Microsoft Dynamics

Microsoft Dynamics AX 2012 R3

Transportation management engines

White paper

January 2015

The Transportation management system (TMS) module includes a number of extension points that let you implement custom algorithms to perform tasks that are related to the rating of transport and freight reconciliation.

www.microsoft.com/dynamics/ax
Send feedback.

Contents

Transportation management engines

The Transportation management system (TMS) module includes a number of extension points that let you implement custom algorithms to perform tasks that are related to the rating of transport and freight reconciliation. The implementations of the algorithms are called Transportation management engines, also known as TMS engines or engines. The engines are delivered as implementations of specific .NET interfaces and deployed on the Microsoft Dynamics AX Application Object Server (AOS) tier. Each Transportation management engine can be switched on and off, and it can also be tuned at runtime, based on Microsoft Dynamics AX data. Some of the most important objectives of these engines include:

- Calculation of transportation rate
- Calculation of travel distance from point to point
- Calculation of the time it takes to travel from point to point
- Zone identification of addresses
- Distribution of transportation charges for shipments across source document lines, also known as apportionment of charges

This white paper describes all of the engines that are available in Microsoft Dynamics AX 2012 R3. It is recommended to also read the following white paper:

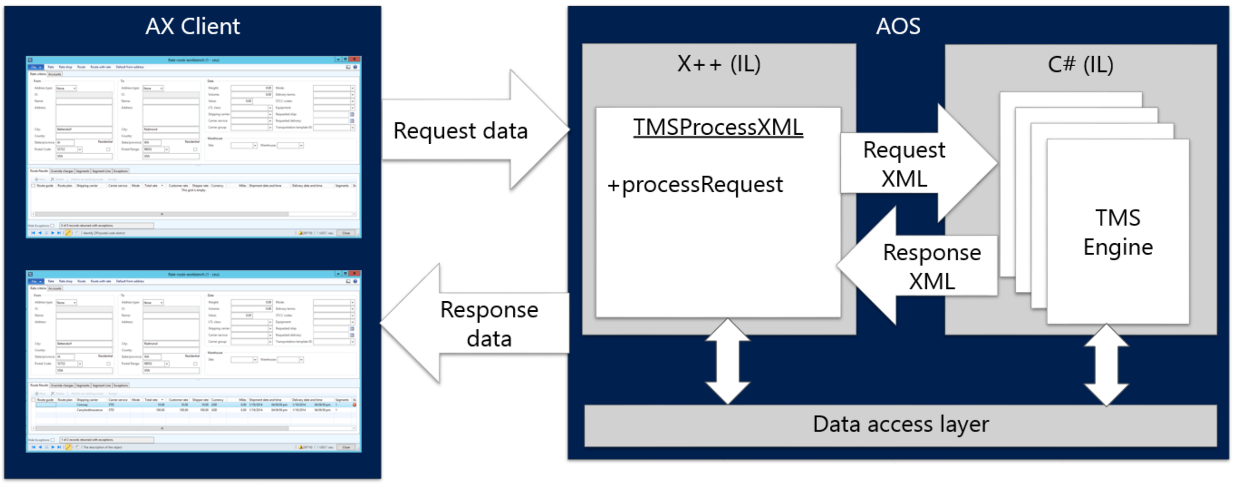
- Implementing and deploying transportation management engines

Audience

The audience of this document is implementation consultants and users who want to analyze the transportation situation at a company and perform the setup that is required in order to use the TMS in Microsoft Dynamics AX 2012 R3.

Architectural background

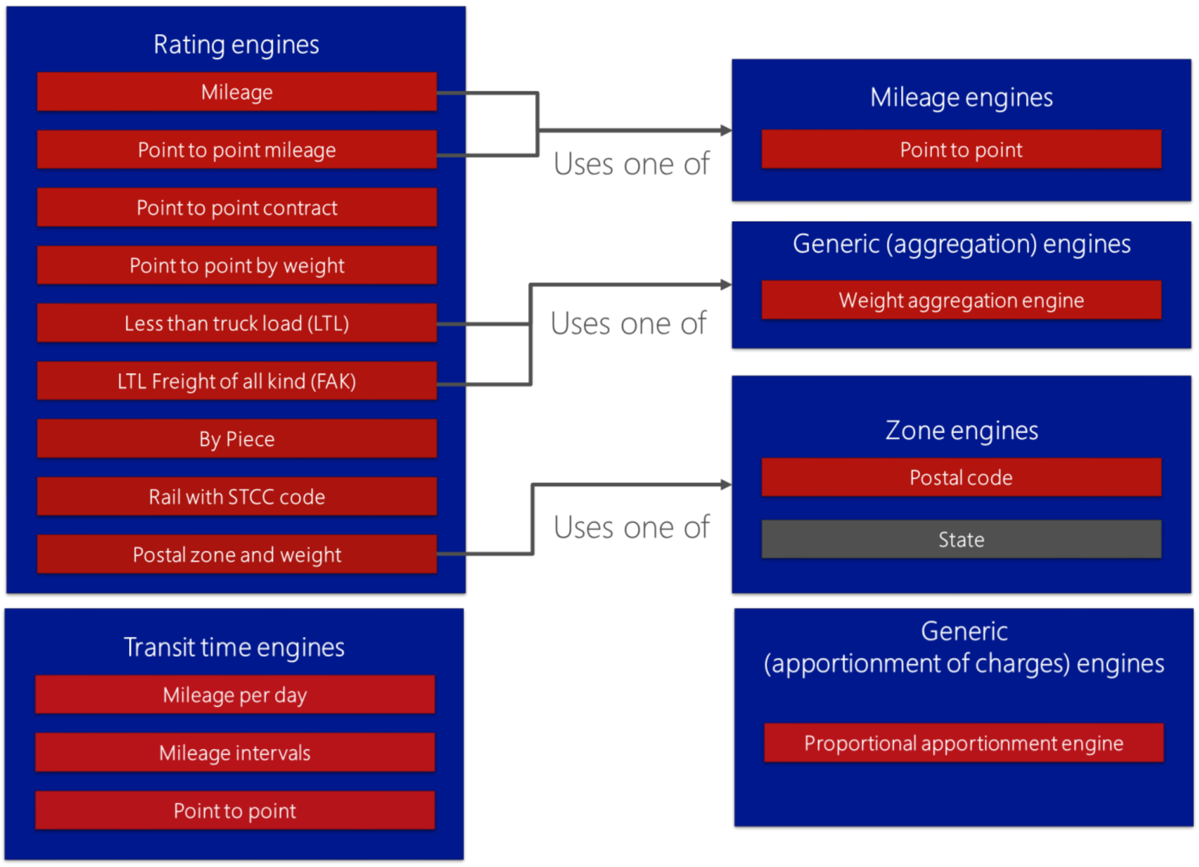
The following illustration shows a simplified view of the transportation management system.



In TMS, several operations require some kind of data processing that is specific to a particular carrier, such as transportation rate calculation. Typically, this kind of calculation requires a lot of input data, such as the origin and delivery addresses, the size, weight, and number of packages, and the requested delivery date. For a rate shopping operation, you can track this information from the **Rate route workbench** form. When you initiate a rate shopping request, request XML is constructed in TMS by using one of the X++ classes that are derived from TMSProcessXML_Base. The request XML is passed to the processing system that is encapsulated in the .NET assembly, named Microsoft.Dynamics.Ax.TMS. Further processing involves instantiation and utilization of one or more Transportation management engines. The final response from the TMS managed system consists of XML, which is interpreted into a result that is persisted in the Microsoft Dynamics AX database.

TMS Engines

TMS engines are divided into subcategories depending on their purpose. This section walks you through each subcategory and the engines that are provided in each of these subcategories. The following illustration shows a view of these categories and engines, and their relationship to each other.



All of the engines available in Microsoft Dynamics AX 2012 R3 rely only on data defined in the system. The engines are implemented in C#, designed for plug-ability and customization. If you are relying on Microsoft Dynamics AX data, the engines can reuse the generic engine data tables and no additional data model and UI is required. Engines that connect to external systems are not included with AX 2012 R3. However, the engine-based extensibility model lets you build extensions using Microsoft Dynamics AX Visual Studio Tools.

You need to initialize base engine data to create engine setup data for all the engines that are included with Microsoft Dynamics AX 2012 R3. If you initialize the base engine data in this manner, data is initialized per company and existing engine data that is set up in individual companies may be disregarded. The existing data will not be removed, but it might create a conflict depending on your setup.

To initialize base engine data automatically, follow these steps:

1. Click Transportation management > Setup > Transportation management parameters.
2. Under the General link, click the Engines FastTab.
3. Click the Initialize base engine data button.

This will initialize all the base engines and rate base types included with Microsoft Dynamics AX 2012 R3.

To rate a shipment using a specific carrier, you must configure multiple transportation management engines. The Rate engine is required, but other transportation management engines may be required to support the Rate engine. For example, the Rate engine can require the Mileage engine to provide mileage for calculating the rate based on mileage between the source and the destination.

The data that you use when you calculate rates in Microsoft Dynamics AX is controlled by the metadata configuration. For example, if you want to search for rates based on postal codes, you must set up metadata based on the lookup type of a postal code. Not all transportation management engines that are used to retrieve the data for rate calculation from external systems need metadata. The rate data for these engines can be retrieved from external transportation carrier systems, usually through a web service. For example, in AX 2012 R3 you can use a mileage engine that retrieves data directly from Bing maps so that you don't need metadata for this engine. To read more about how to create a mileage engine based on Bing maps, visit this blog post:
http://blogs.msdn.com/b/dynamicsaxscm/archive/2014/07/11/transportation-management-tms-mileage-engine-based-on-bing-maps.aspx

Metadata for transportation management engines is configured differently for the different types of engines. The following table explains the metadata configuration of each engine type.

Transportation management engine	Metadata configuration
Rate engine	Requires a **Rate base type**. The rate base type contains metadata for the rate base data and the rate base assignment data. The structure of rate base metadata is determined by the type of rate engine. The structure of the rate base assignment metadata is determined by the type of rate base assigner that is associated with that rate engine (through the rate engine parameters). You set up the rate base type of a rate engine on the **Rate engine** form and the **Rate master** form.
Zone engine	Requires metadata to be set up directly on the zone master.
Transit time engine and Mileage engine	Retrieves the metadata directly from the mileage or transit time engine's configuration setup form.

There are several rate base types included in Microsoft Dynamics AX 2012 R3 that will be created after initializing base engine data in the transportation management parameters. Each rate base type defines the structure of the user created data that is supplied to the rate engine through rate masters. The rate master includes one or more rate base assignments. Each rate base assignment refers to one or more rate bases. Rate engines, together with rate base assigners, consume that data in order to select the applicable transportation rate.

To view the rate base types, go to **Transportation management** > **Setup** > **General** > **Rate base type**. Note that making changes in the existing rate base types might cause the corresponding rate engines and rate base assigners to not work properly.

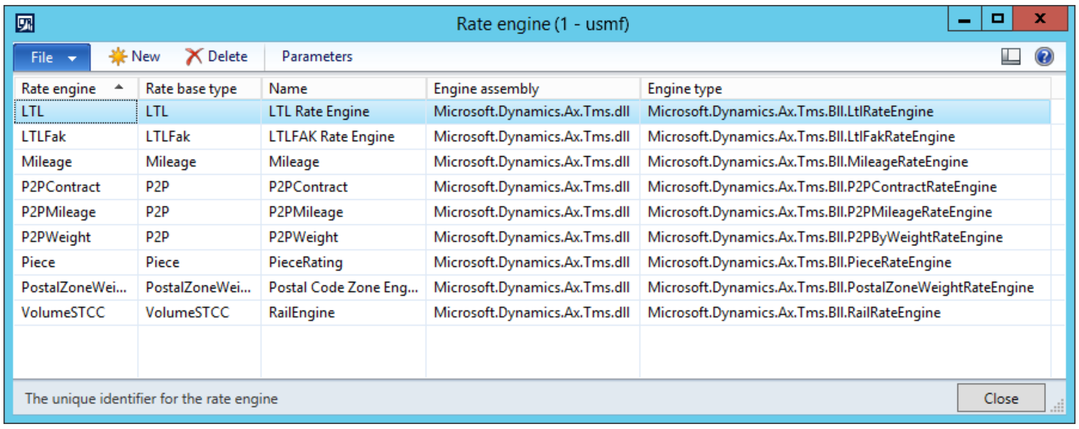
Rate engines

Rate engines calculate the rates based on different criteria. The criterion is defined in the setup of Rate base types, which control the metadata that you use to search for rates. The structure of the rate base type is defined by the type of rate engine and by the type of rate base assigner that is associated with the engine. The rate base type of a rate engine must correspond to the rate base type of the shipping carrier's rate master. To view the rate engines, go to Transportation management > Setup > Engines > Rate engine.

The fields in the Rate engine form are explained in the following table.

Field	Description
Rate engine	The rate engine identification field.
Rate base type	The rate base type, which defines the structure of data for the rate base assignment and the rate base.
Name	The user friendly name of the engine.
Engine assembly	The name of the .NET assembly (dll) containing the engine definition. For all the engines shipped with Microsoft Dynamics AX 2012 R3, the name of the assembly is Microsoft.Dynamics.Ax.Tms.dll.
Engine type	The fully-qualified name (including namespace) of the class that defines the engine within the engine assembly.

The base rate engine setup is shown in the following screenshot.



Rate engine	Rate base type	Name	Engine assembly	Engine type
LTL	LTL	LTL Rate Engine	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.LtlRateEngine
LTLFak	LTLFak	LTLFAK Rate Engine	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.LtlFakRateEngine
Mileage	Mileage	Mileage	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.MileageRateEngine
P2PContract	P2P	P2PContract	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.P2PContractRateEngine
P2PMileage	P2P	P2PMileage	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.P2PMileageRateEngine
P2PWeight	P2P	P2PWeight	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.P2PByWeightRateEngine
Piece	Piece	PieceRating	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.PieceRateEngine
PostalZoneWei...	PostalZoneWei...	Postal Code Zone Eng...	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.PostalZoneWeightRateEngine
VolumeSTCC	VolumeSTCC	RailEngine	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.RailRateEngine

The unique identifier for the rate engine

Click the Parameters button to view the parameters for any specific engine. The fields in the Parameters form are explained in the following table.

Field	Description
Parameter name	The name of the parameter expected at the engine instance initialization. The parameter names found in the various rate engines include: • RateBaseAssigner – This parameter specifies the type of the rate base assigner used by the rating engine for selection of the rate base assignment that is applicable for a specific rating request. Different rate base assigner types use different criteria for selecting rate base assignments. You can see what kind of criteria is associated with a particular rate base assigner by viewing the **Assignment** field type for the rate base type associated with the engine. The value of the RateBaseAssigner parameter is specified as: <assembly name with extension (.dll)>|<fully-qualified class name together with the namespace>. • MileageEngineCode – This parameter specifies the mileage engine that is associated with the rate engine. A rate engine using this parameter calculates rate based on mileage retrieved by the mileage engine specified in the parameter value. The value of the parameter corresponds to the **Mileage engine** key specified in **Mileage engine** form. • ApportionmentEngine – This parameter specifies the apportionment engine that will be used to apportion rates. The value of the parameter corresponds to the **Generic engine** key specified in the **Generic engine** form. • WeightApportionEngine – This parameter specifies the generic weight apportionment engine that is associated with the rate engine. The value of the parameter corresponds to the **Generic engine** key specified in the **Generic engine** form. • ZoneMasterCode – This parameter specifies the zone master that is associated with the rate engine. The value of the parameter corresponds to the **Zone master** key specified in the **Zone master** form. • PieceUnitOfMeasure – This parameter specifies what unit of measure will be used to calculate the effective number of freight pieces on which the transportation rate depends.
Parameter value	The value of the parameter.

Break masters are another essential component needed for rate engines to work properly. A break master is used to define the pricing structure and its breakpoints or intervals. The pricing structure uses tiered pricing that is based on physical dimensions. The break masters are then used by rate bases to define rates for each of the breakpoints. Each of the rate engines should have an appropriate break master associated with it. Some examples for break masters will be given for each rate engine. Use caution when changing break masters. We do not recommend that you change the fields for **Comparison** and **Data type** in any of the examples given. To set up break masters, go to **Transportation management** > **Setup** > **Rating** > **Break master**.

Mileage rate engine

The mileage rate engine calculates the rate depending on distance data and a fixed charge added to the rate. The engine name is **Mileage**.

The mileage rate engine calculation is as follows:

$$Total\ rate = mileage \times rate + addOnCharge$$

Where:

Mileage – Comes from the mileage engine assigned through the rating engine parameters. There is no unit conversion, which means that engine setup data needs to ensure that the mileage engine provides distance data, in the same unit, as specified in the **Rate base** form.

Rate – Transportation price per mileage unit. For example, this can be defined in order to have specific mileage intervals where different prices apply. The mileage intervals are defined by the associated break master. The price for each interval is then defined in the rate base.

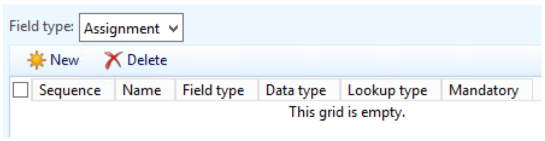
AddOnCharge – Fixed charge added for the rate.

The total rate can be adjusted if its value falls out of the range defined by the minimum/maximum charge specified on the **Rate base** form.

To view the setup for the engine, go to **Transportation management** > **Setup** > **Engines** > **Rate engine**. The setup for the mileage rate engine is shown in the following table.

Rate engine	Rate base type	Name	Engine assembly	Engine type
Mileage	Mileage	Mileage	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.MileageRateEngine

Note that the engine is associated with Rate base type named **Mileage**. The rate base type specifies metadata of data used by the rate engine, as well as metadata of data used by the rate base assigner that is associated with the rate engine. For the **Mileage** Rate base type, the following assignment metadata is specified.



Sequence	Name	Field type	Data type	Lookup type	Mandatory

This grid is empty.

In the **Rate engine** form, to view the parameters for the rate engine, click the **Parameters** button. The parameters for the mileage rate engine are shown in the following table.

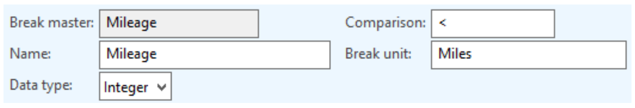
Parameter name	Parameter value
RateBaseAssigner	Microsoft.Dynamics.Ax.Tms.dll|Microsoft.Dynamics.Ax.Tms.Bll.MileageRateBaseAssigner
MileageEngineCode	P2P
ApportionmentEngine	AppWeight

The associated rate base assigner **MileageRateBaseAssigner** will find an applicable rate base assignment for specific rating requests. This requires data of specific format, which shown in the rate base type in the previous screenshot, does not include any specific fields.

The **MileageEngineCode** parameter value means that the rate engine will use the **P2P** mileage engine to retrieve mileage when calculating the rate. Note that the mileage engine is responsible for calculating the actual mileage for the transportation. The value of the parameter refers to the identification of the engine, as shown in the **Mileage engine** form.

There is a generic apportionment engine by weight, AppWeight, associated through the ApportionmentEngine parameter. The AppWeight engine apportions the freight charges based on weight. This is used when generating freight bills.

The break master associated with the mileage rate engine should define the mileage ranges for which different per mile rates apply. An example showing miles as a break unit is shown in the following screenshot.



Break master:	Mileage	Comparison:	<
Name:	Mileage	Break unit:	Miles
Data type:	Integer		

Point to Point Mileage rate engine

The point to point mileage rate engine calculates the rate based on the mileage between two points multiplied with the transportation price per mileage unit. The engine name is P2PMileage.

The point to point mileage rate engine calculation is as follows:

$$Total\ rate = mileage\ \times rate$$

Where:

Mileage – Comes from the mileage engine assigned through the rating engine parameters. There is no unit conversion, which means that engine setup data needs to ensure that the mileage engine provides distance data, in the same unit, as specified in the Rate base form.

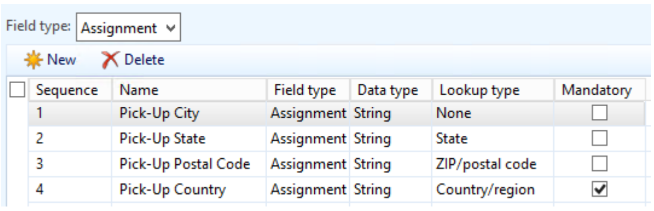
Rate – Transportation price per mileage unit. For example, this can be defined in order to have different prices per mile for different drop-off addresses (defined by city, state, and a range of postal codes).

The total rate can be adjusted if its value falls out of the range defined by the minimum/maximum charge specified on the Rate base form.

To view the setup for the engine, go to Transportation management > Setup > Engines > Rate engine. The setup for the point to point mileage rate engine is shown in the following table.

Rate engine	Rate base type	Name	Engine assembly	Engine type
P2PMileage	P2P	P2PMileage	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.P2PMileageRateEngine

Note that the engine is associated with Rate base type named P2P. The rate base type specifies metadata of data used by the rate engine, as well as metadata of data used by the rate base assigner associated with the rate engine. For the P2P rate base type, the following assignment metadata is specified.



Field type: Assignment

New Delete

Sequence	Name	Field type	Data type	Lookup type	Mandatory
1	Pick-Up City	Assignment	String	None	☐
2	Pick-Up State	Assignment	String	State	☐
3	Pick-Up Postal Code	Assignment	String	ZIP/postal code	☐
4	Pick-Up Country	Assignment	String	Country/region	☑

In the Rate engine form, to view the parameters for the rate engine, click the Parameters button. The parameters for the point to point mileage rate engine are shown in the following table.

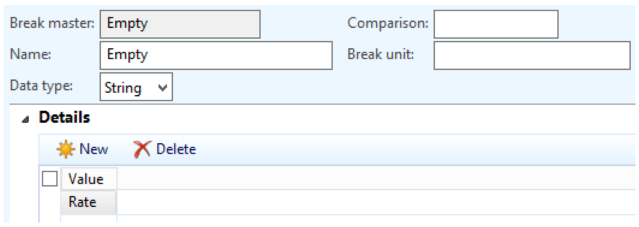
Parameter name	Parameter value
RateBaseAssigner	Microsoft.Dynamics.Ax.Tms.dll|Microsoft.Dynamics.Ax.Tms.Bll.P2PRateBaseAssigner
MileageEngineCode	P2P
ApportionmentEngine	AppWeight

The associated rate base assigner P2PRateBaseAssigner will find an applicable rate base assignment for specific rating requests. This requires data that is in a specific format, which shown in the rate base type in the previous screenshot, includes pick-up city, state, postal code, and country.

The parameter value of MileageEngineCode means that it will use the P2P mileage engine to retrieve mileage when calculating the rate. Note that it is the mileage engine that is responsible for calculating the actual mileage for the transportation. The value of the parameter refers to the identification of the engine, as shown in the Mileage engine form.

There is a generic apportionment engine by weight, AppWeight, associated through the ApportionmentEngine parameter. The AppWeight engine apportions the freight charges based on weight. This is used when generating freight bills.

The break master that is associated with the point to point mileage rate engine is not expected to have actual ranges, only one detail record for the rate is expected, as is shown in the following screenshot.



Break master:	Empty	Comparison:	
Name:	Empty	Break unit:	
Data type:	String		

Details

Value
Rate

Point to Point Contract rate engine

The point to point contract rate engine calculates the total rate based on a fixed rate that can be defined for different drop-off addresses. For example, this can be used if you have a carrier that offers fixed transportation prices between specified addresses. The engine name is P2PContract.

The point to point contract rate engine calculation is as follows:

$$Total\ rate = Rate$$

Where:

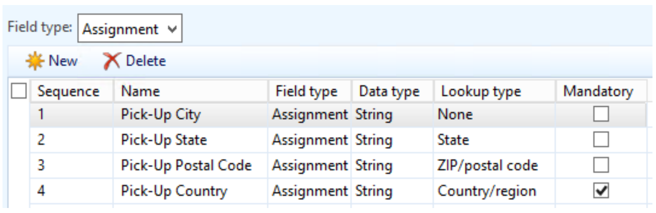
Rate – A fixed rate. For example, this can be defined to apply different fixed rates to different drop-off addresses (defined by city, state, and a range of postal codes).

The total rate can be adjusted if its value falls out of the range defined by the minimum/maximum charge specified on the Rate base form.

To view the setup for the engine, go to Transportation management > Setup > Engines > Rate engine. The setup for the point to point contract rate engine is shown in the following table.

Rate engine	Rate base type	Name	Engine assembly	Engine type
P2PContract	P2P	P2PContract	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.P2PContractRateEngine

Note that the engine is associated with Rate base type named P2P. The rate base type specifies metadata of data used by the rate engine, as well as metadata of data used by the rate base assigner associated with the rate engine. For the P2P Rate base type, the following assignment metadata is specified.



Field type: Assignment

Sequence	Name	Field type	Data type	Lookup type	Mandatory
1	Pick-Up City	Assignment	String	None	☐
2	Pick-Up State	Assignment	String	State	☐
3	Pick-Up Postal Code	Assignment	String	ZIP/postal code	☐
4	Pick-Up Country	Assignment	String	Country/region	✓

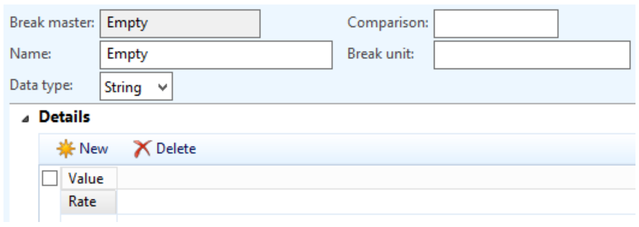
In the Rate engine form, to view the parameters for the rate engine, click the Parameters button. The parameters for the point to point rate engine are shown in the following table.

Parameter name	Parameter value
RateBaseAssigner	Microsoft.Dynamics.Ax.Tms.dll|Microsoft.Dynamics.Ax.Tms.Bll.P2PRateBaseAssigner
ApportionmentEngine	AppWeight

The associated rate base assigner P2PRateBaseAssigner will find an applicable rate base assignment for specific rating requests. This requires data that is in a specific format, which shown in the rate base type in the previous screenshot, includes pick-up city, state, postal code, and country.

It has the generic apportionment engine by weight, AppWeight, associated with it through the ApportionmentEngine parameter. The AppWeight engine apportions the freight charges based on weight. This is used when generating freight bills.

The break master associated with the point to point contract rate engine is not expected to have actual ranges, only one detail record for the rate is expected, as is shown in the following screenshot.



Break master: Empty
Name: Empty
Data type: String
Comparison:
Break unit:

Details

Value
Rate

Point to Point by Weight rate engine

The point to point by weight rate engine calculates the total rate based on the transportation price per unit of weight, times the total of the weight, divided by a fixed factor divisor. The engine name is P2PWeight.

The point to point by weight rate engine calculation is as follows:

$$Total\ rate = rate \times \frac{weight}{divisor}$$

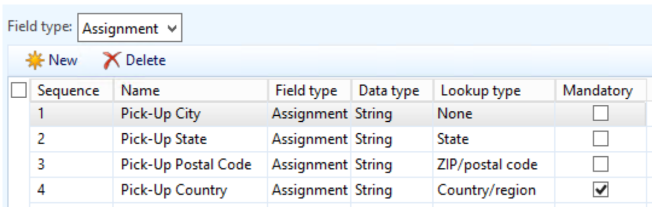
Where:
Rate – Transportation price per unit of weight. For example, this can be defined to apply different rates to different drop-off addresses (defined by city, state, and a range of postal codes). Moreover, different rates can be defined for different intervals of effective weight.
Weight – The actual weight in units as specified for the shipment in the Rate route workbench. There is no conversion support from the value specified in the Rate route workbench.
Divisor – A fixed factor specified in rate engine setup parameter Divisor. It is expected to be a non-zero decimal value.

The total rate can be adjusted if its value falls out of the range defined by the minimum/maximum charge specified on the Rate base form.

To view the setup for the engine, go to Transportation management > Setup > Engines > Rate engine. The setup for the mileage rate engine is shown in the following table.

Rate engine	Rate base type	Name	Engine assembly	Engine type
P2PWeight	P2P	P2PWeight	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.P2PByWeightRateEngine

Note that the engine is associated with Rate base type named **P2P**. The rate base type specifies metadata of data used by the rate engine, as well as metadata of data used by the rate base assigner associated with the rate engine. For the **P2P** Rate base type, the following assignment metadata is specified.



Field type: Assignment

New Delete

Sequence	Name	Field type	Data type	Lookup type	Mandatory
1	Pick-Up City	Assignment	String	None	☐
2	Pick-Up State	Assignment	String	State	☐
3	Pick-Up Postal Code	Assignment	String	ZIP/postal code	☐
4	Pick-Up Country	Assignment	String	Country/region	✓

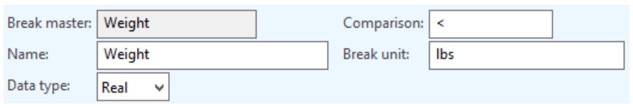
In the **Rate engine** form, to view the parameters for the rate engine, click the **Parameters** button. The parameters for the point to point by weight rate engine are shown in the following table.

Parameter name	Parameter value
ApportionmentEngine	AppWeight
Divisor	(intentionally left blank)

This engine has the generic apportionment engine by weight, **AppWeight**, associated with it through the **ApportionmentEngine** parameter. The **AppWeight** engine apportions the freight charges based on weight. This is used when generating freight bills.

The **Divisor** parameter is a fixed factor that can be used in accessorial charges for unit conversion. The parameter value for divisor is left blank, as this is specified as an accessorial fee type created in the accessorial assignment form by going to **Transportation management** > **Setup** > **Rating** > **Accessorial assignment**. If the divisor in the accessorial charge is not specified, it will be 1 and have no effect on the calculation of the rate.

The break master associated with the point to point by weight rate engine should define weight ranges, for which different rate per unit of weight will apply. An example of this is shown in the following screenshot. You can change weight unit depending on your needs.



Break master:	Weight	Comparison:	<
Name:	Weight	Break unit:	lbs
Data type:	Real		

Less than truckload (LTL) rate engine

The less than truckload (LTL) rate engine calculates the total rate by the sum of each LTL class rate recorded in the shipment. The rate calculation for an LTL class is calculated by multiplying the transportation price per unit of weight for that particular LTL class, with the total weight for that LTL class, divided by 100. The engine name is **LTL Rate Engine**.

The less than truckload rate engine calculation is as follows:

$$Total\ rate = \sum rate \times ceiling \frac{classWeight}{100}$$

Where:
classWeight – The total weight of the particular LTL class in the shipment. If the LTL class is selected in the **Rate route workbench**, the grouping of total shipment weight will be rated using that selected class.
Rate – Transportation price per unit of weight for a particular LTL class. For example, this can be defined to have specific weight intervals where different prices apply.

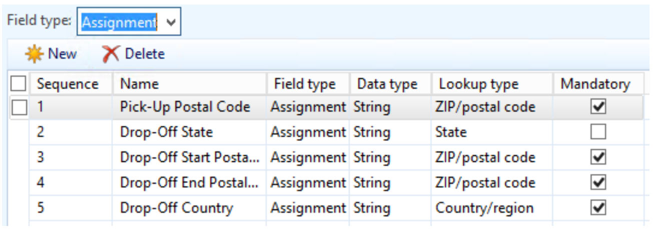
Sum – Rate calculation is executed separately for each of the LTL classes recorded in the shipment. After that, the total price is summed up.

The total rate can be adjusted if its value falls out of the range defined by the minimum/maximum charge specified on the **Rate base** form.

To view the setup for the engine, go to **Transportation management** > **Setup** > **Engines** > **Rate engine**. The setup for the LTL rate engine is shown in the following table.

Rate engine	Rate base type	Name	Engine assembly	Engine type
LTL	LTL	LTL Rate Engine	Microsoft.Dynamics .Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.LtlRateEngine

Note that the engine is associated with Rate base type named **LTL**. The rate base type specifies metadata of data used by the rate engine, as well as metadata of data used by the rate base assigner associated with the rate engine. For the **LTL** Rate base type, the following assignment metadata is specified.



Field type: Assignment

Sequence	Name	Field type	Data type	Lookup type	Mandatory
1	Pick-Up Postal Code	Assignment	String	ZIP/postal code	✓
2	Drop-Off State	Assignment	String	State	☐
3	Drop-Off Start Posta...	Assignment	String	ZIP/postal code	✓
4	Drop-Off End Postal...	Assignment	String	ZIP/postal code	✓
5	Drop-Off Country	Assignment	String	Country/region	✓

In the **Rate engine** form, to view the parameters for the rate engine, click the **Parameters** button. The parameters for the LTL rate engine are shown in the following table.

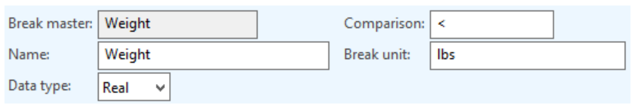
Parameter name	Parameter value
RateBaseAssigner	Microsoft.Dynamics.Ax.Tms.dll|Microsoft.Dynamics.Ax.Tms.Bll.LtlRateBaseAssigner
ApportionmentEngine	AppWeight
WeightApportionEngine	AppShipWeight

The associated rate base assigner **LTLRateBaseAssigner** will find applicable rate base assignment for specific rating requests. This requires data of a specific format, which shown in the rate base type in the previous screenshot, includes pick-up postal code and drop-off state, start postal code, end postal code, and country.

This engine has the generic apportionment engine by weight, **AppWeight**, associated with it through the **ApportionmentEngine** parameter. The **AppWeight** engine apportions the freight based on weight. This is used when generating freight bills.

The **WeightApportionEngine** parameter determines the generic weight apportionment engine that the **AppShipWeight** associates with the rate engine. This generic weight apportionment engine should only be used by the LTL and LTLFAK rate engine.

The break master associated with the less than truckload (LTL) rate engine should define weight ranges, for which different rates per unit of weight will apply. An example of this is shown in the following screenshot. You can change weight unit depending on your needs.



Break master:	Weight	Comparison:	<
Name:	Weight	Break unit:	lbs
Data type:	Real		

Less than truckload Freight of all kind rate engine

The less than truckload freight of all kind (LTLFak) rate engine calculates the total rate by the sum of each LTL class rate recorded in the shipment. The rate calculation for an LTL class is calculated by multiplying the transportation price per unit of weight for that particular LTL class, with the total weight for that LTL class, divided by 100, then multiplied by a discount value. The engine name is **LTLFAK Rate Engine**.

The less than truckload freight of all kind rate engine calculation is as follows:

$$Total\ rate = \sum rate \times ceiling \frac{classWeight}{100} \times (1 - discount)$$

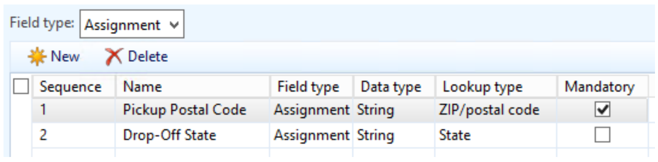
Where:
classWeight – The total weight of a particular LTL class in the shipment. If the LTL class is selected in the **Rate route workbench**, the grouping of total shipment weight will be rated using the selected class.
Rate – Transportation price per unit of weight for a particular LTL class. For example, this can be defined to have specific weight intervals where different prices apply. Different pricing can also be assigned depending on drop-off postal code ranges.
Discount – Discount value expressed as a real value between 0 and 1. This value is recorded as the 3rd dimension on the effective rate base assignment.
Sum – Rate calculation is executed separately for each of the LTL classes recorded in the shipment. After that, the total price is summed up.

The total rate can be adjusted if its value falls out of the range defined by the minimum/maximum charge that is specified on the **Rate base** form.

To view the setup for the engine, go to **Transportation management** > **Setup** > **Engines** > **Rate engine**. The setup for the LTL freight of all kind rate engine is shown in the following table.

Rate engine	Rate base type	Name	Engine assembly	Engine type
LTLFak	LTLFak	LTLFAK Rate Engine	Microsoft.Dynamics .Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.LtlFakRateEngine

Note that the engine is associated with Rate base type named **LTLFak**. The rate base type specifies metadata of data used by the rate engine, as well as metadata of data used by the rate base assigner associated with the rate engine. For the **LTLFak** Rate base type, the following assignment metadata is specified.



Field type: Assignment

Sequence	Name	Field type	Data type	Lookup type	Mandatory
1	Pickup Postal Code	Assignment	String	ZIP/postal code	✓
2	Drop-Off State	Assignment	String	State	☐

In the **Rate engine** form, to view the parameters for the rate engine, click the **Parameters** button. The parameters for the LTL freight of all kind rate engine are shown in the following table.

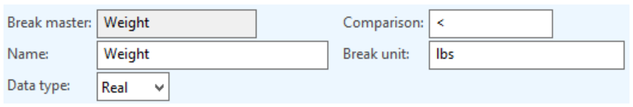
Parameter name	Parameter value
RateBaseAssigner	Microsoft.Dynamics.Ax.Tms.dll|Microsoft.Dynamics.Ax.Tms.Bll.LtlFakRateBaseAssigner
ApportionmentEngine	AppWeight
WeightApportionEngine	AppShipWeight

The associated rate base assigner **LTLFakRateBaseAssigner** will find an applicable rate base assignment for specific rating requests. This requires data of a specific format, which shown in the rate base type in the previous screenshot, includes pick-up postal code and drop-off state.

This engine has the generic apportionment engine by weight, AppWeight, associated with it through the ApportionmentEngine parameter. The AppWeight engine apportions the freight charges based on weight. This is used when generating freight bills.

The WeightApportionEngine parameter determines the generic weight apportionment engine AppShipWeight that is associated with the rate engine. This generic weight apportionment engine should only be used by the LTL and LTLFAK rate engine.

The break master associated with the less than truckload freight of all kind (LTLFak) rate engine should define weight ranges, for which different rate per unit of weight will apply. An example of this is shown in the following screenshot. You can change weight unit depending on your needs.



Break master:	Weight	Comparison:	<
Name:	Weight	Break unit:	lbs
Data type:	Real		

Piece rate engine

The piece rate engine calculates total rate by multiplying the transportation price, per piece of freight, with the number of freight pieces in the shipment. The engine name is PieceRating.

The piece rate engine calculation is as follows:

$$Total\ rate = Rate \times Pieces$$

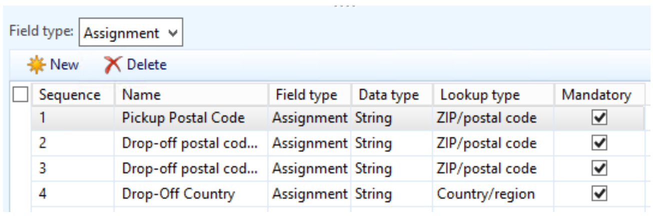
Where:
Rate – Transportation price per piece of freight. For example, this can be defined to have specific intervals for the number of pieces, where different prices apply.
Pieces – The number of freight pieces.

The total rate can be adjusted if its value falls out of the range defined by the minimum/maximum charge that is specified on the Rate base form.

To view the setup for the engine, go to Transportation management > Setup > Engines > Rate engine. The setup for the piece rate engine is shown in the following table.

Rate engine	Rate base type	Name	Engine assembly	Engine type
Piece	Piece	PieceRating	Microsoft.Dynamics .Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.PieceRateEngine

Note that the engine is associated with Rate base type named Piece. The rate base type specifies metadata of data used by the rate engine, as well as metadata of data used by the rate base assigner associated with the rate engine. For the Piece Rate base type, the following assignment metadata is specified.



Field type: Assignment

Sequence	Name	Field type	Data type	Lookup type	Mandatory
1	Pickup Postal Code	Assignment	String	ZIP/postal code	✓
2	Drop-off postal cod...	Assignment	String	ZIP/postal code	✓
3	Drop-off postal cod...	Assignment	String	ZIP/postal code	✓
4	Drop-Off Country	Assignment	String	Country/region	✓

In the Rate engine form, to view the parameters for the rate engine, click the Parameters button. The parameters for the piece rate engine are shown in the following table.

Parameter name	Parameter value
RateBaseAssigner	Microsoft.Dynamics.Ax.Tms.dll|Microsoft.Dynamics.Ax.Tms.Bll.PieceRateBaseAssigner

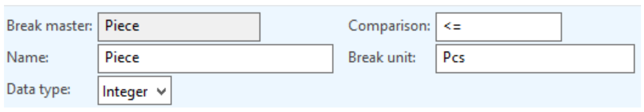
Parameter name	Parameter value
ApportionmentEngine	AppWeight
PieceUnitOfMeasure	Qty

The associated rate base assigner PieceRateBaseAssigner will find applicable rate base assignment for specific rating requests. This requires data of a specific format, which shown in the rate base type in the previous screenshot, includes pick-up postal code, drop-off postal code from, drop-off postal code to, and drop-off country.

This engine has the generic apportionment engine by weight, AppWeight, associated with it through the ApportionmentEngine parameter. The AppWeight engine apportions the freight charges on route, segments, shipment, container, or line based on weight. This used when generating freight bills.

The parameter PieceUnitOfMeasure specifies that quantity will be used as the unit of measure for determining the number of freight pieces.

The break master associated with the piece rate engine should define ranges of number of freight pieces, for which different rates per freight piece apply. An example of this setup, with pieces as break unit, is shown in the following screenshot.



Break master:	Piece	Comparison:	<=
Name:	Piece	Break unit:	Pcs
Data type:	Integer		

Postal Zone Weight rate engine

The postal zone weight rate engine calculates the total rate by the fixed price per container, multiplied with the number of zones from origin to destination. If there are several containers in the shipment, the total rate equals the sum of all container specific rates. The engine name is Postal Code Zone Engine.

The postal zone weight rate engine calculation is as follows:

$$Total\ rate = \sum ContainerRate$$

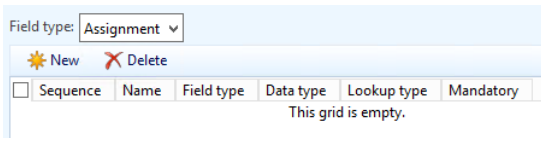
Where:
ContainerRate – Fixed price determined by the number of zones it takes to deliver goods from origin to destination. If the shipment is already separated into containers (after packing), the price is calculated per container. For example, the rate can be defined for applying different prices in specific weight intervals.
Sum – If the shipment is separated into containers, the total rate is equal to the sum of all container specific rates.

The total rate can be adjusted if its value falls out of the range defined by the minimum/maximum charge that is specified on the Rate base form.

To view the setup for the engine, go to Transportation management > Setup > Engines > Rate engine. The setup for the postal zone weight rate engine is shown in the following table.

Rate engine	Rate base type	Name	Engine assembly	Engine type
PostalZoneWeight	PostalZoneWeight	Postal Code Zone Engine	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.PostalZoneWeightRateEngine

Note that the engine is associated with Rate base type named PostalZoneWeight. The rate base type specifies metadata of data used by the rate engine, as well as metadata of data used by the rate base assigner associated with the rate engine. For the PostalZoneWeight Rate base type, the following assignment metadata is specified.


Field type: Assignment

New | Delete

Sequence	Name	Field type	Data type	Lookup type	Mandatory

This grid is empty.

In the **Rate engine** form, to view the parameters for the rate engine, click the **Parameters** button. The parameters for the postal zone weight rate engine are shown in the following table.

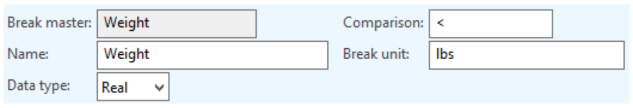
Parameter name	Parameter value
RateBaseAssigner	Microsoft.Dynamics.Ax.Tms.dll|Microsoft.Dynamics.Ax.Tms.Bll.PostalZoneWeightRateBaseAssigner
ApportionmentEngine	AppWeight
ZoneMasterCode	Postal

The associated rate base assigner **PostalZoneWeightRateBaseAssigner** will find the applicable rate base assignment for specific rating requests. It will require data of specific format, which shown in the rate base type in the previous screenshot, does not include any specific fields.

This engine has the generic apportionment engine by weight, **AppWeight**, associated with it through the **ApportionmentEngine** parameter. The **AppWeight** engine apportions the freight charges based on weight. This is used when generating freight bills.

The parameters also specify that the engine has **ZoneMasterCode** set with **Postal** as a parameter value, which means that the engine will need to utilize the postal zone engine in order to find the number of zones for the calculation of the rate. It's important to set up the postal zone master for all destinations that are to be used with this engine.

The break master associated with the postal zone weight rate engine should define weight ranges, for which different rate per unit of weight will apply. An example of this is shown in the following screenshot. You can change weight unit depending on your needs.


Break master:	Weight	Comparison:	<
Name:	Weight	Break unit:	lbs
Data type:	Real		

Rail rate engine

The rail rate engine calculates the total rate based on a fixed rate that is specified for a specific standard transportation commodity code (STCC). The STCC is a unique seven-digit code that is used to classify a commodity or group of commodity. The engine name is **RailEngine**.

The rail rate engine calculation is as follows:

$$Total\ rate = Rate$$

Where:

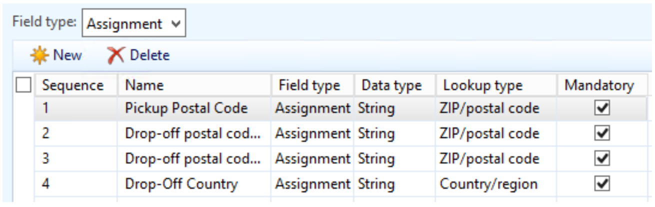
Rate – Fixed rate specific to a particular STCC code. If the shipment contains a number of different freight pieces, the effective rate is determined by the maximum textual value of the STCC codes specific to these items. Alternatively, the STCC code can be specified by the user in the **Rate route workbench**. For example, the rate can be defined so that different prices apply for specific volume intervals.

The total rate can be adjusted if its value falls out of the range defined by the minimum/maximum charge that is specified on the **Rate base** form.

To view the setup for the engine, go to **Transportation management** > **Setup** > **Engines** > **Rate engine**. The setup for the rail rate engine is shown in the following table.

Rate engine	Rate base type	Name	Engine assembly	Engine type
VolumeSTCC	VolumeSTCC	RailEngine	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.RailRateEngine

Note that the engine is associated with Rate base type named **VolumeSTCC**. The rate base type specifies metadata of data used by the rate engine, as well as metadata of data used by the rate base assigner associated with the rate engine. For the **VolumeSTCC** Rate base type, the following assignment metadata is specified.



Field type: Assignment

Sequence	Name	Field type	Data type	Lookup type	Mandatory
1	Pickup Postal Code	Assignment	String	ZIP/postal code	✓
2	Drop-off postal cod...	Assignment	String	ZIP/postal code	✓
3	Drop-off postal cod...	Assignment	String	ZIP/postal code	✓
4	Drop-Off Country	Assignment	String	Country/region	✓

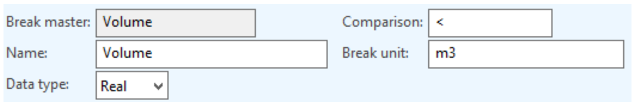
In the **Rate engine** form, to view the parameters for the rate engine, click the **Parameters** button. The parameters for the rail rate engine are shown in the following table.

Parameter name	Parameter value
RateBaseAssigner	Microsoft.Dynamics.Ax.Tms.dll|Microsoft.Dynamics.Ax.Tms.Bll.RailRateBaseAssigner
ApportionmentEngine	AppVolume

The associated rate base assigner **RailRateBaseAssigner** will find the applicable rate base assignment for specific rating requests. It will require data of specific format, which shown in the rate base type in the previous screenshot, includes pick-up postal code, drop-off postal code from, drop-off postal code to, and drop-off country.

It has the generic apportionment engine by weight, **AppVolume**, associated with it through the **ApportionmentEngine** parameter. The **AppVolume** engine apportions the freight charges based on volume. This is used when generating freight bills.

The break master associated with the rail rate engine should define volume ranges for which different rates will apply. An example of this setup, with m^3 as break unit, is shown in the following screenshot.



Break master:	Volume	Comparison:	<
Name:	Volume	Break unit:	m3
Data type:	Real		

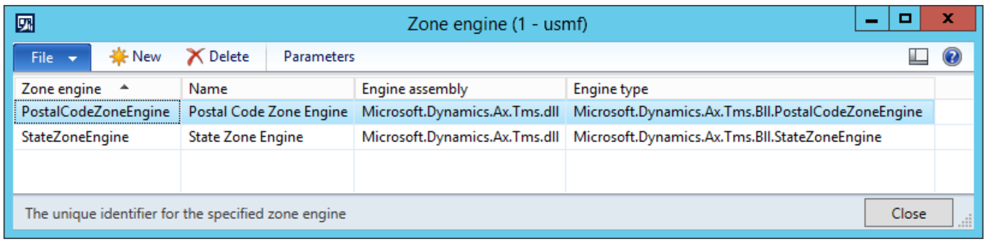
Zone engines

Zone engines calculate the zone based on the current address and calculate the number of zones that must be crossed in order to travel from address A to address B. The zone engine form can be viewed in Transportation management > Setup > Engines > Zone engine. In order to use a zone engine, you need to configure and define metadata for the zone master. The zone master can be viewed in Transportation management > Setup > General > Zone master.

The fields in the Zone engine form are explained in the following table.

Field	Description
Zone engine	The zone engine identification field.
Name	The user friendly name of the engine.
Engine assembly	The name of the .NET assembly (dll) containing the engine definition. For all the engines shipped with Microsoft Dynamics AX 2012 R3, the name of the assembly is Microsoft.Dynamics.Ax.Tms.dll.
Engine type	The fully-qualified name (including namespace) of the class that defines the engine within the engine assembly.

The base zone engine setup is shown in the following screenshot.



Zone engine	Name	Engine assembly	Engine type
PostalCodeZoneEngine	Postal Code Zone Engine	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.PostalCodeZoneEngine
StateZoneEngine	State Zone Engine	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.StateZoneEngine

The unique identifier for the specified zone engine

You can click on the Parameters button to view the parameters for a specific engine. The fields in the Parameters form are explained in the following table. None of the zone engines require any parameters.

Field	Description
Parameter name	The name of the parameter expected at the engine instance initialization.
Parameter value	The value of the parameter.

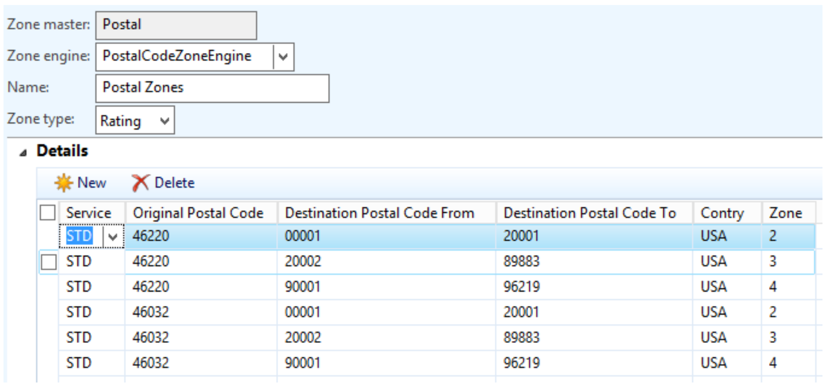
Postal Code Zone engine

The postal code zone engine identifies the number of zones from a source to the end destination of transportation. For each origin postal code, you can set up a range of destination postal codes that will be within specific zones.

To view the metadata set up for this zone engine, go to Transportation management > Setup > General > Zone master.

The Zone type for this zone engine is set to Rating, because this zone engine is used to interpret zones for rating scenarios. An example would be to calculate how many zones it takes to get from point A to B.

An example of how the setup for the zone master for the postal code zone engine is shown in the following screenshot. If you initialize base engine data in a new company, the details section will be empty and will need to be populated before using the engine.



Zone master:	Postal
Zone engine:	PostalCodeZoneEngine
Name:	Postal Zones
Zone type:	Rating

Details

Service	Original Postal Code	Destination Postal Code From	Destination Postal Code To	Contry	Zone
STD	46220	00001	20001	USA	2
STD	46220	20002	89883	USA	3
STD	46220	90001	96219	USA	4
STD	46032	00001	20001	USA	2
STD	46032	20002	89883	USA	3
STD	46032	90001	96219	USA	4

To view the setup for the engine, go to Transportation management > Setup > Engines > Zone engine. The setup for the postal code zone engine is shown in the following table.

Zone master	Zone engine	Name	Engine assembly	Engine type
Postal	PostalCodeZoneEngine	Postal Code Zone Engine	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.PostalCodeZoneEngine

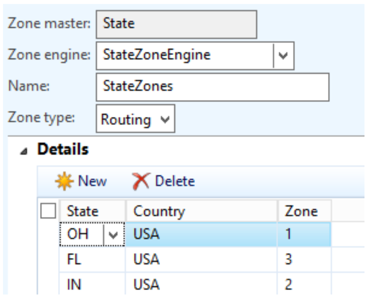
State Zone engine

The state zone engine identifies the zone that a particular address belongs to, depending on the state. It can be used to divide states into different user-defined zones that can be used when defining routing guides.

To view the metadata set up for this zone engine, go to Transportation management > Setup > General > Zone master.

The Zone type for this zone engine is set to Routing, because the zone engine is used for determining zones in routing scenarios. An example would be determining hub accessorial charges and selecting applicable route guides that use zones for applicability criteria.

An example of the zone master setup for this engine is shown in the following screenshot. If you initialize base engine data in a new company, the details section will be empty and will need to be populated before using the engine.



Zone master:	State
Zone engine:	StateZoneEngine
Name:	StateZones
Zone type:	Routing

Details

State	Country	Zone
OH	USA	1
FL	USA	3
IN	USA	2

To view the setup for the engine, go to Transportation management > Setup > Engines > Zone engine. The setup for the state zone engine is shown in the following table.

Zone master	Zone engine	Name	Engine assembly	Engine type
State	StateZoneEngine	State Zone Engine	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.StateZoneEngine

Transit time engines

Transit time engines calculate the time, in days, that is required to travel from the start to the end destination. Transit time engines are set up using the Transit time engine form in Transportation management > Setup > Engines > Transit time engine. In the Transit time engine form, on the Details FastTab, you can associate different carrier services with the specific parameters of the engine. This means that shipping carriers that are configured with several services can also use different transit time calculations depending on the service.

The fields in the Transit time engine form are explained in the following table.

Field	Description
Transit time engine	The transit time engine identification field.
Name	The user friendly name of the engine.
Engine assembly	The name of the .NET assembly (dll) containing the engine definition. For all the engines shipped with Microsoft Dynamics AX 2012 R3, the name of the assembly is Microsoft.Dynamics.Ax.Tms.dll.
Engine type	The fully-qualified name (including namespace) of the class that defines the engine within the engine assembly.

The transit time engine is associated with a shipping carrier on the Rating profile form, which can be viewed in Transportation management > Setup > Rating > Rating profile.

Calculation Transit Time engine

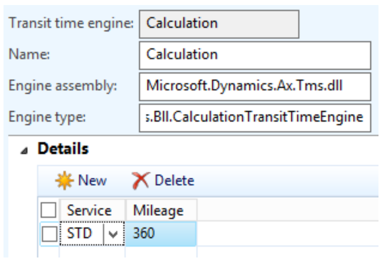
The calculation transit time engine calculates the transit time, in days, by dividing the mileage of shipment with the number of miles that the transportation carrier can handle per day.

The calculation transit time engine calculation is as follows:

$$Transit\ time\ days = \frac{Miles}{Mileage}$$

Where:
Miles - The actual mileage specific to a particular transportation.
Mileage – The number of miles that the transportation carrier can handle per day. There is no unit conversion, so you must use the same unit of length by mileage and miles.

This engine calculates transit time in days, based on the data that is put in for the shipment in the Rate route workbench. As shown in the example in the screenshot, a carrier service called STD is defined to handle 360 miles per day. This means that the shipping carrier with the associated service STD will use this transit time calculation. The shipping carrier will also need to be associated with the transit time engine in the rating profile. If you initialize base engine data in a new company, the details section will be empty and will need to be populated before using the engine.



Transit time engine:	Calculation
Name:	Calculation
Engine assembly:	Microsoft.Dynamics.Ax.Tms.dll
Engine type:	;.BII.CalculationTransitTimeEngine

Details

Service	Mileage
STD	360

Using the setup in the previous screenshot, if you have a shipment with 450 miles, transit time calculation will be as follows:

$$\frac{450}{360} = 1.25\ days$$

To view the setup for the engine, go to Transportation management > Setup > Engines > Transit time engine. The setup for the calculation transit time engine is shown in the following table.

Transit time engine	Name	Engine assembly	Engine type
Calculation	Calculation	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.CalculationTransitTimeEngine

Mile Breaks Transit Time engine

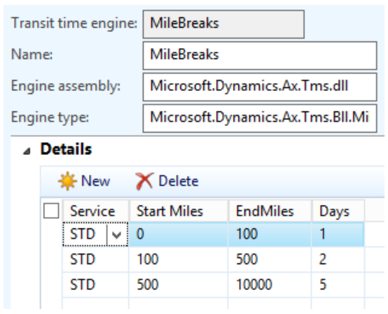
The mile breaks transit time engine calculates the transit time in day by a predefined number of days based on distance range.

The mile breaks transit time engine calculation is as follows:

$$Transit\ time\ days = Days$$

Where:
Days – The predefined real number of days that transportation can take for a particular distance.

In the following screenshot, the example setup for this transit time engine shows that for shipments with a mileage between 0 and 100, there will always be a transit time of 1 day, shipments with a mileage between 100 and 500 will have a transit time of 2 days, and shipments with a mileage between 500 and 10000 will have a transit time of 5 days. If you initialize base engine data in a new company, the details section will be empty and will need to be populated before using the engine.



Transit time engine:	MileBreaks
Name:	MileBreaks
Engine assembly:	Microsoft.Dynamics.Ax.Tms.dll
Engine type:	Microsoft.Dynamics.Ax.Tms.Bll.Mi

Details

Service	Start Miles	EndMiles	Days
STD	0	100	1
STD	100	500	2
STD	500	10000	5

To view the setup for the engine, go to Transportation management > Setup > Engines > Transit time engine. The setup for the mile breaks transit time engine is shown in the following table.

Transit time engine	Name	Engine assembly	Engine type
MileBreaks	MileBreaks	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.MileBreaksTransitTimeEngine

Point To Point Transit Time engine

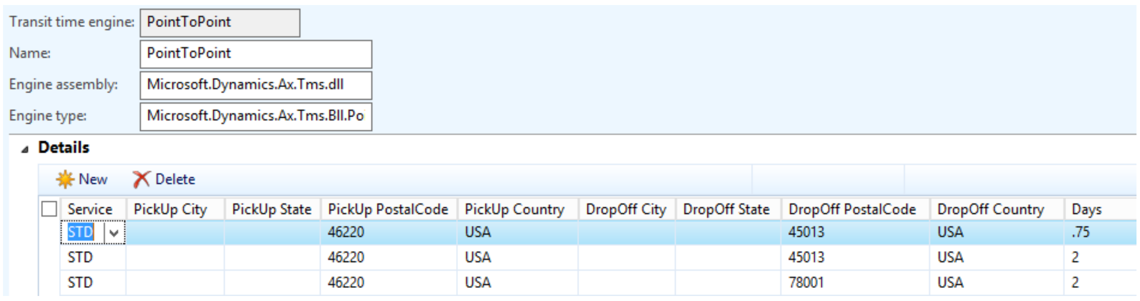
The point to point transit time engine defines the transit time based on origin and destination postal codes. For each origin and address, a predefined number of days can be set up.

The point to point transit time engine calculation is as follows:

$$Transit\ time\ days = Days$$

Where:
Days – The predefined real number of days that transportation can take for a particular distance. The data is recorded separately for each origin and destination address.

This engine calculates transit time in days, based on the postal codes in the shipment. In the following screenshot, the example shows where the origin postal code 46220 has three different drop-off postal codes with different transit times defined. When planning shipments from the origin postal code to any of the destination postal code, the specified transit time will be used. If you initialize base engine data in a new company, the details section will be empty and will need to be populated before using the engine.


Transit time engine:	PointToPoint
Name:	PointToPoint
Engine assembly:	Microsoft.Dynamics.Ax.Tms.dll
Engine type:	Microsoft.Dynamics.Ax.Tms.Bll.Po

Details

Service	PickUp City	PickUp State	PickUp PostalCode	PickUp Country	DropOff City	DropOff State	DropOff PostalCode	DropOff Country	Days
STD			46220	USA			45013	USA	.75
STD			46220	USA			45013	USA	2
STD			46220	USA			78001	USA	2

To view the setup for the engine, go to Transportation management > Setup > Engines > Transit time engine. The setup for the point to point transit time engine is shown in the following table.

Transit time engine	Name	Engine assembly	Engine type
PointToPoint	PointToPoint	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.PointToPointTransitTimeEngine

Mileage engines

Mileage engines calculate the transportation distance for shipments. If there is no distance data available for a shipment, some rate engines will not be able to calculate a rate for the shipment. To set up mileage engines, go to Transportation management > Setup > Engines > Mileage engine.

The fields in the Mileage time engine form are explained in the following table.

Field	Description
Mileage engine	The mileage engine identification field.
Name	The user friendly name of the engine.
Engine assembly	The name of the .NET assembly (dll) containing the engine definition. For all the engines shipped with Microsoft Dynamics AX 2012 R3, the name of the assembly is Microsoft.Dynamics.Ax.Tms.dll.
Engine type	The fully-qualified name (including namespace) of the class that defines the engine within the engine assembly.
Default	If selected, this mileage engine will be used by default for retrieving mileage.

There is only one mileage engine provided with Microsoft Dynamics AX 2012 R3. In the following blog post, you can learn how to set up a mileage engine that retrieves distance data from Bing Maps:
Transportation management mileage engine based on Bing maps.

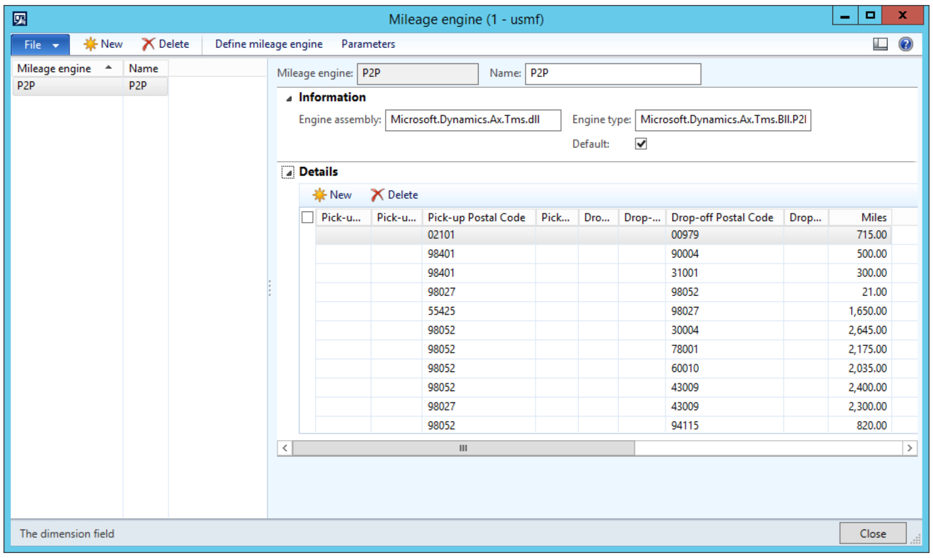
Point to Point Mileage engine

The point to point mileage engine retrieves mileage based on point to point data defined in Microsoft Dynamics AX.

To view the setup for the engine, go to Transportation management > Setup > Engines > Mileage engine. The setup for the point to point mileage engine is shown in the following table.

Mileage engine	Name	Engine assembly	Engine type
P2P	P2P	Microsoft.Dynamics.Ax.Tms.dll	Microsoft.Dynamics.Ax.Tms.Bll.P2PMileageEngine

In the following screenshot, an example using demo data for mileage is shown using predefined mileage for each origin and destination. This means that for every shipment, in order to retrieve mileage, you need to have mileage set up between the origin postal code and the destination postal code of the shipment. For example, with this setup, if you have a shipment going from postal code 98401, with a destination postal code of 31001, it would show as 300 miles.



Mileage engine	Name
P2P	P2P

Mileage engine: P2P
Name: P2P

Information

Engine assembly: Microsoft.Dynamics.Ax.Tms.dll
Engine type: Microsoft.Dynamics.Ax.Tms.Bll.P2I
Default: ✓

Details

Pick-u...	Pick-u...	Pick-up Postal Code	Pick...	Dro...	Drop-...	Drop-off Postal Code	Drop...	Miles
		02101				00979		715.00
		98401				90004		500.00
		98401				31001		300.00
		98027				98052		21.00
		55425				98027		1,650.00
		98052				30004		2,645.00
		98052				78001		2,175.00
		98052				60010		2,035.00
		98052				43009		2,400.00
		98027				43009		2,300.00
		98052				94115		820.00

The dimension field

Generic engines

Generic engines are simple auxiliary engines that are used by other engines that do not require data from Microsoft Dynamics AX 2012 R3. For example, an apportionment engine. Apportionment engines are used to reduce the final costs of transportation to specific orders and lines based on dimensions, such as volume and weight.

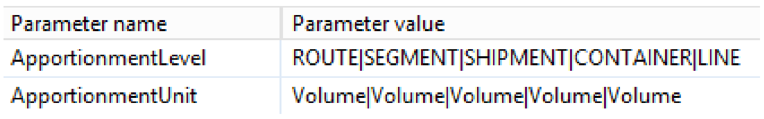
Generic Weight Apportionment engine

The generic weight apportionment engine aggregates weight in the route structure based on system weight, such as the total weight on shipment, segment, and route. System weight is typically defined by the product master data. This engine is typically used by the LTL rate engines, and should not be used for apportionment of transportation charges.

Generic Apportionment engine

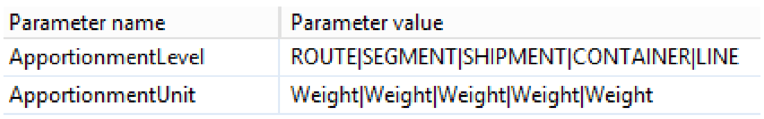
The generic apportionment engine apportions the freight charges based on different dimensions, such as by volume, weight, or the number of freight pieces. The type of the dimension to apportion on is set in the Engine parameters form.

By volume:



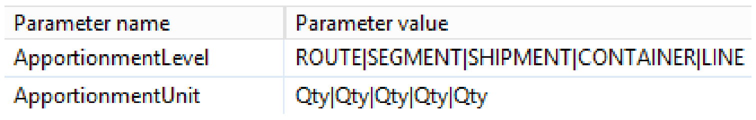
Parameter name	Parameter value
ApportionmentLevel	ROUTE|SEGMENT|SHIPMENT|CONTAINER|LINE
ApportionmentUnit	Volume|Volume|Volume|Volume|Volume

By weight:



Parameter name	Parameter value
ApportionmentLevel	ROUTE|SEGMENT|SHIPMENT|CONTAINER|LINE
ApportionmentUnit	Weight|Weight|Weight|Weight|Weight

By quantity:



Parameter name	Parameter value
ApportionmentLevel	ROUTE|SEGMENT|SHIPMENT|CONTAINER|LINE
ApportionmentUnit	Qty|Qty|Qty|Qty|Qty

Freight bill type

Freight bill types standardize the freight invoice and the freight bill lines and are used for automatic freight bill matching.

Conclusion

This white paper has provided detailed information about all of TMS engines that are shipped with Microsoft Dynamics AX 2012 R3. It explains the architectural background of the engines and the relation between the different types of engines. It also describes the initialization of the engines and the metadata configuration that is needed.

Additional resources

Use the following links to access blog posts and white papers on the subjects not covered in this document or resources that provide supplemental guidance:

- Transportation management online Help on TechNet
- Transportation management system implementation guide
- Implementing and deploying transportation management engines
- Transportation management mileage engine based on Bing maps
- Third-party transportation management engines (UPS, FedEx, Progistics, PCMiler)
- Using transportation management without the new warehouse management module

Send feedback.

Microsoft Dynamics is a line of integrated, adaptable business management solutions that enables you and your people to make business decisions with greater confidence. Microsoft Dynamics works like and with familiar Microsoft software, automating and streamlining financial, customer relationship, and supply chain processes in a way that helps you drive business success.

United States and Canada toll free: (888) 477-7989
Worldwide: (1) (701) 281-6500
www.microsoft.com/dynamics

Microsoft Dynamics