

SmartSpace Paths and Queues Configuration Guide

For version 3.4.1

Ubisense Limited, St Andrew's House, St Andrew's Road, Cambridge CB4 1DL, United Kingdom.

Telephone: +44 (0)1223 535170. Website: http://www.ubisense.net.

Copyright © 2018, Ubisense Limited 2014 - 2018. All Rights Reserved. You may not reproduce this document in whole or in part without permission in writing from Ubisense at the following address:

Ubisense Limited St Andrew's House St Andrew's Road Cambridge CB4 1DL United Kingdom Tel: +44 (0)1223 535170

WWW: http://www.ubisense.net

All contents of this document are subject to change without notice and do not represent a commitment on the part of Ubisense. Reasonable effort is made to ensure the accuracy of the information contained in the document. However, due to on-going product improvements and revisions, Ubisense and its subsidiaries do not warrant the accuracy of this information and cannot accept responsibility for errors or omissions that may be contained in this document.

Information in this document is provided in connection with Ubisense products. No license, express or implied to any intellectual property rights is granted by this document.

Ubisense encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.

Windows[®] is a registered trademark of Microsoft Corporation in the United States and/or other countries. The other names of actual companies and products mentioned herein are the trademarks of their respective owners.

Contents

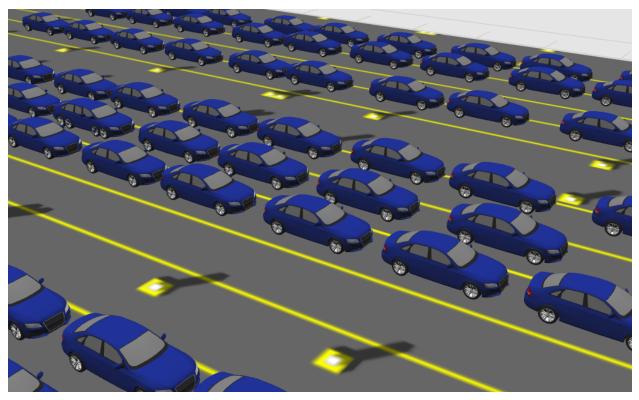
Contents	i
Overview of Paths and queues	1
What is Paths and queues?	1
Paths	1
Path constraints	2
Path sections	2
Path points	2
Path groups	3
How do I get Paths and queues?	3
Version and license	3
Installation	3
Where does Paths and queues appear?	4
Service Manager	4
Application Manager	5
SmartSpace Config	6
Creating the data model for Paths and queues	9
Using the PATHS task	9
Draw a path	9
Add path constraints	10
Generate path points	12
Moving path points	14
Controlling path tracking	16
Path-to-group mapping	16
Default containment	17
Controlling objects using business rules	18
Advanced path tracking parameters	19
default stderr	20
handover variance threshold	20

innovation multiplier	21
max consecutive outliers	21
max valid position variance	21
max variance before reset	21
outlier distance	22
raw stderr multiplier	
stickiness	22
variance multiplier	22
Paths and queues simulation	23
What is the simulator?	23
Services	
Admin tool	
Service parameters	
Example simulation walkthrough	24
Create the XML script	24
Create objects to attach to simulated tags	25
Attach tags to the objects	25
Create paths	27
Assign a path group	
Monitor the spatial relation	
Run the simulation	29

Overview of Paths and queues

What is Paths and queues?

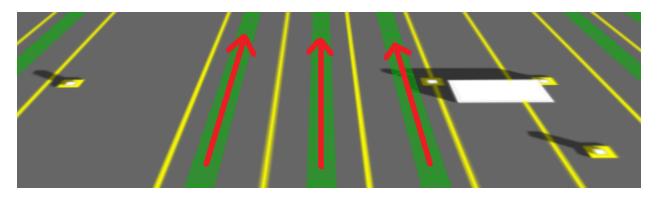
Paths and queues is a feature introduced into the Location Rules component of SmartSpace 3.4. Its purpose is to allow you to introduce prior knowledge of object locations in your system, in particular when objects travel along fixed paths like this:



When you configure *Paths and queues* to control objects, they will be snapped to nearby paths as tags move around. You can introduce further constraints to control object speed and separation such that objects form queues along the paths. You can then use the object sequence information in your application to report things like "number of vehicles in front".

Paths

Paths are physical routes that you know objects will follow. You draw these in SmartSpace Config:



Paths configured to keep object neatly positioned within the lanes

Path constraints

Path constraints are normally intended to match the real-world, physical, immutable constraints of the thing being modeled. In SmartSpace 3.4, the following constraints are configurable:

- *Rotates* object rotations, which can change as they move along a path
- *HasSpeed* object speeds
- *IsInTrain* object separations

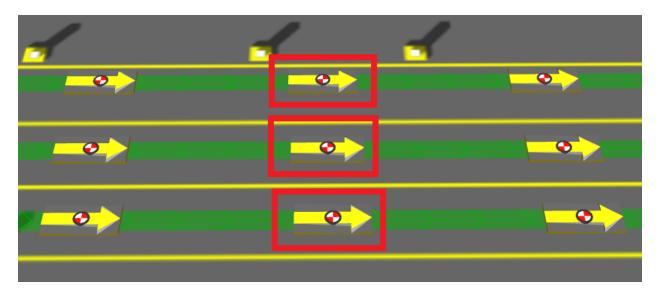
For example, a production line might move at a known constant speed that won't change over time. In this case, you could use the *HasSpeed* constraint.

Path sections

Paths can be split into sections in order to assign different constraints to different sections of the path.

Path points

A path point is an object intended to be located exactly on a path. There is an object type called *Path Point* that you can use directly, or inherit, to give objects special UI features to get them to snap to paths or generate them at regular intervals.



Path points, automatically generated at regular intervals

Path groups

A path group is a logical grouping of paths used to determine which objects are being controlled by the path tracking location rules. A path group has an extent which, by default, is used to determine which objects to control.

How do I get Paths and queues?

Version and license

You will need SmartSpace 3.4 with a license for Location rules.

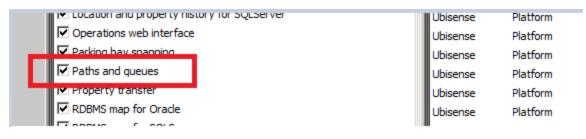
Installation

Use the INSTALL SERVICES task in Service Manager to install service packages:

🟮 Service Manager			
MANAGE SERVICES	Select the packages directory from which to install C:\Users\jonathan.price\Documents\Ubisense Releases\ss_3_3_rc2_14	12 17\Sr Br	owse D:Vhome
INSTALL SERVICES	Select features to install	Vendor	Package A
Select features and install the required	AVL/GPS connect	Ubisense	Application inter
service packages	AngleID connect	Ubisense	Application intec
	Application .NET API	Ubisense	Application intec
CLEANUP SERVICES	Application REST API	Ubisense	Business rules
	Automated tag association	Ubisense	Business rules
	Business object properties	Ubisense	Business rules
	Business rules engine	Ubisense	Business rules
	Driven objects	Ubisense	Business rules
	Email	Ubisense	ISO24730
	Failover control	Ubisense	ISO24730
	ISO 24730	Ubisense	ISO24730
	LLRP interface	Ubisense	IT support
	Location and property history for Oracle	Ubisense	IT support
	Location and property history for SQLServer	Ubisense	Location rules
	Operations web interface	Ubisense	Location rules
	Parking bay snapping	Uhisense	Location rules

Installing services using Service Manager

If you have a license, it will appear in the list of features:



Selecting the Paths and queues feature in Service Manager

Where does Paths and queues appear?

Once you have installed *Paths and queues*, since it is integrated into existing SmartSpace components, it will appear in various places in Ubisense programs.

Service Manager

In MANAGE SERVICES, there are some new Location rules services:

Ubisense	Location rules	Containment inference	3.3.6713
Ubisense	Location rules	Driven objects configuration	3.3.6713
Ubisense	Location rules	Location inference	3.3.6713
Ubisense	Location rules	Location inference	3.3.6713
Ubisense	Location rules	Parking and stale configuration	3.3.6713
Ubisense	Location rules	Path constraint configuration	3.3.6713
Ubisense	Location rules	Path simulation configuration	3.3.6713
Ubisense	Location rules	Path simulator	3.3.6713
Ubisense	Location rules	Path simulator	3.3.6713
Ubisense	Location rules	Path tracking configuration	3.3.6713

Service Manager showing Paths and queues services

Application Manager

In **DOWNLOADABLES**, there are new items in the Location rules topic:

📮 Application Manager		
APPLICATIONS	Download selected items • Group by topic •	Group by type
DOWNLOADABLES	Downloadable	
Download tools and documentation to	Ubisense	
this computer	H Business rules	
	H ISO24730	_
	Location rules	
	Path simulation tools	
	ubisense_path_simulation_admin.exe	
	Path tracking admin tools	
	ubisense_path_tracking_admin.exe	
	⊥ Location system	
	Platform	
	RFID integration	
	H RTLS integration	
	SmartSpace core	

Application Manager showing Paths and queues downloadables

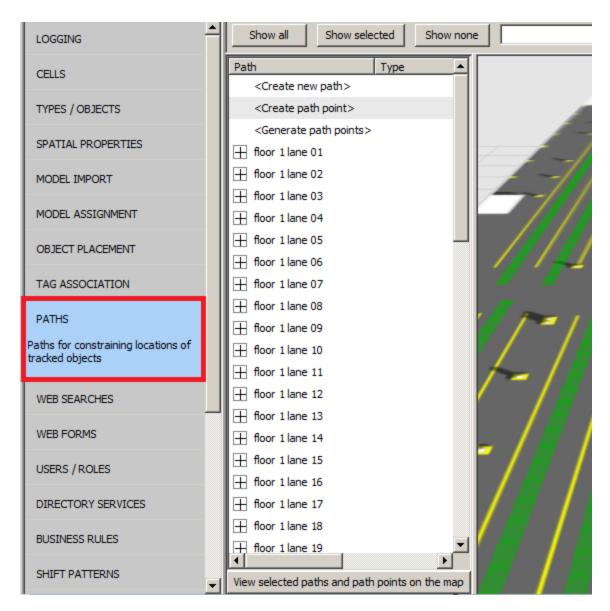
SmartSpace Config

In **TYPES / OBJECTS**, there are 2 new types:

SmartSpace Config				
	Туре	Parents	Identified By 🔺	
LOGGING	Object			🤨 Path Group objects
CELLS	Parking Area	Assertion Area		Show only matching items
	Parking Bay	Assertion Point		
TYPES / OBJECTS	Path Group	Object		<create new="" object=""></create>
ata model of object types, objects	Path Point	Object		1st floor group
nd their properties	Tag Association Point	Object		2nd floor group
SPATIAL PROPERTIES	Web Location Source	Object	-	3rd floor group
SPATIAL PROPERTIES				
MODEL IMPORT	Properties of Object	Туре	Inherited From	🛑 Path Point objects
	<create new="" property=""></create>			<show items<="" matching="" only="" td=""></show>
MODEL ASSIGNMENT	delete pending flag	Bool		<sriow items<="" matching="" only="" td=""></sriow>
OBJECT PLACEMENT	ignore tag locations flag	Bool		
	path group	Path Group		<create new="" object=""></create>
TAG ASSOCIATION	remove location pending flag	Bool		
DATUS	remove tag pending flag	Bool		
PATHS	stale flag	Bool		
WEB SEARCHES	•			<u> </u>

SmartSpace Config TYPES / OBJECTS automatically has 2 new types

There is a new task called **PATHS**:



SmartSpace Config PATHS task

In SERVICE PARAMETERS, there are new options in the drop-down list:

WEB FORMS	Path tracking			•	Drag obj
USERS / ROLES	Path			•	
DIRECTORY SERVICES	Object			_	
BUSINESS RULES	<all objects="" of="" path="" type=""> floor 1 lane 01</all>				
SHIFT PATTERNS	floor 1 lane 02				
EMAIL	floor 1 lane 03 floor 1 lane 04				
RDBMS MAP	floor 1 lane 05 floor 1 lane 06				
PROPERTY HISTORY	floor 1 lane 07			•	
	Parameter Name	Туре	Default	▲	
SERVICE PARAMETERS	combined multiplier	Double	1		
Runtime parameters for various services	default stderr	Double	0.2		
services	group	Path Group			
TRACE VIEWER	handover variance threshold	Double	10		
	innovation multiplier	Double	1		
RULES ENGINE TRACE	max consecutive outliers	Int	5		
	max valid position variance	Double	10		

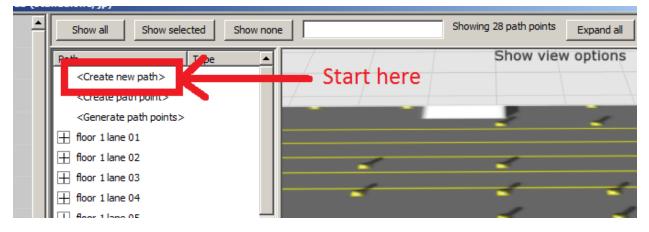
SmartSpace Config path tracking service parameters

Creating the data model for Paths and queues

Using the PATHS task

Use the **PATHS** task in SmartSpace Config to model the physical layout and constraints of your environment.

Draw a path



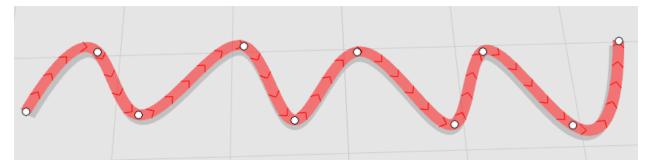
Getting started in SmartSpace Config PATHS

When drawing a path, read the instructions carefully to ensure you know all the capabilities of the tool. You can line paths up neatly using the *Snap grid* and *Path height* controls.

Show all Show selected Show none		Showing 28 path points	Expand all Collapse all
Path name floor 3 lane 01 Snap grid 0.1 : Path height 12 :	CREATE PATH ADD POINT REMO E POINTS SMOOTH PATH SELECT POINTS MULTI SELECT TOGGLE SELECTION SELECTION RECTANGLE SELECT ALL SELECT NONE DRAG POINTS UNSNAPPED HORIZONTALLY ERTICALLY FREELY ALONG X AXIS ALONG X AXIS ALONG Z AXIS CANCEL COMMANDS UNDO REDO HIDE / SHOW	<pre>shift + click delete select points ctrl + s select points ctrl + shift + s shift + click points ctrl + click points 'r' + hold down left mouse buttor 'r' + hold down left mouse buttor 'r' + hold ctrl left mouse button right mouse button right mouse button 'x' + left button 'x' + left button 'x' + left button 'z' + left button esc ctrl + z ctrl + y 'h'</pre>	Hide view options 2D3[Show background objects Show foreground objects Show paths Show path points Show cords when editing Show cell extents Show name labels Show path sequence labels Show help
Drawing instructions			4
Path being drawn	7		

Drawing a path in SmartSpace Config PATHS

See the green drawing instructions on the map for how to smooth a path to look like this:



Drawing a smoothed path in SmartSpace Config PATHS

The path will always go through the control points, so if the smoothing doesn't behave as required, add more control points.

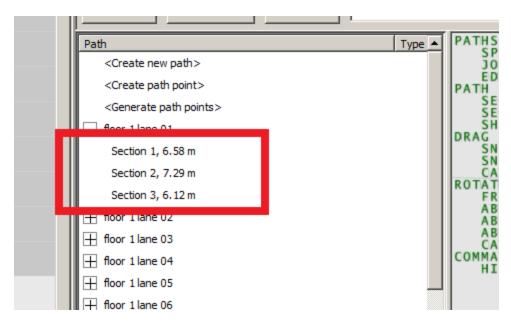
Add path constraints

If your environment has physically constrained paths, use the path editor to add constraints:

Path name lane 1 Snap grid 0.1 Path height 1	
Object constraints:	
✓ Rotates	Object will rotate as it
rotation initial yaw 0	moves along the path
rotation initial pitch 0	
rotation initial roll 0	
rotation total yaw 360	
rotation total pitch 0	
rotation total roll 0	
rotation absolute	If correct, this helps
HasSpeed	the location filtering
speed value 0.01	
speed std dev 0	
🔲 IsInTrain	
Save Cancel	\rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow

Adding path constraints in SmartSpace Config PATHS

If your path has different constraints in different sections, you can split the path by following the instructions in the in-place help in the **PATHS** map.



A path split into three sections in SmartSpace Config PATHS

Double-click individual path sections to edit constraints separately.

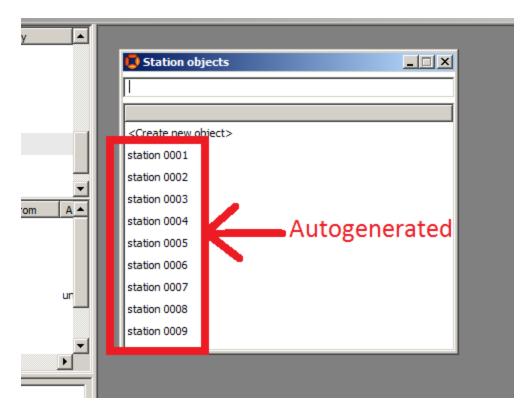
Generate path points

Use *<Generate path points>* to automatically generate objects at regular intervals along paths. You can create types that inherit from *Path Point* in **TYPES / OBJECTS**. To avoid confusion, give them a representation using **MODEL IMPORT** and **MODEL ASSIGNMENT** before creating any.

-	Show all Show selected Show none	Showing 28 path points
		Create types in TYPES / OBJECTS
	Choose the type of path point	
	Station	
	Choose a path on which to generate path points	¥
	floor 1 lane 02	<u> </u>
	Enter the start offset	y > > > > > > > > > > > > > > > > > > >
	10	
	Enter the interval between path points	
	10	
	Enter the path point name prefix (optional)	× ×
	station	
	Enter the first path point number Use leading zeroes to set the naming scheme width	
	0001	
	Enter the path point name postfix (optional)	
	Save Cancel	

Generating path points in SmartSpace Config PATHS

There is nothing special about objects created in this way, other than the fact that their name and location were generated by SmartSpace Config. You can see them in **TYPES / OBJECTS**, give them properties, and use them in SmartSpace Business rules if you have the appropriate licenses.



Path points are normal objects in SmartSpace Config TYPES / OBJECTS

Moving path points

Since they are normal objects, you can drag path points around in **OBJECT PLACEMENT**. However, this doesn't snap them to paths. The **PATHS** map will warn you when path points are not on a path:



SmartSpace Config PATHS map warns about orphaned path points

Fix this error by double-clicking on the path point to bring up the editor:

	Sho
Enter the path point name	
pp 0003	
Choose the type of path point	
Path Point	
Select a path for the path point	
Enter the offset along the path	
0	
Yaw	This path point is not on a path!
-0.0968922	
Pitch	
-0	
Roll	
0	
Unset path point location	Drag on the map or type
	in the editors
Save Cancel	

Moving a path point using the editor in SmartSpace Config PATHS

Controlling path tracking

Path-to-group mapping

You must assign paths to groups using **SERVICE PARAMETERS**. Select *Path tracking* and *Path* in the dropdowns. Drag the *group* parameter onto the right-hand panel. Assign path groups as required.

Path tracking				📮 Path tracking : Pat	:h : group
Path			•	Create new parameter	row>
Object				floor 1 lane 01	1st floor group
<all objects="" of="" path="" type=""></all>				floor 1 lane 02	1st floor group
floor 1 lane 01				floor 1 lane 03	1st floor group
floor 1 lane 02				floor 1 lane 04	1st floor group
floor 1 lane 03	efault is e	empty -		floor 1 lane 05	1st floor group
floor 1 lane 04 V	ou must s	set this		floor 1 lane 06	1st floor group
floor 1 lane 05				floor 1 lane 07	1st floor group
floor 1 lane 06			•	floor 1 lane 08	1st floor group
Parameter Name	Type	Default		floor 1 lane 09	1st floor group
combined multiplier	Duvb	1		floor 1 lane 10	1st floor group
1.6.11.11		<u>^</u>	-11	floor 1 lane 11	1st floor group
group	Path Group			floor 1 lane 12	1st floor group
nanuover vanance unesnou	Double	10		floor 1 lane 13	1st floor group
innovation multiplier	Double	1		floor 1 lane 14	1st floor group
max consecutive outliers	Int	5		floor 1 lane 15	1st floor group

Assigning paths to groups in SmartSpace Config SERVICE PARAMETERS

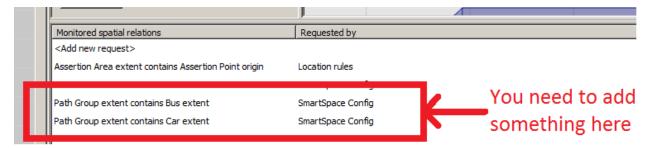
Default containment

The default path tracking behavior is to control objects that are contained in the extent of a path group. To set this up, create objects of type *Path group* in **TYPES / OBJECTS**. Set their *extent* property in **SPATIAL PROPERTIES**:

Path Group	
extent 💌	
Default Spaces <create default="" new="" space=""></create>	
Filter object names <all objects=""> Specific Spaces <create new="" space="" specific=""></create></all>	
1st floor group 2nd floor group 3rd floor group	
Edit Space	

Creating path group extents in SmartSpace Config SPATIAL PROPERTIES

Create a SPACE property for the objects to be controlled in **TYPES / OBJECTS** and **SPATIAL PROPERTIES** and monitor the spatial relation:



Setting up default containment in SmartSpace Config SPATIAL PROPERTIES

Controlling objects using business rules

For some applications, you might need to decide which objects are controlled by path tracking based on some other business logic. For this, you will need SmartSpace Business rules licenses.

First, turn off default containment in SERVICE PARAMETERS:

-	Path tracking	Path tracking : Path Group : use default containment
	Path Group	
	Object	<all group="" objects="" of="" path="" type=""> false</all>
	<all group="" objects="" of="" path="" type=""></all>	
	1st floor group	
	2nd floor group	You can turn off default
	3rd floor group	fou can turn on default
		containment for all path
	Darameter Name Turne Default	
	use default containment Bool true	groups at once
•		default is to use default containment

Turning of default containment in SmartSpace Config SERVICE PARAMETERS

Then, use **BUSINESS RULES** to set and unset rows for *<Path Group* > *controls <Object* > as required:

when			
as	sertion point has located object		
becomes	true do		
set	the path group of (assertion point) controls (object)	to	true

Setting <Path Group> controls <Object> using SmartSpace Config BUSINESS RULES

Advanced path tracking parameters

Path tracking uses filtering to estimate the offsets of objects along paths. You might need to change filter parameters for some or all paths in your model, depending on the layout, quality of the location system, and so on. For example, if you have lots of lanes alongside each other, you might need to make them "stickier" such that objects change lane less readily.

Always test your application with the default values before changing anything described in this section.

In this context, "variance" means how uncertain the filter for a particular path is that it has the object in the right place.

📮 Path tracking parameters for <all objects="" of="" path="" type=""> 📃 🔲 🗙</all>					
default stderr	0.2				
group					
handover variance threshold	10				
innovation multiplier	1				
max consecutive outliers	5				
max valid position variance	10				
max variance before reset	10				
outlier distance	5				
raw stderr multiplier	1				
stickiness	2				
variance multiplier	1				
Edit					

Using SmartSpace Config SERVICE PARAMETERS to change path tracking filter parameters

default stderr

The value to use when the sensor system fails to provide an estimate of the measurement accuracy.

Do not normally change this parameter.

handover variance threshold

Filter variance threshold above which path tracking will allow other location rules to be applied.

- higher = path tracking will hold onto objects more, possibly without generating locations
- lower = path tracking will allow other location rules to take over more easily

Set this higher if tags are coming off paths too easily.

Set this lower if tags are getting stuck on paths when you want other location rules to take over.

innovation multiplier

Multiplier applied to the inferred error based on the distance from the estimated tag position to the measurement. For example, if this is zero, the distance from the estimated tag position to the measurement will not directly affect the filter variance at all.

- higher = measurements further from the estimated tag position will be trusted less
- lower = distance from estimated tag position to measurement doesn't matter as much

Set this higher if tag locations that are far away from the estimated tag position are placing the object on the path incorrectly.

max consecutive outliers

The maximum number of consecutive measurements deemed to be outliers before the filter resets to the nearest point on the path to the next measurement.

- higher = filter will reset less readily; objects will make fewer big jumps
- lower = objects will jump to the nearest point to the measurement more easily

max valid position variance

The maximum value of the filter variance for which an object location will be generated.

- higher = generate more locations, potentially of lower quality
- lower = generate fewer locations, but with more certainty that they are correct

Set this lower if your application requires locations to be generated with more certainty.

Note that it doesn't make sense for this to be larger than *max variance before reset*; the filter variance will never be higher than that (because it will reset instead).

Note that it doesn't make sense for this to be larger than *handover variance threshold*; the path tracking location rule will have handed over to other rules already.

max variance before reset

The maximum possible variance of the filter state. When the variance goes over this value, the filter resets to the nearest point on the path to the next measurement.

• higher = filter will reset less readily; objects will make fewer big jumps

• lower = objects will jump to the nearest point to the measurement more easily

outlier distance

The minimum distance between the estimated tag position and the measurement where the reading is defined as an outlier. See *max consecutive outliers* for a description of what outliers do.

- higher = filter will reset less readily; objects will make fewer big jumps
- lower = objects will jump to the nearest point to the measurement more easily

Set this higher if your readings are very noisy and you want to stop objects making a lot of jumps.

raw stderr multiplier

Multiplier applied to the raw error computed by the sensor system. For example, if this is zero, sensor measurements will be assumed to be completely correct by the filter.

- higher = noisy measurements from the sensor system will be trusted less
- lower = sensor system noise doesn't matter as much

Set this higher if noisy sensor measurements are placing the object on the path incorrectly.

stickiness

Number of consecutive tag measurements for which a path filter needs the lowest variance in order to take control of the object.

- higher = path will hold onto objects; objects will flicker between paths less
- lower = path will release objects to other paths in the same group more easily

Set this higher if objects jump back and forth incorrectly between paths.

variance multiplier

Multiplier applied to the variance prediction as time elapses. When the filter predicts a new state, the variance increases because time has elapsed since the last measurement. The elapsed time is multiplied by this value when increasing the variance over time.

- higher = filter will be quicker to reset or handover to other paths or location rules
- lower = filter will take longer to reset or handover to other paths or location rules

Paths and queues simulation

What is the simulator?

The SmartSpace Location rules product includes some support for simulating tags or objects moving along paths. This is a good way to ensure you have set everything up before deploying *Paths and queues* into production.

Services

Path simulation services are included in the service packages for the SmartSpace Paths and queues feature:

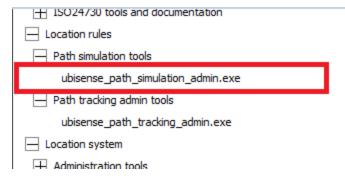
Ubisense	Location rules	Parking and stale configuration	3.3.6713	Site	
 1 this second	Less time will be	Dath constraint configuration	2.2.6712	Cit-	
Ubisense	Location rules	Path simulation configuration	3.3.6713	Site	
Ubisense	Location rules	Path simulator	3.3.6713	Location Cell 00001	
Ubisense	Location rules	Path simulator	3.3.6713	Location Cell 00002	
Ubisense	Location rules	Path tracking configuration	3.3.6713	Site	۰.
Ubisense	Location system	Boot server	3.3.6713	Site	

Path simulation services shown in Service Manager MANAGE SERVICES

The *Path simulator* services wait for a simulation request, so you can leave all these services running all the time. Simulation only starts when you use the admin program to request a simulation.

Admin tool

You can download the simulation admin tool using Application Manager DOWNLOADABLES:



Path simulation admin tool shown in Application Manager DOWNLOADABLES

Use the admin tool to set the service parameters used by the simulator.

Service parameters

SmartSpace Config **SERVICE PARAMETERS** shows the parameters that are set by the admin tool and consumed by the simulator services:

USERS / ROLES	Path tracking simulation	ı		Drag objects and
DIRECTORY SERVICES	Location Cell		•	
BUSINESS RULES	Object			
SHIFT PATTERNS	<all loca<br="" objects="" of="" type="">Location Cell 00001</all>	ition Cell>		
EMAIL	Location Cell 00002			
RDBMS MAP				
PROPERTY HISTORY				
SERVICE PARAMETERS	Parameter Name	Туре	Default	
Runtime parameters for various	pause simulation	Bool	false	
services	simulation wrapper	String		
	start simulation	Bool	false	
TRACE VIEWER	stop simulation	Bool	false	
RULES ENGINE TRACE				
	- -			

Path tracking simulation in SmartSpace Config SERVICE PARAMETERS

Example simulation walkthrough

Create the XML script

To create a simulation script, start by running this command:

ubisense_path_simulation_admin.exe example > path_simulation.xml

Edit path_simulation.xml to see the format and example data. The following steps will make it work without modification.

Create objects to attach to simulated tags

In SmartSpace Config TYPES / OBJECTS, create 20 objects of type "Car":

Туре	Parents	Identified By				
<create new="" type=""></create>				Car objects		
AngleID Recipe	Object	name		VIN	00001	
Assertion Area	Object				00002 00003	
Assertion Point	Object				00004	
Bus	Object	number			00005 00006	
Bus 40 Ft	Bus	number			00007 00008	
Bus 60 Ft	Bus	number			00009	
Car	Object	VIN			00010 10001	
GPS Reference Point	Object				10002 10003	
-000.7	<u>əl:</u>		<u> </u>		10004	
Properties of Car	Туре	Inherited From			10005 10006	
<create new="" property=""></create>					10007	
VIN	String				10008 10009	
delete pending flag	Bool	Object			10010	
extent	Space					
ignore tag locations flag	Bool	Object		delete pending flag	false	•
path group	Path Group	Object		ignore tag locations flag	false	
remove location pending flag	Bool	Object		path group	Tube	
remove tag pending flag	Bool	Object		paurgroup		_
stale flag	Bool	Obiect .	21	remove location pending flag	false	
			_	remove tag pending flag	false	•
<show items="" matching="" only=""></show>				stale flag	false	
Complex Property		Туре			Tube	
<create complex="" new="" property=""></create>						
'Assertion Area' contains 'Assertion Point'	1	Bool				
'Assertion Area' controls 'Object'		Bool		Save Cancel		
'Path Group' controls 'Object'		Bool				

Creating 20 cars in SmartSpace Config TYPES / OBJECTS

Make sure "Car" has an assigned representation model by using **MODEL IMPORT** and **MODEL ASSIGNMENT**.

Attach tags to the objects

Use **TAG ASSOCIATION** to attach tags. The ids need to match those that will be simulated in the XML script:

Tag Id.	Owner	Position Name	X	Y	Z
<associate object="" tag="" with=""></associate>					
00:11:CD:00:00:00:10:E1	1290	Roof	7.344	0	3
00:11:CE:00:00:00:00:01	00001	Origin	0	0	0
00:11:CE:00:00:00:00:02	00002	Origin	0	0	0
00:11:CE:00:00:00:00:03	00003	Origin	0	0	0
00:11:CE:00:00:00:00:04	00004	Origin	0	0	0
00:11:CE:00:00:00:00:05	00005	Origin	0	0	0
00:11:CE:00:00:00:00:06	00006	Origin	0	0	0
00:11:CE:00:00:00:00:07	00007	Origin	0	0	0
00:11:CE:00:00:00:00:08	00008	Origin	0	0	0
00:11:CE:00:00:00:00:09	00009	Origin	0	0	0
	00010	Origin	0	0	0

<Create new tag range>

<Assign tag range to type>

First 10 simulated tags assigned to cars in SmartSpace Config TAG ASSOCIATION

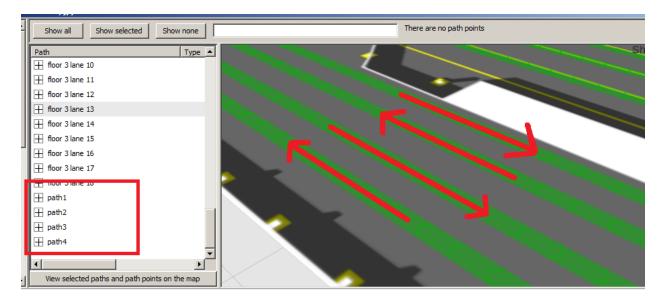
Tag Id.	Owner	Position Name	X	Y	Z
00:11:CE:00:00:00:11:9C	1244	Roof	7.344	0	3
00:11:CE:00:00:00:11:9D	6794	Roof	6	0	3
00:11:CE:00:00:01:00:01	10001	Origin	0	0	0
00:11:CE:00:00:01:00:02	10002	Origin	0	0	0
00:11:CE:00:00:01:00:03	10003	Origin	0	0	0
00:11:CE:00:00:01:00:04	10004	Origin	0	0	0
00:11:CE:00:00:01:00:05	10005	Origin	0	0	0
00:11:CE:00:00:01:00:06	10006	Origin	0	0	0
00:11:CE:00:00:01:00:07	10007	Origin	0	0	0
00:11:CE:00:00:01:00:08	10008	Origin	0	0	0
00:11:CE:00:00:01:00:09	10009	Origin	0	0	0
00:11:CE:00:00:01:00:10	10010	Origin	0	0	0
Tag Ranges		Tag Rang	es allowed for	Types	
Tag Range Name	From Tag Id.	To Tag Id. Type			Allowed T

Second 10 simulated tags assigned to cars in SmartSpace Config TAG ASSOCIATION

Create paths

Use **TYPES / OBJECTS** to create a *Path group*.

Use **PATHS** to create "path1", "path2", "path3" and "path4":



Creating paths for simulation in SmartSpace Config PATHS

Assign a path group

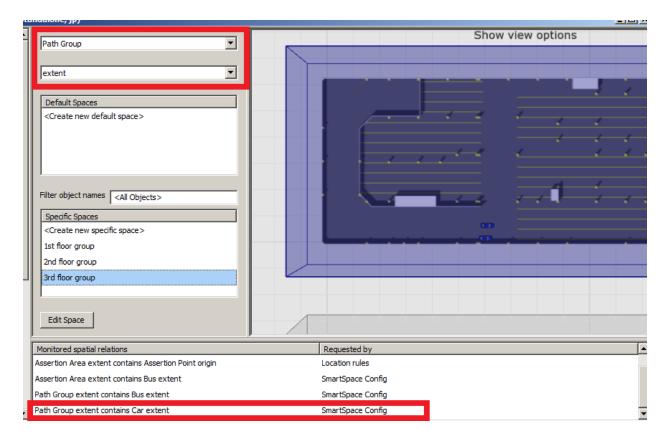
Assign the paths to a group in **SERVICE PARAMETERS**:

📮 Path tracking : Path : group		_ 🗆 🗡
		
floor 3 lane 17	3rd floor group	
floor 3 lane 18	3rd floor group	
path1	3rd floor group	
path2	3rd floor group	
path3	3rd floor group	
path4	3rd floor group	
J		_

Assigning path groups in SmartSpace Config SERVICE PARAMETERS

Monitor the spatial relation

In **SPATIAL PROPERTIES**, make the path group extent contain the paths and monitor the spatial relation:



Monitoring the spatial relation in SmartSpace Config SPATIAL PROPERTIES

Run the simulation

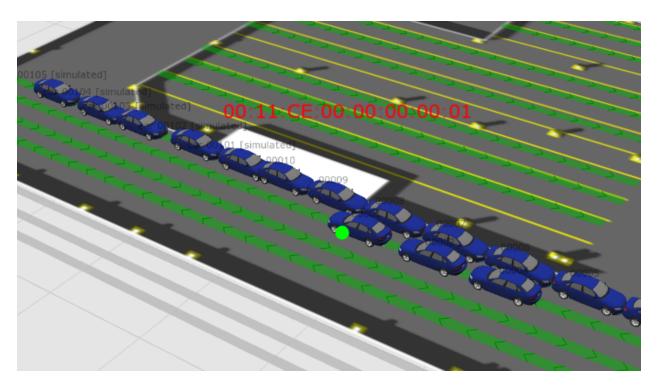
Assuming your cell is called "Location Cell 00001", run the following on a Windows command line:

type path_simulation.xml | ubisense_path_simulation_admin.exe start "Location Cell 00001"

Or this on a Unix-style command line:

ubisense_path_simulation_admin start "Location Cell 00001" < path_simulation.xml

Back in SmartSpace Config, turn on *Show foreground objects* in the PATHS map. Double-click an object to see its tag being simulated:



Simulated cars shown on the map in SmartSpace Config PATHS