

Material Operation Tables

A Material Operation Table is a CSV file, where each row contains a material tracking operation.

The first line of the file must contain comma-separated header identifiers. The system will look for the following reserved headers (case sensitive):

TIME	Represents the timestamp for the operation. Contains a positive integer value that must fit in 64bits.
OP	<p>It contains the operation type for the row. The possible operations are:</p> <ol style="list-style-type: none">1. load – Starts tracking a new volume of material2. unload – Makes material disappear from the tracking system3. reload – Makes material re-appear in the tracking system4. move – Tracks movement from one point in space to another5. update – Changes attribute values for one volume of material6. expand – Grows the volume of tracked material and branches the tracking in both space and time7. jump – Moves material from one singularity to another <p>The operations will be discussed in more detail below.</p>
FROM_X	Contains the X coordinate, in Project coordinates, of the source voxel
FROM_Y	Contains the Y coordinate, in Project coordinates, of the source voxel
FROM_Z	Contains the Z coordinate, in Project coordinates, of the source voxel
FROM_TIME	A timestamp used by some singularity operations (jump and reload) to unambiguously refer to a tracked portion of material in space-time.
FROM_ID	Contains the identifier of the singularity from which the material has reappeared
TO_X	X coordinate, in Project coordinates, of the destination voxel
TO_Y	Y coordinate, in Project coordinates, of the destination voxel

TO_Z	Z coordinate, in Project coordinates, of the destination voxel
TO_ID	Contains the identifier of the singularity where the material disappeared.

The following table lists the available operations and their effects:

load	<p>The load operation starts tracking a volume of material at the specified location/time. The operation also assigns values to attributes within the volume.</p> <p>Parameters: TIME, TO_X, TO_Y, TO_Z [, attributes]</p>
unload	<p>The unload operation tells the system a volume of material has disappeared from the spatial domain and it has entered a singularity, for instance a truck or a crusher. The operation moves from a voxel address to an opaque singularity ID.</p> <p>Parameters: TIME, FROM_X, FROM_Y, FROM_Z, TO_ID</p>
reload	<p>The reload operation tells the system a volume of material has reappeared into the spatial domain from a singularity. For instance, material can temporarily disappear into a truck and emerge later from the truck at the dump location.</p> <p>Parameters: TIME, FROM_X, FROM_Y, FROM_Z, FROM_TIME, FROM_ID, TO_X, TO_Y, TO_Z</p>
update	<p>The update operation writes new values into attributes for the tracked material, but does not change the identity of the tracked material. This operation is used to model a situation where the contents of a tracked volume of material have changed.</p> <p>Parameters: TIME, FROM_X, FROM_Y, FROM_Z [, attributes]</p>
move	<p>The move operation removes a tracked volume of material from one location and places it into a different location.</p> <p>Parameters: TIME, FROM_X, FROM_Y, FROM_Z, TO_X, TO_Y, TO_Z</p>
expand	<p>The expand operation branches the current volume of tracked material into a new location of space, and starts a new genealogy branch there. The original material ID is preserved, however, allowing to track expanded material back to its origin.</p> <p>Parameters: TIME, FROM_X, FROM_Y, FROM_Z, TO_X, TO_Y, TO_Z</p>
jump	<p>Moves material from one singularity to another.</p> <p>Parameters: TIME, FROM_X, FROM_Y, FROM_Z, FROM_TIME, FROM_ID, TO_ID</p>

In addition to these special columns, the CSV may contain custom columns, one for each attribute that is tracked.

The system supports two types of columns: numerical and string. The type of a column will be inferred from the column's identifier. To specify a column that contains string values, the "\$" character must appear at the beginning of the column's name.

The following image shows an example of a Material Operation Table:

TIME	OP	FROM_X	FROM_Y	FROM_Z	TO_X	TO_Y	TO_Z	TO_ID	fe
0	load				0	0	0		10
0	load				0	0	1		20
0	load				0	0	2		5
0	load				0	0	3		30
100	unload	0	0	0				DT01	
100	unload	0	0	1				DT01	
100	unload	0	0	2				DT01	
100	unload	0	0	3				DT01	

This table models an excavator-to-truck load event.

The first four entries perform load operations over four different voxels (vertically stacked along Z) and assign different iron grades to them. In this example, their timestamp is zero, reflecting the fact the model considers the resource model to be there from the beginning of time.

The next four entries are about the actual excavator operation, which occurred at timestamp 100. Here, the four stacked voxels are unloaded from the tracking system and sent to a singularity, a truck with unique ID "DT01".

Singularities represent entities in the system where material can disappear, and from where material could later reappear. Unlike voxels in the material tracking layer, singularities do not have a spatial address, rather an application-assigned ID that is opaque to the material tracking system.

Since they could have zero volume, but may have received vast amounts of material (e.g. a crusher), singularities cannot be stored spatially. They require traditional sequential storage, for instance, a table in a relational database.

The spatial and relational aspects of the material tracking are deeply intertwined. The material tracking system must be equally aware of these non-spatial constructs, since queries will ultimately

involve spatial and relational elements at the same time, allowing to reconcile spatial facts to relational facts.

Singularities also provide means for simulations to access information about their preconditions. For instance, when simulating a dump truck event, the simulation agent only needs to know the singularity ID for the truck and the timestamp at which the dump occurred. From this, it can obtain the voxel contents of the truck from the singularity.

The following Material Operation Table shows the results of simulating a dump event on a stockpile, based on the previous table:

TIME	OP	FROM_X	FROM_Y	FROM_Z	FROM_ID	TO_X	TO_Y	TO_Z	fe
200	reload	0	0	0	DT01	10	10	0	
200	reload	0	0	1	DT01	10	10	1	
200	reload	0	0	2	DT01	10	10	2	
200	reload	0	0	3	DT01	10	10	3	
200	update	10	10	0					16.25
200	update	10	10	1					16.25
200	update	10	10	2					16.25
200	update	10	10	3					16.25

The first four entries declare the material that went into truck DT01 is now back at the site, but at a different location.

The next four entries set new attribute values. This could be to model a change of grade in the tracked material due to transportation, mixing at the time it was dumped, and other simulated factors.

It is equally possible to model the same process described by the two tables above without using a singularity. The following Material Operation Table shows this approach:

TIME	OP	FROM_X	FROM_Y	FROM_Z	TO_X	TO_Y	TO_Z	fe
0	load				0	0	0	10
0	load				0	0	1	20

0	load				0	0	2	5
0	load				0	0	3	30
200	move	0	0	0	10	10	0	
200	move	0	0	1	10	10	1	
200	move	0	0	2	10	10	2	
200	move	0	0	3	10	10	3	
200	update	10	10	0				16.25
200	update	10	10	1				16.25
200	update	10	10	2				16.25
200	update	10	10	3				16.25

While the outcome is the same, and voxel ledgers are much shorter, this approach does not allow to properly answer where the material was in the time span from 100 to 200, as it still thinks the material has not moved from the original location in the resource model until timestamp 200, where we learn the material has appeared at the bottom of a stockpile. Equally, the system will not be able to answer any queries that involve trucks, for instance, visualizing the origin of the material currently carried by a truck.

Revision #1

Created 17 March 2025 20:48:40 by admin

Updated 17 March 2025 20:50:35 by admin