</b>							
Field name	Data type	Al-lows nulls	De-fault Value	Domain	Preci-sion	Scal-e	Length
OBJECTID	Object ID						
Shape	Geometry	Yes					
Transport-AreaID	ShortInteger	No			0		Identifier of the Transport Area
Transport-AreaType	ShortInteger	Yes		Transport-AreaType	0		One-digit area type code
Transport-AreaNamID	ShortInteger	Yes			0		Identifier of the TransportArea Names
RecordDate	Date	Yes					Date record was created
RecordStatus	Short Integer	Yes	1	RecordCodes	0		Status of record
EntityStatus	Short Integer	Yes	1	EntityCodes	0		Status of the link
FromDate	Date	Yes					Starting date of validity
ToDate	Date	Yes					Ending date of validity
DataOrigin	Short Integer	Yes		DataOrig-inCodes	0		Origin of the data
GlobalID	GUID	No			0		Global Unique Identifier
Shape_length	Double	Yes			0	0	
Shape_area	Double	Yes			0	0	

Coded value domain	
<b>TransportAreaType</b>	
Description	<i>AreaType</i>
Field type	<i>Short Integer</i>
Split policy	<i>Default value</i>
Merge policy	<i>Default value</i>
Code	Description
1	RoadArea
2	RoadServiceArea
3	RailwayArea
4	RailwayStationArea
5	RailwayYardArea
6	WaterwayArea
7	PortArea

## Transportation Network

The Transportation Network consists of ComplexEdgeFeatures and SimpleJunctionFeatures. The ComplexEdgeFeatures correspond to one or more JunctionFeatures. They are specified as Centerlines. Subclasses of Centerlines are Road, Rail, WaterWay, BicyclePath and WalkingPath. The SimpleJunctionFeatures are specified to be Intersections. Subtypes are at-grade intersection where two or more routes meet, railroad grade crossing where a route cross a railroad, access point, which represents a location on a route where a nonmapped facility connects to mapped facility like shopping center driveway. Together these objects form the Geometric Network.



Like the definition in the traditional networks the Edgefeatures are called **Links** and the Junctionfeatures **Nodes**.

Centerlines

Simple feature class							
Link							
Field name	Data type	Al-lows nulls	De-fault Val-ue	Domain	Pre-cisio-n	Scal-e	Length
OBJECTID	Object ID						
Shape	Geometry	Yes					
TransportSub-supportSub-classes	Short Integer	No			0		
LinkID	String	No					80
FromNode	String	No					80
ToNode	String	No					80
AdministrativeAreaID	String	No					80
LinkNameID	Long Integer	Yes			0		
LinkCourse	Short Integer	Yes		LinkCourse	0		
LinkType	Short Integer	Yes	1	LinkTypes	0		
Brunnel	Short Integer	Yes	1	Brunnel	0		
LRMPositionID	Long Integer	Yes			0		
Direction	Short Integer	Yes	1	DirectionTypes	0		
RecordDate	Date	Yes					
RecordStatus	Short Integer	Yes	1	RecordCodes	0		
EntityStatus	Short Integer	Yes	1	EntityCodes	0		
FromDate	Date	Yes					
ToDate	Date	Yes					
DataOrigin	Short Integer	Yes		DataOriginCodes	0		
GlobalID	GUID	No			0		
Shape_length	Double						

Subclasses

Identifier of link

Starting node

Ending node

Identifier of the Administrative Area

Identifier of the traffic facility names

Type of course

Transport function

Identifier of LRM position

Direction

Date record was created

Status of record

Status of the link

Starting date of validity

Ending date of validity

Origin of the data

Global Unique Identifier

Each links has a permanent Identity **LinkID** build by the Identifier of the regional Administrative area and x-coordinate of the starting point (FromNode) and the y-coordinate of the ending point (ToNode).

The LinkNameID refers to a table with a collection of road names.

There are different kind of **LinkNames**, e.g EuroRoadNumber like “E45”, StateRoadName like “A 9” or “B 10”, and StreetName like “Westerndorfer Straße”. The LinkNameTable collects all these names of each link.

Table							
<b>LinkNames</b>							
Field name	Data type	Al-lows nulls	De-fault Val-ue	Domain	Pre-cisio-n	Scal-e	Length
OBJECTID	Object ID						
LinkNameID	GUID	Yes			0		
EuroRoad-Number	String	Yes					50
StateRoadName	String	Yes					50
StreetName	String	Yes					50
Name	String	Yes					50
RecordDate	Date	Yes					
RecordStatus	Short Integer	Yes	1	RecordCodes	0		
EntityStatus	Short Integer	Yes	1	EntityCodes	0		
FromDate	Date	Yes					
ToDate	Date	Yes					
DataOrigin	Short Integer	Yes		DataOriginCodes	0		

Identifier of the link name collection  
Name of the Europe road  
Administrative road name  
Street name  
Other link name  
Date record was created  
Status of record  
Status of the link  
Starting date of validity  
Ending date of validity  
Origin of the data

Relationship class		
<b>LinkHasNames</b>		
Type	<i>Simple</i>	
Cardinality	<i>One to one</i>	Forward label <i>Has</i>
Notification	<i>None</i>	Backward label <i>Represents</i>
Origin table		Destination table
Name	<i>Link</i>	
Primary key	<i>LinkNameID</i>	Name <i>LinkNames</i>
Foreign key	<i>LinkNameID</i>	

Coded value domain	
<b>LinkTypes</b>	
Description	<i>RoadType</i>
Field type	<i>Short Integer</i>
Split policy	<i>Default value</i>
Merge policy	<i>Default value</i>
Code	Description
1	Motorway
2	Trunk
3	Primary
4	Secondary
5	Tertiary
6	Local Street
7	Service Road
8	Track
9	Bridleway
10	Cycleway
11	Footway
12	Path
13	Floors
14	Stairs
15	Lifts

Coded value domain	
<b>LinkCourse</b>	
Description	<i>RoadType</i>
Field type	<i>Short Integer</i>
Split policy	<i>Default value</i>
Merge policy	<i>Default value</i>
Code	Description
1	Transport facility
2	Transport facility equal carriageway
3	Carriageway
4	Lane
5	Shoulder
6	Ramp
7	Roundabout
10	Main track
11	Multi- track

## PermanentNet Data Model

Seite 7

Coded value domain

**LinkTypes**

Description *RailwayType*  
 Field type *Short Integer*  
 Split policy *Default value*  
 Merge policy *Default value*

Code	Description
20	Rail
21	Commuter railway
22	Underground railway
23	Tram
30	Other

Coded value domain

**LinkTypes**

Description *WaterwayType*  
 Field type *Short Integer*  
 Split policy *Default value*  
 Merge policy *Default value*

Code	Description
30	Stream
31	River
32	Canal
33	Drain

Coded value domain

**Brunnel**

Description *RoadSubType*  
 Field type *Short Integer*  
 Split policy *Default value*  
 Merge policy *Default value*

Code	Description
1	Bridge
2	Tunnel

Intersections

The Nodes are build to serve information about road and rail intersections, administrative border crossing as well as identities and names of the connected links. This information can be used e.g. for TPEG messages.

Simple feature class							
<b>Node</b>							
Field name	Data type	Al-lows nulls	De-fault Val-ue	Domain	Pre-cisio-n	Scale	Length
OBJECTID	Object ID						
Shape	Geometry	Yes					
NodeID	String	No					80
NodeType	Short Integer	No	1	Node Code	0		
NodeName	String						80
LinkType	ShortInteger	No	1	LinkType	0		
AdministrativeAreaID	String	No					80
LinkNameID	Long Integer	Yes			0		
LinkNoID	Long Integer	Yes			0		
LRMPositionID	Long Integer	Yes			0		
RecordDate	Date	Yes					
RecordStatus	Short Integer	Yes	1	RecordCodes	0		
EntityStatus	Short Integer	Yes	1	EntityCodes	0		
FromDate	Date	Yes					
ToDate	Date	Yes					
DataOrigin	Short Integer	Yes		DataOriginCodes	0		
GlobalID	GUID	No			0		

Identifier of the Node  
Type of node  
Type of highest connected Link  
Identifier of the Administrative Area  
Identifier of the collection of names of connected streets or roads  
Identifier of the collection of linkIDs of connected streets or roads  
Identifier of LRM position  
Date record was created  
Status of record  
Status of the node  
Starting date of validity  
Ending date of validity  
Origin of the data  
Global Unique Identifier

Cded value domain	
<b>NodeType</b>	
Description	<i>RoadType</i>
Field type	<i>Short Integer</i>
Split policy	<i>Default value</i>
Merge policy	<i>Default value</i>
Code	Description
1	Intersection transport facility
2	Intersection transport facility equal carriageway
3	Intersection carriageway
4	Administrative line
5	Bridge/tunnel enter
6	Cul de sac
20	Other



## PermanentNet Data Model

Seite 9

Coded value domain	
<b>NodeType</b>	
Description	<i>RailwayType</i>
Field type	<i>Short Integer</i>
Split policy	<i>Default value</i>
Merge policy	<i>Default value</i>
Code	Description
10	Station
11	Tram stop
20	Other

The NodeID is a function of Latitude and Longitude

$\text{round}(\text{int}(@\text{Value}(\_x) * 1000000)) * 100000000 + \text{int}(@\text{Value}(\_y) * 1000000)$

Table							
<b>LinkToNodeID</b>							
Field name	Data type	Al- lows nulls	De- fault Val- ue	Domain	Pre- cisio n	Scal e	Length
OBJECTID	Object ID						
NodeID	String	Yes					80
LinkID	String	Yes					80

Identifier of the node  
Identifier of the link

Relationship class	
<b>NodeHasLinks</b>	
Type	<i>Simple</i>
Cardinality	<i>One to many</i>
Notification	<i>None</i>
	Forward label <i>Has</i> Backward label <i>Represents</i>
Origin table	Destination table
Name <i>Node</i> Primary key <i>NodeID</i> Foreign key <i>NodeID</i>	Name <i>LinkToNodeID</i>

Coded value domain

**EntityCodes**

Description                      *Entity status*  
Field type                         *Short Integer*  
Split policy                       *Default value*  
Merge policy                      *Default value*

Code	Description
1	Proposed
2	Under review
3	In design
4	Under construction
5	Substantial completion
6	Open to traffic
7	Accepted, in service
11	Damaged
12	Under repair
21	Jurisdiction transferred
31	Closed to traffic
32	Removed from service
33	Abandoned
34	Surplused
35	Stored
86	Demolished

Coded value domain

**RecordCodes**

Description                       *Record status*  
Field type                         *Short Integer*  
Split policy                       *Default value*  
Merge policy                      *Default value*

Code	Description
0	Work in progress
1	Proposed
2	Withdrawn
3	Rejected
4	Accepted
5	Active
6	Replaced
7	Retired

Coded value domain

**DirectionTypes**

Description                       *NodeType*  
Field type                         *Short Integer*  
Split policy                       *Default value*  
Merge policy                      *Default value*

Code	Description
0	To-From travel permitted
1	To travel permitted
2	From travel permitted

Coded value domain

**DataOriginCodes**

Description                       *DataOrigin*  
Field type                         *Short Integer*  
Split policy                       *Default value*  
Merge policy                      *Default value*

Code	Description
1	Open Street Map
2	Basis-DLM (AAA)
3	NavTeg
4	Tele Atlas
11	IPM geoNet

Coded value domain

**SideCode**

Description                       *DataOrigin*  
Field type                         *Short Integer*  
Split policy                       *Default value*  
Merge policy                      *Default value*

Code	Description
L	Left side only
R	Right side only
B	Both sides & street
S	Street only
N	Both sides & not street

Coded value domain

**Referents**

Description                       *DataOrigin*  
Field type                         *Short Integer*  
Split policy                       *Default value*  
Merge policy                      *Default value*

Code	Description
0	None
1	Edge of pavement
2	Face of curb
3	Back of curb
4	Back of sidewalk
5	Edge of right of way
6	Centerline of road
7	Centerline of ditch

Coded value domain	
<b>OffsetDir</b>	
Description	<i>DataOrigin</i>
Field type	<i>Short Integer</i>
Split policy	<i>Default value</i>
Merge policy	<i>Default value</i>
Code	Description
0	None
1	In
2	Out
3	North
4	East
5	South
6	West
7	Toward
8	Away from

Linear Referencing

A sequence of Links in one direction are building a directed Polyline called Route. This linear object together with an implementation of a linear referencing system (LRS) allows the use of different kinds of linear referencing methods (LRM) like dynamic segmentation.

The table LinkSequence represents a many-to-many relationship that can be used to identify all segments participating in a route and also when one segment is shared by multiple (overlapping) routes

Table							
LinkSequence							
Field name	Data type	Al-lows nulls	De-fault Val-ue	Domain	Pre-cisio n	Scale	Length
OBJECTID	Object ID						
LinkID	String	Yes					80
RouteID	String	Yes					80
TraversalID	String	Yes					80
RecordDate	Date	Yes					
RecordStatus	Short Integer	Yes	1	RecordCodes	0		
EntityStatus	Short Integer	Yes	1	EntityCodes	0		
FromDate	Date	Yes					
ToDate	Date	Yes					
DataOrigin	Short Integer	Yes		DataOriginCodes	0		

Identifier of the Link  
Identifier of the route  
Identifier of the traversal  
Date record was created  
Status of record  
Status of the link  
Starting date of validity  
Ending date of validity  
Origin of the data

Relationship class		
LinksHaveSequence		
Type	<i>Simple</i>	
Cardinality	<i>One to many</i>	Forward label <i>Sequence</i>
Notification	<i>None</i>	Backward label <i>Links</i>
Origin table		Destination table
Name	<i>Links</i>	
Primary key	<i>LinkID</i>	Name <i>LinksSequence</i>
Foreign key	<i>LinkID</i>	

The use of directed lines and polylines with linear measures is to find exactly the position of objects and events on and at main roads, railways and waterways like:

- Building sites
- Measuring points
- Traffic information (accident, weather)

Simple feature class							
<b>Route</b>							
Field name	Data type	Al-lows nulls	Default Value	Domain	Pre-cision	Scale	Length
OBJECTID	Object ID						
Shape	Geometry	Yes					
RouteID	String	No					80
RouteName	String	Yes					80
RouteType	Short Integer	No		RouteTypes	0		
RecordDate	Date	Yes					
RecordStatus	Short Integer	Yes	1	RecordCodes	0		
EntityStatus	Short Integer	Yes	1	EntityCodes	0		
FromDate	Date	Yes					
ToDate	Date	Yes					
DataOrigin	Short Integer	Yes		DataOrig-inCodes	0		
GlobalID	GUID	No			0		
Shape_length	Double						

Identifier of the route  
Route name  
Date record was created  
Status of record  
Status of the link  
Starting date of validity  
Ending date of validity  
Origin of the data  
Global Unique Identifier

Coded value domain	
<b>RouteTypes</b>	
Description	<i>DataOrigin</i>
Field type	<i>Short Integer</i>
Split policy	<i>Default value</i>
Merge policy	<i>Default value</i>
1	Road
2	Railroad
3	Navigable waterway
4	Transit pattern
5	

Relationship class		
<b>RoutesHaveLinkSequence</b>		
Type	<i>Simple</i>	
Cardinality	<i>One to many</i>	Forward label <i>Route</i>
Notification	<i>None</i>	Backward label <i>Sequence</i>
Origin table		Destination table
Name	<i>Route</i>	Name <i>LinkSequence</i>
Primary key	<i>RouteID</i>	
Foreign key	<i>RouteID</i>	

The **LRMpositionID** allows multiple datums to be accommodated, for both Linear Referencing Methods like miles from a selected StartNode and geographic positions in coordinates. An attributed relationship class handles the LRM position to geographic equivalencies, as this is a many to many relationship. So every Route can be build out of n links, and any link can be related to m Routes.

Table								
<b>LRMPosition</b>								
Field name	Data type	Al-lows nulls	Default Value	Domain	Pre-cision	Scale	Length	
OBJECTID	Object ID							
Shape	Geometry	Yes						
LRMPositionID	String	No					80	Identifier of LRM position
LRMType	Short Integer	Yes		LRMTypes	0			Type of LRM
RouteID	String	No					80	
FromMeasure	Double	Yes			7	3		Starting LRM measure
ToMeasure	Double	Yes			7	3		Ending LRM measure
SideOfRoad	String	Yes		SideCode			1	Side of road
OffsetReferent	Short Integer	Yes		Referents	0			Lateral offset referent
OffsetDirection	Short Integer	Yes		OffsetDir	0			Lateral offset direction
RecordDate	Date	Yes						Date record was created
RecordStatus	Short Integer	Yes	1	RecordCodes	0			Status of record
EntityStatus	Short Integer	Yes	1	EntityCodes	0			Status of the link
FromDate	Date	Yes						Starting date of validity
ToDate	Date	Yes						Ending date of validity
DataOrigin	Short Integer	Yes		DataOriginCodes	0			Origin of the data

The LinksHaveLRMPositions table is the relationship between the Links and their position on a route. All the position values are stored in the LRMPosition table, but because there is a many-to-many relationship between these two tables, the data is maintained in a bridge table.

Table								
<b>LinksHaveLRMPositions</b>								
Field name	Data type	Al-lows nulls	De-fault Value	Domain	Pre-cision	Scal-e	Length	
OBJECTID	Object ID							
LinkID	String	Yes					80	Identifier of the Link
LRMPositionID	GUID	Yes			0			Identifier of the LRM position

Relationship class	
<b>LinksToLinksHaveLRMPositions</b>	
Type	<i>Simple</i>
Cardinality	<i>One to many</i>
Notification	<i>None</i>
	Forward label <i>LinksHaveLRMPositons</i> Backward label <i>Links</i>
Origin table	Destination table
Name <i>Links</i> Primary key <i>LinkID</i> Foreign key <i>LinkID</i>	Name <i>LinksHaveLRMPositions</i>

Table							
<b>GeoPosition</b>							
Field name	Data type	Al-lows nulls	De-fault Val-ue	Domain	Pre-cisio-n	Scal-e	Length
OBJECTID	Object ID						
GeoPositionID	GUID	Yes			0		
Datum	Short Integer	Yes		Datums	0		
Latitude	Double	Yes			7	3	
Longitude	Double	Yes			7	3	
RecordDate	Date	Yes					
RecordStatus	Short Integer	Yes	1	RecordCodes	0		
EntityStatus	Short Integer	Yes	1	EntityCodes	0		
FromDate	Date	Yes					
ToDate	Date	Yes					
DataOrigin	Short Integer	Yes		DataOriginCodes	0		

Identifier of the coordinate  
Type of datum  
Latitude coordinate  
Longitude coordinate  
Date record was created  
Status of record  
Status of the link  
Starting date of validity  
Ending date of validity  
Origin of the data

Attributed Relationship class						
<b>LRMPositionHasGeoPosition</b>						
Type	<i>Simple</i>		Forward label <i>Has</i>			
Cardinality	<i>Many to many</i>		Backward label <i>Locates</i>			
Notification	<i>None</i>					
Origin table			Destination table			
Name <i>LRMPosition</i>			Name <i>GeoPosition</i>			
Primary key <i>LRMPositionID</i>						
Foreign key <i>LRMPositionID</i>						
Field name	Data type	Al-lows nulls	Precision	Scale	Length	
OBJECTID	Object ID					
LRMPositionID	GUID	Yes	0			
GeoPositionID	GUID	Yes	0			

Coded value domain	
<b>LRMTypes</b>	
Description	<i>LRMTypes</i>
Field type	<i>Short Integer</i>
Split policy	<i>Default value</i>
Merge policy	<i>Default value</i>
Code	Description
1	Route milelog
2	Note-offset
3	
4	Other

Coded value domain	
<b>Datums</b>	
Description	<i>Geographic datums</i>
Field type	<i>Short Integer</i>
Split policy	<i>Default value</i>
Merge policy	<i>Default value</i>
Code	Description
1	WGS84
2	
3	
4	

## Objects

All kind of point and line objects can be referenced to the PermanentNet like

- POI
- Devices (detectors)
- TMCLocations
- Other geometric networks

Simple feature class							
PointObjects							
Field name	Data type	Al-lows nulls	Default Value	Domain	Pre-cision	Scal-e	Length
OBJECTID	Object ID						
Shape	Geometry	Yes					
ObjectID	String	No					80
ObjectName	String	Yes					80
LRMPositionID	Long Integer	Yes			0		
RecordDate	Date	Yes					
RecordStatus	Short Integer	Yes	1	RecordCodes	0		
EntityStatus	Short Integer	Yes	1	EntityCodes	0		
FromDate	Date	Yes					
ToDate	Date	Yes					
DataOrigin	Short Integer	Yes		DataOriginCodes	0		
GlobalID	GUID	No			0		

Identifier of the object  
Name of object  
Identifier of LRM position  
Date record was created  
Status of record  
Status of the station  
Starting date of validity  
Ending date of validity  
Origin of the data  
Global Unique Identifier

They can be referenced to the route with an Event table.

Table							
Event							
Field name	Data type	Al-lows nulls	Default Value	Domain	Pre-cision	Scal-e	Length
OBJECTID	Object ID						
EventID	GUID	Yes			0		
LRMPositionID	Long Integer	Yes			0		
RecordDate	Date	Yes					
RecordStatus	Short Integer	Yes	1	RecordCodes	0		
EntityStatus	Short Integer	Yes	1	EntityCodes	0		
FromDate	Date	Yes					
ToDate	Date	Yes					
DataOrigin	Short Integer	Yes		DataOriginCodes	0		

Identifier of the event  
Identifier of LRM position  
Date record was created  
Status of record  
Status of the link  
Starting date of validity  
Ending date of validity  
Origin of the data

The same structure of Event tables can be used for point and line objects, because the LRMPosition represents the FromMeasure and ToMeasure value, but a point event has only a FromMeasure value in the LRMPosition table.