

Version 6.2.0



Location Awareness Services for WebSphere Sensor Events



June 2009

This edition applies to Location Awareness Services for WebSphere Sensor Events version 6 release 2 modification 0. This edition applies to all subsequent releases and modifications until otherwise indicated in new editions.

A form for readers' comments appears at the back of this publication. If the form has been removed, address your comments to:

International Business Machines Corporation
Department 9BSA
P.O. Box 12195
Research Triangle Park, NC 27709-2195

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2006, 2009.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Location Awareness Services for WebSphere Sensor Events 1

Overview	1
What is Location Awareness Services for WebSphere Sensor Events?.	1
How the data is used	2
Roles and access	3
Defining Location Awareness Services for WebSphere Sensor Events	5
Administering	19
Defining areas and subareas.	19
Defining zones	31
Defining the topology	38
Planning for classes and items	45
Defining classes, items, and groups	46
Defining rules	52
Defining how to handle alerts	59
Setting system properties.	62
Formatting data types for importing data to Location Awareness Services for WebSphere Sensor Events	70
Importing resource data to Location Awareness Services for WebSphere Sensor Events	70
Configuring properties	71
Using the dispatcher	74
Backing up and restoring data	77

Scheduling deletion of tag data.	79
Tracking replay accounts	80
Operating	81
Starting the Spatial Management Client	81
Starting and stopping tag processing	87
Replaying tag movements and events	88
Handling alerts	88
Searching tags	90
Generating reports	92
Developing	93
Web services	93
Use cases.	121
SOA integration	121
Using containers	124
Evacuating locations	127
Troubleshooting	128
Logging	128
Handling alerts for Location Awareness Services for WebSphere Sensor Events event providers and receivers	130
General troubleshooting tips	130
Troubleshooting the Spatial Management Client Messages.	135
Glossary	141

Index 145

Location Awareness Services for WebSphere Sensor Events

After you have installed Location Awareness Services for WebSphere® Sensor Events, use these topics to access the component information:

Overview

This section provides an overview of Location Awareness Services for WebSphere Sensor Events components.

What is Location Awareness Services for WebSphere Sensor Events?

IBM® Location Awareness Services for WebSphere Sensor Events allows companies to continuously track active tags in real time in predefined areas of refineries, plants, and office buildings. Third-party asset location systems provide the tags, which may be carried by employees or visitors, or fixed to assets. Third-party systems also include reader infrastructure and a location engine, which is software that calculates tag positions based on the tag signals received by different readers. The Location Awareness Services for WebSphere Sensor Events solution works with these systems to visualize locations that are being monitored and to display the current position of personnel or assets carrying the tags.

Location Awareness Services for WebSphere Sensor Events provides a visual console that automatically locates the tags that are monitored in real time, allowing for real time response to emergency situations or security breaches. Personnel or assets can be monitored in virtual danger zones and Location Awareness Services for WebSphere Sensor Events will send safety and security breach alerts if assets or personnel are not qualified for entry or exit. Zones are virtual areas that can have rules or permissions assigned to them, and may vary over time. For example, temporary dangerous construction zones may be created.

The solution does the following:

- Cooperates with third-party position determination systems to acquire information about the current location of personnel or assets.
- Allows visualizing monitored areas and the current position of personnel and assets within these areas.
- Supports the rule-based specification of supervision policies.
- Supports a flexible alerting concept.
- Supports integration with existing human resource or asset management applications.
- Supports flexible reporting on monitored areas, providing the current or historical position of personnel and assets within these areas.
- Offers Web services that allow the whole solution or part of its functionality to be used in Service Oriented Architecture (SOA) applications.

An adapter (not shown in Figure 1 on page 2) gathers data from real time location systems (RTLS) monitoring the area and sends the data to Location Awareness Services for WebSphere Sensor Events, which stores information in the resources database and performs runtime processing, such as determining when zones are entered or exited, checking business rules, and calling registered programs in case

of business rule violations. Information in the resources database, such as position coordinates and zones, is displayed on the Spatial Management Client. The configuration table in this same database is used to define, update, and display administration and operation information through portlets in the WebSphere Application Server administrative console. You can also use the WebSphere Application Server administrative console to define business rules, notification programs, and various system properties.

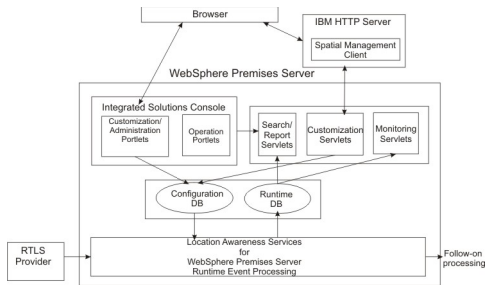


Figure 1. Overview of Location Awareness Services for WebSphere Premises Server

How the data is used

This section provides a brief overview of how the data provided to Location Awareness Services for WebSphere Sensor Events is used. Each item that is represented is described in greater detail in the sections that follow.

Data is used in Location Awareness Services for WebSphere Sensor Events as follows (see Figure 2 on page 3):

1. A third-party event provider sends provider-specific data.
Location Awareness Services for WebSphere Sensor Events regularly receives data from the event providers, which usually are connected to a number of event devices (receivers). The receivers regularly receive signals from active tags that are usually attached to assets or carried by people. The event providers consolidate the signals that are received and create fixed-format messages that are generated from the signals. Each of these messages contains details about a single tag, including tag ID and current position.
2. Location Awareness Services for WebSphere Sensor Events recognizes the event and transforms it to a provider independent format that is used throughout the internal processing of Location Awareness Services for WebSphere Sensor Events.
3. The event is linked to a zone. Because zones can overlap, a location event can affect multiple zones.
4. Rules are checked and, if necessary, an alert event is issued.
5. The event is sent to the subscribing programs and notification programs. See "Location Awareness Services for WebSphere Sensor Events events" on page 17 for more information.

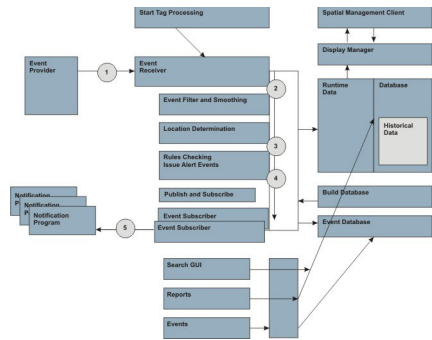


Figure 2. Data flow and usage

Roles and access

This topic lists the groups, allowed actions, and role associated with the Location Awareness Services for WebSphere Sensor Events portlets, servlets, and Web services.

Portlets

Table 1. Groups, actions, and roles for portlets

Portlet name	Default groups	Allowed action	Roles
Boundary Zones	lasadministergrp, laslocategrp	view and change settings	lasadminister, laslocate
Business Rules	lasadministergrp, lasconfiguregrp	view and change settings	lasadminister, lasconfigure
CEI Events	lasadministergrp, lasmonitorgrp	view and change settings	lasadminister, lasmonitor
Classes/Items Manager	lasmonitorgrp	view settings	lasmonitor
	lasadministergrp, lasregistrategrp	view and change settings	lasadminister, lasregistrategrp
Control Processing	lasoperategrp	view and change settings	lasoperate
Devices	lasadministergrp, laslocategrp	view and change settings	lasadminister, laslocate
Event Provider	lasadministergrp, laslocategrp	view and change settings	lasadminister, laslocate
Gate Manager	lasadministergrp, laslocategrp	view and change settings	lasadminister, laslocate
Groups Manager	lasadministergrp, lasregistrategrp	view and change settings	lasadminister, lasregistrategrp
Mail Host Configuration	lasadministergrp, lasconfiguregrp	view and change settings	lasadminister, lasconfigure
Mail Receiver Configuration	lasadministergrp, lasconfiguregrp	view and change settings	lasadminister, lasconfigure
Notification Channels	lasadministergrp, lasconfiguregrp	view and change settings	lasadminister, lasconfigure

Table 1. Groups, actions, and roles for portlets (continued)

Portlet name	Default groups	Allowed action	Roles
Notification Program Manager	lasadministergrp, lasconfiguregrp	view and change settings	lasadminister, lasconfigure
Registration Units	lasadministergrp, laslocategrp	view and change settings	lasadminister, laslocate
Replay Accounts Administration	lasoperategrp	view and change settings	lasoperate
Reports Administration	lasadministergrp, lasconfiguregrp	view and change settings	lasadminister, lasconfigure
Reports Operation	lasadministergrp, lasmonitorgrp, lasoperategrp	perform reports	lasadminister, lasmonitor, lasoperate
Search Tags	lasadministergrp, lasmonitorgrp	perform searches	lasadminister, lasmonitor
System Properties	lasadministergrp, lascustomizegrp	view and change settings	lasadminister, lascustomize

Servlets

Table 2. Groups, actions, and roles for servlets

Servlet name	Default groups	Allowed action	Roles
LasVisualizationEAR	lassmcadministergrp, lasmonitorgrp	can use the Spatial Management Client, but cannot change the configuration	allrole
	lassmcadministergrp, lasmonitorgrp	view zones and areas	getrole
	lassmcadministergrp	change zones and areas	putrole
AtlasReportingServletEAR	lassmcadministergrp, lasmonitorgrp, lasoperategrp, lasadministergrp	view reports	getrole
TagProcessingServlet	lasoperategrp	view and change settings	allrole
db2AssetMgmtEAR	lassmcadministergrp, lasmonitorgrp	view and change settings	allrole

Web services

Table 3. Groups, actions, and roles for Web services

Web service name	Default groups	Allowed action	Roles
LasQueryEAR	lassmcadministergrp, lasmonitorgrp	view tag details and reports	allrole
LasEventHandlingEAR	lassmcadministergrp	view and change settings	allrole

Table 3. Groups, actions, and roles for Web services (continued)

Web service name	Default groups	Allowed action	Roles
AtlasImportEAR	lassmcadministergrp	view and change settings	writerole
	lassmcadministergrp, lasmonitorgrp	view settings	viewrole
PremisesCEPRuleDefinitionEJB EAR	lassmcadministergrp	change settings	allrole

Defining Location Awareness Services for WebSphere Sensor Events

This section explains the structure of Location Awareness Services for WebSphere Sensor Events and how to define it.

Figure 3 shows the structure of Location Awareness Services for WebSphere Sensor Events and how the topological items are related.

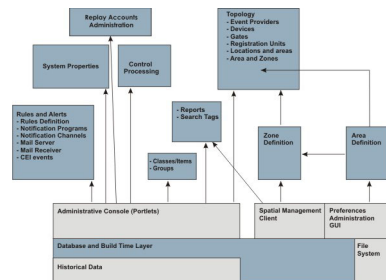


Figure 3. Structure of Location Awareness Services for WebSphere Sensor Events

The following user interfaces are provided with Location Awareness Services for WebSphere Sensor Events:

- Preferences Administration GUI

The Preferences Administration GUI allows you to define areas and subareas and set preferences for those areas, such as scaling and coordinate transformation values. You can also define the SVG images for your areas.

- Spatial Management Client

The Spatial Management Client allows you to define zones and provides a visualization of the defined topology with the tags and their current positions. You can see if a tag has violated a business rule. You can also search for items and request reports.

There are two versions of the Spatial Management Client: one that allows you to both define and delete zones and monitor tags, and one that only allows you to monitor tags.

- Location Awareness Services for WebSphere Sensor Events Administrative Console

The Location Awareness Services for WebSphere Sensor Events Administrative Console is based on WebSphere Application Server and consists of portlets that allow you to define the topology of a location where assets and personnel are tracked. You can use the portlets to define the items, such as assets and personnel, and the rules that control compliance with safety and security regulations. You can also set or change Location Awareness Services for

WebSphere Sensor Events system properties. Additional portlets are available to view, maintain, and search events and also to generate and view reports on tag activity.

To define Location Awareness Services for WebSphere Sensor Events, complete the following steps using the GUIs and Location Awareness Services for WebSphere Sensor Events Administrative Console:

- Define areas and subareas for the location. Do so by creating scalable vector graphic (SVG) images of the areas that you want to monitor and point to them in the Preferences Administration GUI. Because the Spatial Management Client is typically accessed through IBM HTTP Server, store the SVG images and item icons on the Location Awareness Services for WebSphere Sensor Events on which the IBM HTTP Server is located.
- Define zones in the administration version of the Spatial Management Client. Zones are related to the area in which they are created.
- Define the location topology in the Location Awareness Services for WebSphere Sensor Events Administrative Console:
 - Define the event providers and associated devices:
 1. Define event providers.
 2. Define devices for each event provider (if required).
 - Define the registration units you want to use to define items.
 - Define any boundary zones.
 - Define gates.
- Define the item class hierarchy of items to be tracked, such as Person and Asset classes in the Location Awareness Services for WebSphere Sensor Events Administrative Console.
- Define the item groups to be used and the relationship between the groups in the Location Awareness Services for WebSphere Sensor Events Administrative Console.
- Register the items to be tracked by the system and the association of items to classes and groups in the Location Awareness Services for WebSphere Sensor Events Administrative Console.
- Define the business rules that determine the item tracking behavior of the system in the Location Awareness Services for WebSphere Sensor Events Administrative Console. For example, you might define a rule that allows only those persons belonging to a specific group, such as Security, to enter a specific zone, such as HazardousZone. You can also set system properties to determine the alert behavior for other rules, such as a low battery threshold.
- Define the programs or web services to be called for filtered alerts in the Location Awareness Services for WebSphere Sensor Events Administrative Console. For a predefined e-mail notification program, define who to notify about alerts.
- Define the customized reports available for this installation.

Topology

This topic lists the location topology to be defined.

Define the location topology:

Event providers:

This topic explains third-party event providers that are mainly for location events.

Location Awareness Services for WebSphere Sensor Events relies on third-party *event providers* to provide tag position data. The third-party hardware and software must be installed and configured to work before configuring Location Awareness Services for WebSphere Sensor Events. In general, sufficient hardware must be available to calculate three-dimensional position data for each tag, and it must be installed and configured to provide the necessary positional data.

Event providers monitor areas and feed Location Awareness Services for WebSphere Sensor Events with tag location data. In turn, devices read the tag signals and send those to the event provider. Device receivers are always associated to an event provider (hub). Event providers are not part of Location Awareness Services for WebSphere Sensor Events, so they must be defined within Location Awareness Services for WebSphere Sensor Events. They must be configured for an existing area so that Location Awareness Services for WebSphere Sensor Events can track tags within that area. They are defined in the Location Awareness Services for WebSphere Sensor Events Administrative Console.

Event providers can be set up and calibrated so that they deliver absolute coordinates or coordinates can be transformed to fit the Spatial Management Client display. Coordinates can be transformed in the following ways:

- *Base point displacement*: Maps the first coordinate system to that of the second one.
- *Scaling*: Scales the coordinate systems if they use different scale units.
- *Rotation*: Makes axes of the coordinate systems congruent.
- *Permutation of axes*: Makes X and Y coordinates in both systems point in the same directions.
- *Smoothing algorithm*: Smooths position estimates.

Such transformation rules must be configured for each event provider.

Devices:

This topic explains devices and device groups and their importance within Location Awareness Services for WebSphere Sensor Events.

Devices can be either readers or a device group to which you can associate several readers. Devices represent physical or logical equipment from event providers that provide location data for tags. Devices can be readers that are simple devices or logical device groups (virtual groups) that are used to group the devices into logical units. For example, by grouping the devices into logical units, you can optimize location calculation.

Devices are always related to an event provider (hub). After defining an event provider, you can define the devices. After defining the devices, you can then define gates, registration units based on devices or device groups, or barrier zones with relation to devices.

Gates:

This topic explains gates and their role in controlling access within zones for Location Awareness Services for WebSphere Sensor Events.

Gates provide access control for the entry way and exit of a zone. With gates, you can associate one device that specifically monitors the entry to or exit from a zone. Gates are defined after defining the devices for the event provider (hub).

When monitoring zones in areas, you will need to define the gate twice, for the zone and for the area. Otherwise, Location Awareness Services for WebSphere Sensor Events cannot correctly monitor tag counts for the zone and area.

Registration units:

This topic defines registration units and explains their purpose.

Registration units are location event providers that you designate for the specific purpose of registering tag IDs with Location Awareness Services for WebSphere Sensor Events when you create items. For example, you can define a hub as a registration unit and then use it to read tags when defining items, which means you do not have to enter the tag IDs manually.

Locations and areas:

A *location* is made up of many *areas*, each of which represents a real physical space within the location to be monitored. *Subareas* are areas nested inside of other areas.

Areas are graphically represented and are the container for all zones. Areas have a flat lower and an optionally flat upper boundary.

Define an area by creating an SVG file of the area and then importing it into the Spatial Management Client by referencing it in the Preferences Administration GUI. See “Defining areas and subareas” on page 19.

Zones:

This topic defines and explains zones, including boundary zones, and describes how to monitor the entrances and exits of different classes of zones.

Zones are designated logical sections within areas that are associated with those areas and for which rules can be defined. Zones can overlay each other and are the units on which rules can be performed and on which counts and statistics for a tag entering or leaving can be calculated.

Zones within an area are defined with the Spatial Management Client and can be of different types. An entire area can also be considered a zone. You cannot change its size in the Spatial Management Client and it is not displayed as a colored region. However, rules can be attached to it.

Within the IBM Location Awareness Services for WebSphere Sensor Events system, zones are used for different purposes. Depending on their purpose, they are classified into one or several of the following zone classes:

Alarm zones

Alarm zones are the most common type of zone. Access restriction rules or similar rules can be triggered when an item (usually a person or asset) enters or exits a zone of this class. The restriction rules can be set for all other zone types as well, but they have additional semantics, as described in the following definitions.

Privacy zones

Currently privacy zones behave like alarm zones.

Shadow zones

Tags entering shadow zones might not be visible temporarily because they are out of reach of the tag reader infrastructure or the signals are shielded.

Location Awareness Services for WebSphere Sensor Events assumes that a tag continues to be in the shadow zone at the last reported position after it has been seen. No alert is generated if the tag is no longer visible.

Boundary zones

These zones are used for implementing access control to areas that are not covered by event devices and therefore cannot be controlled completely or directly. See “Monitoring the entrance and exit of zones that are not fully covered by devices” on page 11.

Note: In the Spatial Management Client, the term *barrier zone* is a synonym for a *boundary zone*.

Exit zones

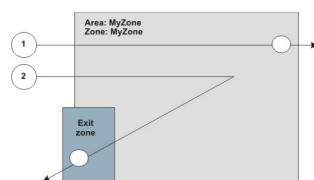
These zones are used to determine if a tag has passed and no signals can be received thereafter. They indicate that an item has left the premises and so there is no reason to be concerned about not receiving a signal.

Monitoring the entry and exit in an area

The following example describes how an item (person or asset) might be tracked when entering and exiting a zone. Assume you have X (0.0, 0, 100, 100, and 100) and Y (100 and 0) coordinates representing the area MyZone. Whenever a tag is visible within these coordinates and a signal comes from the related hub, Location Awareness Services for WebSphere Sensor Events registers the tag within the zone.

Consider the following scenarios:

1. The tag enters the area and follows the path indicated in the graphic below. If the tag is no longer visible, Location Awareness Services for WebSphere Sensor Events stores information about the last location where the tag was seen (indicated by the small circle at right edge of the area), and after a configurable time generates an event indicating that the tag is not responsive and was last seen at the stored location.
2. The tag follows the path in the graphic below and is last seen in an exit zone. Location Awareness Services for WebSphere Sensor Events no longer displays the tag in the area and recognizes that the tag has left the area.



So that Location Awareness Services for WebSphere Sensor Events knows that tags have left an area, you should define exit zones. Otherwise Location Awareness Services for WebSphere Sensor Events assumes that the tag is still at the edge of the area, but not responsive anymore.

Monitoring the entry and exit in an area (gates)

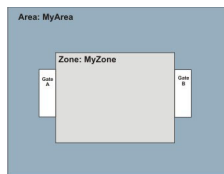
You can also define one device to be responsible for reporting the entry or exit to or from an area or zone. Do this by creating a *gate*. You define an event provider and then define a device to monitor the gate. You then specify whether the device monitors tags that are entering the zone (IN) or tags that are exiting the zone

(OUT). When the device sees a tag that fits the parameter you specified (IN or OUT), it reports the event and generate an alert if a rule is broken.

When monitoring zones in areas, define the gate for the zone and for the area. Otherwise, Location Awareness Services for WebSphere Sensor Events cannot correctly monitor tag counts for the zone and area.

Consider the following scenario for a zone inside of an area that is not fully monitored by devices:

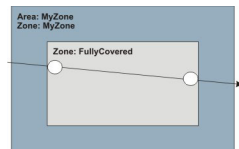
1. Create one gate to monitor tags with device A that enter the zone. Define the gate twice, once for the zone and once for the area.
2. Create another gate to monitor tags with device B that leave the zone. Also define this gate twice, once for the zone and once for the area.



Monitoring zones that are fully covered by devices

The following graphic depicts a zone in which devices can cover all of the areas. The following example describes how Location Awareness Services for WebSphere Sensor Events tracks a tag that follows the path indicated by the arrow.

1. When the tag reaches the first point (indicated by a circle), Location Awareness Services for WebSphere Sensor Events generates an event internally that indicates that the tag entered the zone and checks whether any existing rules apply to the situation. The tag count for the zone increases by 1.
2. Within the FullyCovered zone, Location Awareness Services for WebSphere Sensor Events can usually track the position of the tag continuously. If Location Awareness Services for WebSphere Sensor Events loses contact with the tag, an `AtlasTagNotResponsive` event is generated, indicating an abnormal condition. Within the zone, no location-dependent rules are checked.
3. When the tag leaves the zone (indicated by the second circle) Location Awareness Services for WebSphere Sensor Events generates an event indicating that the tag left the zone and checks whether any existing rules apply to this situation. The zone tag count decreases by 1 when the tag leaves the zone.



Thus, zones that are fully covered by devices allow you to fully track the activity of a tag. This type of zone is usually an alarm zone, where you define business rules for monitoring activity within the zone.

Monitoring shadow zones

The following scenarios describe situations in which zones are not completely covered by devices.

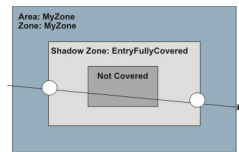
Assume you want to monitor a closed room in which the tags cannot be seen in all sections at all times. For example, the room might contain metal, which reflects signals in a certain section of the zone so that signals are too low to register, or a chimney is located above the devices in one section.

The following graphic depicts two scenarios:

1. The entry and exit for the zone are fully covered:

- Entry to the zone (indicated by the first circle) and exit from the zone (indicated by the second circle) are covered by devices. However, there are spots in the zone (indicated as "Not Covered") where signals from the tag cannot be received.
- Location Awareness Services for WebSphere Sensor Events assumes that a tag continues to be in the shadow zone at the last reported position after it has been seen. No alert is generated if the tag is no longer visible.

Note: This situation is different from the situation described in "Monitoring the entry and exit in an area" on page 9.



2. The entry into the zone is not fully covered:

In this special scenario, you must define a zone or zones outside of the entry area to monitor tags that enter or exit the zone. Solutions include the following:

- Two boundary zones
- Single boundary zone
- Mixed approach of zones

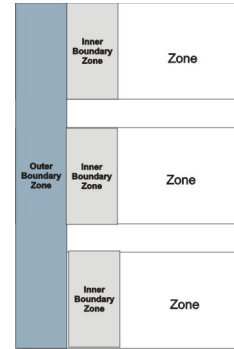
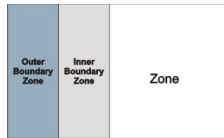
Monitoring the entrance and exit of zones that are not fully covered by devices

Some zones are not fully covered by event devices; however, a precise count of tags within a zone is still needed. To accomplish this, the entrance and exit of the zone must be monitored. To monitor these zones, you define *boundary zones* around or at the entrance of the zone to be monitored.

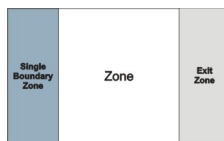
Boundary zones can be related to a target zone in the following ways:

- **Outer boundary zone:** The tag is assumed to be out of the target zone.
- **Inner boundary zone:** The tag is assumed to be in the target zone, even it is not visible.
- **Single boundary zone:** The tag is assumed to be in the target zone, even it is not visible. However, you do not use an outer boundary in this case.

As shown in the following figures, target zones can be monitored by one or more inner and outer boundary zones, or multiple target zones can share the same inner or outer boundary zones.

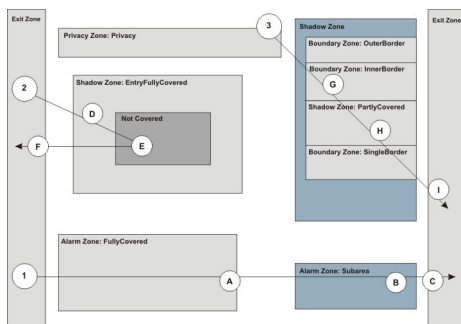


You can use a single boundary zone to monitor the entrance of the target zone and an exit zone to monitor the exit. If no exit zone is defined, Location Awareness Services for WebSphere Sensor Events assumes that the tag remains in the zone, even though it cannot see it.



Sample zone layout

The following figure combines different zones in one area. Notice that you can overlap zones. You can open an overlapping area to see different graphical representations, and for each zone you can see summary counts for the all child zones of the area in focus. However, note that you cannot see details for more than one zone at a time.



In the graphic, the arrows represent persons with tags walking through the area. The circles represent points along the path they take:

1. A person walks from the left exit zone to the right exit zone, passing through two zones:
 - When the person reaches point **A**, the tag is seen by the devices and Location Awareness Services for WebSphere Sensor Events knows that the tag is in the FullyCovered alarm zone.
 - When the person reaches point **B**, the tag is seen by the devices and Location Awareness Services for WebSphere Sensor Events knows that the tag is in the

Subarea alarm zone. Because Subarea is another zone, if you had imported and defined the image of the other area, you could navigate to it and see the tag there as well.

- When the person reaches point **C**, the tag is seen by the devices and Location Awareness Services for WebSphere Sensor Events knows that the tag is in the right exit zone. When the tag is no longer seen in the exit zone, Location Awareness Services for WebSphere Sensor Events removes the tag from the area and assumes that the tag has left the area.
2. A person walks from the left exit zone into the shadow zone and then exits through the left exit zone:
- When the person reaches point **D**, the tag is seen by the devices and Location Awareness Services for WebSphere Sensor Events knows that the tag is in the EntryFullyCovered shadow zone.
 - When the tag reaches point **E**, the tag is no longer seen. Because it is in a shadow zone, Location Awareness Services for WebSphere Sensor Events does not expect the tag to respond and does not generate an alert event. Location Awareness Services for WebSphere Sensor Events continues to assume that the tag is in the EntryFullycovered shadow zone.
 - When the tag reaches point **F**, the tag is seen by the devices and Location Awareness Services for WebSphere Sensor Events knows that the tag is in the left exit zone. When the tag is no longer seen in the exit zone, Location Awareness Services for WebSphere Sensor Events removes the tag from the area and assumes that the tag has left the area.
3. A person walks from a privacy zone within the area to the right exit zone, passing through several zones:
- When the person reaches point **G**, the tag is seen by the devices and Location Awareness Services for WebSphere Sensor Events knows that the tag is in the InnerBorder boundary zone. Because this zone is a border area for the PartlyCovered alarm zone, Location Awareness Services for WebSphere Sensor Events assumes that the tag is within this area.
 - When the person reaches point **H**, the tag might not be seen by a device, but Location Awareness Services for WebSphere Sensor Events assumes that the tag is in the area because it was last seen in the InnerBorder boundary zone. Also, because these zones overlap a shadow zone, the TagNotResponsive alert event is not issued.
 - When the person reaches point **I**, the tag is seen by the devices and Location Awareness Services for WebSphere Sensor Events knows that the tag is in the right exit zone. When the tag is no longer seen in the exit zone, Location Awareness Services for WebSphere Sensor Events removes the tag from the area and assumes that the tag has left the area.

Items

This topic explains items and their importance within Location Awareness Services for WebSphere Sensor Events.

Items represent the entities within a location that can be equipped with tags so that you can track their positions. Each item has attributes, including the tag ID, label, and icon link update interval. An item also has key properties that are required to set the item apart and properties that complete the description. Key properties and properties vary by class. A key property for a person might be a social security number and a property might be a person's first name.

When defining items for the first time, you can use a registration unit or external device to read the tag IDs into the system or you can enter them manually.

People and assets are the most common items that are monitored by Location Awareness Services for WebSphere Sensor Events; however, tags can also be attached to other items like product parts being consumed in discrete manufacturing processes like vehicle assembly. Therefore, Location Awareness Services for WebSphere Sensor Events uses the term *item* for everything that can be equipped with a tag.

Item classes

This topic describes item classes and subclasses and their importance within Location Awareness Services for WebSphere Sensor Events.

Item classes define items through a set of properties and attributes for them. For example, you might have the following classes: Person and Asset. Within these classes, you can also have subclasses with extended properties and attributes. For example, the Person class might have the subclass Administrators.

Items must belong to a class. Once an item is created and assigned to a class, you cannot move the item to another class. Because classes are in the form of a tree-structured hierarchy, an item cannot belong to more than one class directly. However, an item is automatically considered to be an instance of any superclasses of the given class. Items have the attributes defined for the class that they belong to. Class attributes are either defined directly for the class, or are inherited from its superclasses (if any).

Using the example of the Person class and the Administrator subclass, if a tag is assigned to class Administrator, it is also considered to be an instance of the superclass Person. Therefore the rule, "let me know when a Person enters the HAZARD zone," triggers an alert for Administrators as well as any other subclasses of "Person."

All classes have some common attributes, such as an icon label and tag ID. You can define required, or key, properties such as social security number or first name and last name, as well as optional properties such as telephone number. You can also define properties that are specific to your organization.

One specific property for all classes is the container attribute. If this is set, all items of the class are potential containers and can contain other items.

Subclasses inherit the properties of the parent (superclass), but you can also define new properties for them.

The Smoothing Algorithm attribute is also available for class-based smoothing. The Smoothing Algorithm is not inherited; changing it has no effect on subclasses or items belonging to those subclasses.

For more information on the properties and attributes available for item classes, see "Classes/Items Manager" on page 47.

Restrictions for classes:

- Properties for subclasses cannot have the same name as properties in parent classes. Also, once items or subclasses are defined for a class or subclass, you cannot change the class attributes.
- Within one class you cannot have two properties with the same name, regardless of whether the properties are key properties or not. The comparison of property

names is case-sensitive, though, so the property, `Name`, is not equal to the property, `name`.

Groups

This topic defines a group and how it functions within Location Awareness Services for WebSphere Sensor Events.

Groups are containers that allow grouping of items from different classes for common rules, searches, or so forth. For example, the Fire Brigade group can contain fire fighters (persons) and fire extinguishers (assets). Such containers are often referred to as *views* because you can view the items from a distinct perspective.

An item can belong to one or more groups; however, it does not have to belong to a group. A group can be a member of one or more groups; however, it does not have to be a member of another group.

You can specify that a group hierarchy be used by setting the `HierarchicalGroups` property in “System Properties” on page 62. The default value for `HierarchicalGroups` property is `Yes`, meaning group hierarchy is used.

Important: When group hierarchy is used (the default), this enforces a tree-like group hierarchy, which means that you cannot assign an item to more than one group and you cannot make a group a member of multiple groups. The characteristic of a group, when group hierarchy is used, is more similar to a class than a container.

You can select a group color in the Group Manager portlet if the `HierarchicalGroups` property is set to *true*. The color then displays behind all group member icons on the Spatial Management Client.

Rules

This topic explains different types of rules and how they trigger Location Awareness Services for WebSphere Sensor Events events.

Rules define conditions and policies that need to be met. For example, rules can be used to restrict access to certain zones or to limit the amount of time an item stays in a zone. Business rules are implemented based on a generic Complex Event Processing (CEP) engine which facilitates the development of additional rule types. Events (alerts) occur when rules are violated. Events are published and saved in the event database. Subscriber programs can subscribe to Location Awareness Services for WebSphere Sensor Events events. Violations of rules related to items can also be displayed in the Spatial Management Client.

Location Awareness Services for WebSphere Sensor Events rules typically refer to the aspects of the real world as it is modeled in Location Awareness Services for WebSphere Sensor Events - items and persons equipped with tags and the topology of the location to be monitored. Some basic rule types are supported by Location Awareness Services for WebSphere Sensor Events and you can use portlet-based user interfaces to create instances of these rule types. An example of a rule type is “must not enter” whereas an example of a related rule instance is “members of the Visitor class must not enter protected zones”.

When you define a rule, you can check for future events, but not for events that have already happened. This means that you cannot define an rule for conditions that have already occurred.

The following rules are related to zones and can be set and maintained in the Business Rules portlet in the WebSphere Application Server administrative console:

- Zone entry and exit rules
When a tag is considered to have entered or exited a zone is also affected by the `MaxUnrecognizedMovement` system property. See “System Properties” on page 62.
- Visitor escorting rules
These rules check whether a visitor is accompanied in specified zones by an escort. An alert is triggered if the visitor is away from the escort for a specified amount of time. The visitor must belong to a container class, the escort must not contain a container class, and the `DynamicContainerSupportOn` system property must be selected. See “System Properties” on page 62.
- Duration of stay rules
When an item is in a zone longer than specified by the rule, an alert is triggered for the tag.
- Maximum items per zone rules
When a defined threshold of items in a zone is exceeded, an alert is triggered.
- Rules for associating items
This rule type triggers an alert when a tag is near a base item, which must be a container item, for longer than a specified period of time.
- Rules for detecting when a tag is not moving
This type of rule is called the Man Down Detection and is typically associated with a tag on a person. When a tag does not move or blink for more than a specified time interval, an alert is triggered.

The following global rule types do not have different instances, but can be customized in the “System Properties” on page 62 portlet in the Location Awareness Services for WebSphere Sensor Events Administrative Console:

- Tag not responsive rule
When a tag is no longer detected by the event provider, an alert is triggered. See `MaxUnrecognizedMovement` and `TagNotResponsiveAlertAction` in “System Properties” on page 62 for more information. In addition, the tag icon fades on the Spatial Management Client.
- Tag battery low rule
When a tag has a low or empty battery, a `BatteryLowAlert` is issued. See `BatteryLowAlertAction` or `BatteryExhaustedAlertAction` in “System Properties” on page 62 for more information. In addition, the Spatial Management Client displays a small battery icon.
- Unknown tag rule
When a tag is detected that is not related to a defined item, an `UnknownTagAlert` is generated. See `UnknownTagAlertAction` in “System Properties” on page 62 for more information. In addition, the Spatial Management Client displays an unknown tag icon.
- Stationary tag rule
If a tag that belongs to a class that is defined as stationary moves, an alert is generated. The movement must exceed the value specified in the `MaxUnrecognizedMovement` system property.
To avoid a flooding condition of stationary alerts, if the tag moves twice the amount of units defined in `MaxToleratedMovement`, then a stationary alert will be generated once every 5 minutes.

Location Awareness Services for WebSphere Sensor Events events

This topic explains Location Awareness Services for WebSphere Sensor Events event details and notification programs to subscribe events.

The main purpose of monitoring items is to make sure that the position of a tagged item conforms to the awareness and security rules defined for the monitored locations. Nonconformance to such security or business rules triggers alerts that inform security staff or automated emergency systems about the event. In addition to the Spatial Management Client, other subscriber programs can also subscribe to Location Awareness Services for WebSphere Sensor Events events.

Event details

A subscriber must have sufficient information about the event to trigger corrective action or inform others sufficiently. Event information includes the following:

- Type of event:
 - LasBatteryLow - indicates that the tag battery is low on the tagged asset.
 - LasDurationOfStay - indicates that a tag has stayed longer in a zone than allowed.
 - LasDiagnosticEvent - a diagnostic message coming from an event provider.
 - LasItemsAssociation - indicates that two items referred to in the rule instance stayed close to each other for a predefined time and were associated.
 - LasManDownDetection - indicates that a tag (typically assigned to a person) did not move or blink for a predefined period of time.
 - LasMissingEscort - indicates that an item defined as "must be escorted" is missing the required proximity of an escort longer than allowed.
 - LasNotification - indicates that a notification was sent from a tag. Most providers have tags with programmable buttons on them. Pressing sequences of buttons can be evaluate in business rules, and can be translated into this type of event.
 - LasStationaryMoved - indicates that a tag that is defined as stationary has moved.
 - LasTagNotResponsive - indicates that no signal is being received from the tag.
 - LasTelemetry - indicates that a sensory value, such as a temperature, has exceeded or gone below a threshold.
 - LasUnknownTag - indicates an unknown tag is found.
 - LasZoneEntry - indicates that an unauthorized tag entered the zone.
 - LasZoneExit - indicates that an unauthorized tag exited the zone.
 - LasZoneThresholdExceeded - indicates that the maximum number of items, satisfying the filter criteria specified in the rule instance, that are allowed to be in a zone at a time was exceeded.
- Alert details are shown about the event dependent on event type. For example, they might include the following information:
 - Tag ID
 - Icon label (as tag identification)
 - Last valid time that the tag was reported
 - Last valid position where the tag was reported
 - Battery level
 - Zone or area exit time or entry time

- Groups of which the tag is a member (if a subscriber is interested only in specific groups)
- Class to which the tag is related (if a subscriber is interested only in a specific class)
- Specific message text that describes the situation
- Event history (status of event, time handled, and how it was handled)
- If the event was triggered independent of a specific tag, but is related to third-party infrastructure elements, the following information that is necessary to identify the failing element is provided:
 - Event time
 - Hub name

Depending on the situation and the information given by the event provider, more details might be in the specific message text.

Notification programs to subscribe events

The *event group*, or group of persistent related events, with its related messages queues is defined during installation and configuration. A filter is defined that identifies which Location Awareness Services for WebSphere Sensor Events alert messages are routed to these queues. As a result, an application can query the Common Event Infrastructure (CEI) event database, where all Location Awareness Services for WebSphere Sensor Events events are stored, for events or a *subscriber program* can subscribe to the topic related to the event group.

When installing Location Awareness Services for WebSphere Sensor Events, a predefined subscriber program listens to all events on the All events group. It dispatches the arriving events to the Location Awareness Services for WebSphere Sensor Events *notification programs*. The notification programs are the programs and web services that can be triggered when an event occurs. For example, a notification program might be an e-mail program that notifies authorized personnel of an event. By default, Location Awareness Services for WebSphere Sensor Events has only one event group defined: All events. However, you can add additional subscribers as a customization task.

Finally, you define *notification channels* for a given subscriber (defined as attributes for a channel definition) to specify the program that should be called for an event.

Customization tasks include the following:

- Implementing a new notification program and deploying it.
- Deploying a new program or web service that is called on entry of an event.

See “Defining how to handle alerts” on page 59 for details about these tasks.

To publish and subscribe, administrators can perform the following tasks (based on a set of event group topics) in the Location Awareness Services for WebSphere Sensor Events Administrative Console, specifically in “Notification Program Manager” on page 60 and “Notification Channels” on page 60:

- Define programs or services to be triggered.
- Define the channels triggering the program.

Administering

Perform administration tasks for Location Awareness Services for WebSphere Sensor Events using the Spatial Management Client and the Location Awareness Services for WebSphere Sensor Events Administrative Console.

Defining areas and subareas

Use the Preferences Administration GUI to define areas and subareas.

1. Import an SVG image of the area. See “Importing a graphic of your area.”
2. Use the “Preferences Administration GUI” on page 27 to reference the graphic and set preferences, such as scaling and coordinate transformations, for the area. See “Transforming coordinates for your areas” on page 21.
3. Optionally, use the “Preferences Administration GUI” on page 27 to nest another area inside of an existing area or create a subarea. Use the following fields in the GUI:
 - a. In **Parent SVG area name**, enter the name of the parent area. For example, Matrix.

Note: Area names must be unique across the installation.

 - b. In **X offset value**, enter the X offset value in units for placement of the subarea within the parent area. For example, if you want to nest the subarea 40 feet inside the X axis of the existing area, enter 40.
 - c. In **Y offset value**, enter the Y offset value in units for placement of the subarea within the parent area. For example, if you want the subarea 20 feet inside the Y axis of the existing area, enter 20.
4. Save your preferences and exit the GUI.
5. Open the Spatial Management Client and verify that your settings are correct. See “Starting the Spatial Management Client (administration)” on page 30.

Importing a graphic of your area

This topic describes how to import and convert a graphic of your area.

Before you begin

Make sure that you have installed Adobe® SVG viewer and Internet Explorer 6.0.

1. Convert the graphic to an SVG (Scalable Vector Graphics) format and copy the SVG file to the svg directory of the Spatial Management Client.

The directory is usually located in the *IHS_HOME*/htdocs/en_US/Tracking GUI/path.

Restriction: Using SVGs larger than 4 MB is not recommended.

2. Import the graphic of the area by referencing the SVG file in the **SVG path** field in the “Preferences Administration GUI” on page 27.

Make sure you scale the graphic correctly in the Preferences Administration GUI before creating zones in the area using the Spatial Management Client.

Converting a graphic to SVG format

You can import any graphic format supported by the SVG specification. All conformant SVG implementations must support PNG (Portable Network Graphics), JPEG (Joint Photographic Experts Group), and SVG images. The Adobe SVG viewer required by Location Awareness Services for WebSphere Sensor Events also supports GIF (Graphics Interchange Format) images. Graphic formats, such as

CAD, TIFF, or BMP, can be converted to one of the supported formats: PNG, JPEG, or SVG. Use a graphic editing tool such as CorelDraw, Adobe Photoshop, or Adobe Illustrator to perform these conversions.

Tip: Starting from a bitmap format results in lower quality. Vector formats result in higher quality.

Restriction: Using SVGs larger than 4 MB is not recommended.

Converting a vector format to SVG:

About this task

If the input format is a vector graphic, such as CAD, convert the graphic directly to SVG.

1. Import the graphic into a graphic editing tool, such as Adobe PhotoShop.
2. Prepare the graphic to be used in the Spatial Management Client by making the file as small as possible. The larger the graphic file, the more time it takes to load and render in the GUI. Do the following:
 - Remove excess layers of detail from the graphic before converting it. Layers of unwanted details such as plumbing, electrical, landscaping, and wall types, can be hidden under the layers of the basic area shape.
 - If possible, remove extra rooms or areas from your CAD drawing. For example, export only a room if you do not need the entire floor.
 - Do not embed fonts in the SVG file if given a choice; use system fonts instead. Embedding fonts significantly increases the file size.
 - Pre-scale the image so that the longest axis is no more than 600 pixels in length.
3. Make sure the upper-left corner of the graphic is the position you want to be 0, 0 on the X, Y coordinates in the Spatial Management Client. If it is not, crop the graphic until the positioning is correct.
4. Write down the graphic width and height values for later use.
5. Export the file as an SVG file. For example, floor1.svg.
6. Open the SVG file in a text editor, add the onload attribute, and modify the width and height, as necessary:

```
<svg onload="clearSvgArray(evt)" width="width" height="height" viewBox="0 0 width height">
```

Note: Make sure the graphic *width* and *height* values are those you wrote down while using the graphic editing tool. You can round the values to whole numbers if preferred.

For example:

```
<svg onload="clearSvgArray(evt)" width="586" height="452" viewBox="0 0 586 452">
```

Note: Because a PDF is vector-based, you can also convert PDF files to the SVG format. However, with the PDF format you cannot alter or delete unnecessary layers, and so the file size is larger.

Converting a bitmap format to SVG:

About this task

If the input format is a bitmap, you can convert the graphic to a PNG or JPEG format, which can then be linked to an SVG container graphic. Complete the following steps:

1. Import the graphic into a graphic editing tool, such as Adobe PhotoShop.

2. Prepare the graphic to be used in the Spatial Management Client by making the file as small as possible. The larger the graphic file, the more time it takes to load and render in the GUI. Do the following:
 - Remove excess layers of detail from the graphics before converting them. Layers of unwanted details such as plumbing, electrical, landscaping, and wall types, can be hidden under the layers of the basic area shape.
 - Pre-scale the image so that the longest axis is no more than 600 pixels in length.
 - Change the image bit depth of bitmap formats so that they are as small as possible. For example, in a simple line bitmap graphic, the graphic does not have to have a 24-bit depth when a 1-bit depth conveys the same image.
3. Make sure the upper-left corner of the graphic is the position you want to be 0, 0 on the X, Y coordinates in the Spatial Management Client. If it is not, crop the graphic until the positioning is correct.
4. Write down the graphic width and height values for later use.
5. Export the file as a PNG or JPEG format. For example, floor1.jpg.
6. Open an empty file in a text editor and create an SVG container file for the graphic by copying the following:

```
<svg onload="clearSvgArray(evt)" width="width" height="height" viewBox="0 0 width height">
  <g>
    <image height="height" width="width" xlink:href="file_name"></image>
  </g>
</svg>
```

Note: The graphic *width* and *height* values are those you wrote down while using the graphic editing tool.

For example:

```
<svg onload="clearSvgArray(evt)" width="586" height="452" viewBox="0 0 586 452">
  <g>
    <image height="452" width="586" xlink:href="floor1.jpg"></image>
  </g>
</svg>
```

7. Save the file as an SVG file. For example, floor1.svg.

Transforming coordinates for your areas

This topic explains how to transform coordinates so that an area displays properly on the Spatial Management Client.

Usually the map on the Spatial Management Client relates to the coordinate system of the event provider (hub) in such a way that the logical 0.0 point is either in the bottom left or top left corner (abstracting from the offset). In those cases it would be sufficient to define a scaling factor when defining the area and X and Y offsets. If your default 0.0 is in the top left corner and you want to change it to the bottom left, specify cartesian when defining the area.

The detailed information in this topic is necessary for more complex situations, where one hub is related to different maps with various orientation, overlap, and so on.

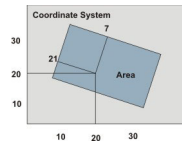
There can be up to three types of coordinate systems in an Location Awareness Services for WebSphere Sensor Events environment:

1. The systems defined by the event providers.
2. The systems defined by Location Awareness Services for WebSphere Sensor Events in the Spatial Management Client. The point of origin of the coordinate

system defined by the Spatial Management Client is the upper-left corner, with the Y-axis pointing downwards and the X-axis pointing to the right.

3. Logical reference systems.

The following figure shows a simple scenario that demonstrates why at least one coordinate transformation is required in almost all cases.

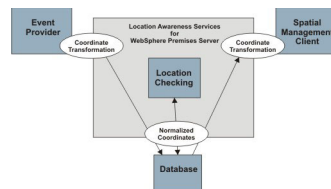


The point has coordinates 20, 20 in the event provider coordinate system. These coordinates must be translated to Spatial Management Client coordinates which are 7, 21 in this sample. Depending on the complexity of the setup, separate logical reference systems might be needed. In many cases, such as in the sample illustrated here, a system defined by the event provider or the Spatial Management Client can be used as a reference system. In some setups, the systems might even be identical.

When the systems are identical, no coordinate transformation is needed. If either system acts as a reference system, one coordinate transformation must be made. If there is a separate reference system, two transformations must be made: one between the event provider and the Location Awareness Services for WebSphere Sensor Events server and one between the Location Awareness Services for WebSphere Sensor Events server and the Spatial Management Client.

Processing coordinates

Location events from event providers are processed by Location Awareness Services for WebSphere Sensor Events and transformed as required:



1. Coordinates from an event provider are transformed before they are stored in the Location Awareness Services for WebSphere Sensor Events database.
2. Internal server side processing, such as location checking, is based on those normalized coordinates.
3. Location Awareness Services for WebSphere Sensor Events also transforms tag positions when sending them to the Spatial Management Client.

Configuring coordinate transformations

The event provider transformation rules are specified in the Location Awareness Services for WebSphere Sensor Events Administrative Console, specifically in the **Event Provider** portlet. In the **Coordinate Transformation** section of the Details view you can supply values for the following base transformation operations:

- **Horizontal Rotation:** Rotates the X-Y plane around the point of origin.

- **X-Y Permutation:** Permutates, or switches, the X and Y axis.
- **X Offset:** Displaces the area in the X direction.
- **Y Offset:** Displaces the area in the Y direction.
- **Scaling:** Scales the area to a larger or smaller size.

The transformations apply only to the X and Y coordinates. There is no need for three dimensional transformations because all components can be configured so that the Z-axis of their coordinate systems points upward.

The following samples show the effects that different values have.

Input coordinates are transformed based on the values of the listed parameters. Input coordinates (X,Y,Z) are converted to (X',Y',Z') according to the following rules:

- $X' =$
 - $\text{Scaling} * (X * \cos(\text{HorizontalRotation}) + Y * \sin(\text{HorizontalRotation})) + \text{XOffset}$ (if X and Y axis are *not* permuted)
 - $\text{Scaling} * (-X * \sin(\text{HorizontalRotation}) + Y * \cos(\text{HorizontalRotation})) + \text{YOffset}$ (if X and Y axis are permuted)
- $Y' =$
 - $\text{Scaling} * (-X * \sin(\text{HorizontalRotation}) + Y * \cos(\text{HorizontalRotation})) + \text{YOffset}$ (if X and Y axis are *not* permuted)
 - $\text{Scaling} * (X * \cos(\text{HorizontalRotation}) + Y * \sin(\text{HorizontalRotation})) + \text{XOffset}$ (if X and Y axis are permuted)
- $Z' = \text{Scaling} * Z$ ($\sin()$ and $\cos()$ are the standard trigonometric functions)

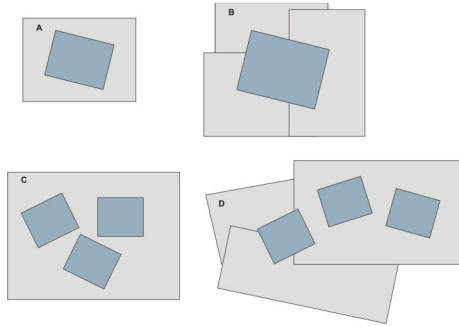
In the Spatial Management Client, you cannot configure horizontal rotation and axis permutation. However, you can specify the following base transformations when you define a new area in the Preferences Administration GUI:

- **X Offset value:** Displacement in X direction.
- **Y Offset value:** Displacement in Y direction.
- **Area scale:** Scaling factor.

As a consequence, areas must be aligned with either the event provider's coordinate system or the intermediate reference system.

Event provider and area configurations

The following figures show the different event provider and area configurations that can occur. The event provider-defined coordinate system is depicted in light gray and the coordinate system defined in the Spatial Management Client for the area is in blue.



Scenarios A and B depict a single event provider configuration and scenario C depicts a configuration where all event providers refer to the same coordinate system.

In scenarios C and D, the X and Y axes of all areas must be aligned. In other words, the X axis of each area must point in the same direction and the Y axis of each area must point in the same direction.

The following table summarizes the scenarios depicted above and shows which system should be used as a reference system:

Case	Event providers	Area	Reference system	Comment
A	1	1	Event provider or Spatial Management Client	When the Spatial Management Client is used frequently by multiple users, use the GUI's coordinate system as the reference. In all other cases, use the event provider's coordinate system to reduce the number of transformations.
B	None	1	Spatial Management Client	
C	1	None	Event provider	
D	None	Multiple	Separate reference system	The separate coordinate system can coincide with the coordinate system of the event provider, the Spatial Management Client, or both.

In scenarios A, B, and C, a separate reference system can also be used. However, doing so increases the number of required transformations.

Transformation samples

The following figures show some basic transformation scenarios. The original coordinate system is labeled with a "1", such as X-1, Y-1. The target system is labeled with a "2", such as X-2, Y-2. The scaling factor depends on the base units of both systems. The configuration settings are shown in the tables.

Table 4. Sample 1

Configuration setting	Value
Rotation	0
X-Y permutation	No
X-offset	dx
Y-offset	dy

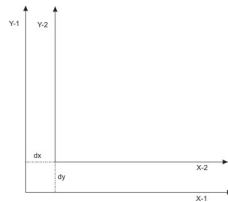


Table 5. Sample 2

Configuration setting	Value
Rotation	0
X-Y permutation	Yes
X-offset	dy
Y-offset	dx

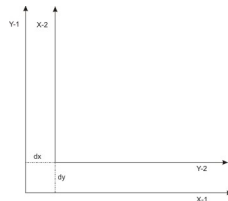


Table 6. Sample 3

Configuration setting	Value
Rotation	90
X-Y permutation	No
X-offset	dx
Y-offset	dy

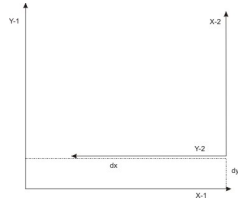


Table 7. Sample 4

Configuration setting	Value
Rotation	90
X-Y permutation	Yes
X-offset	dy
Y-offset	dx

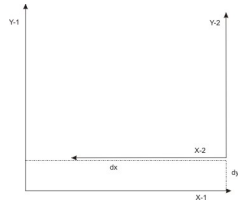


Table 8. Sample 5

Configuration setting	Value
Rotation	180
X-Y permutation	No
X-offset	dx
Y-offset	dy

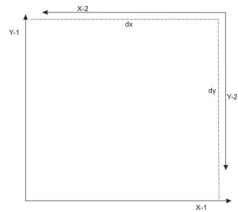


Table 9. Sample 6

Configuration setting	Value
Rotation	180
X-Y permutation	Yes
X-offset	dy
Y-offset	dx

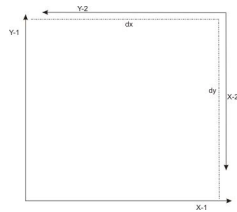


Table 10. Sample 7

Configuration setting	Value
Rotation	270
X-Y permutation	No
X-offset	dx
Y-offset	dy

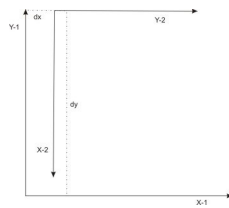
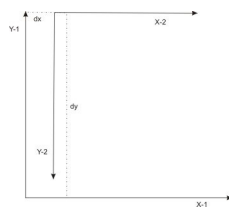


Table 11. Sample 8

Configuration setting	Value
Rotation	270
X-Y permutation	Yes
X-offset	dy
Y-offset	dx



The transformation shown in the last sample occurs frequently in Location Awareness Services for WebSphere Sensor Events configurations. The target coordinate system is like that defined by the Spatial Management Client, with the point of origin in the *upper-left* corner, with the Y axis pointing downward, and the X axis pointing to the right.

Preferences Administration GUI

This topic describes how to use the Preferences Administration GUI to define your Spatial Management Client and build time preferences for Location Awareness Services for WebSphere Sensor Events.

It is necessary to define your preferences only once per server installation instance for the installation entries.

Open the Preferences Administration GUI by opening the following URL:
`http://fully_qualified_host_name/Tracking GUI/AtlasPrefsAdmin.html`

Important: After you create a new area using the Preferences Administration GUI, you must close all browser windows, wait about 60 seconds, and then reopen the browser before trying to use the new area in the Spatial Management Client.

Spatial Management Client

The Spatial Management Client is a monitoring application that polls every n seconds for new data in a defined area, as specified in the `prefsV3.xml` file. The default value is to poll every second.

Note: Some browser functions are not supported. For example, using the **Back**, **Forward**, and **Refresh** buttons in the browser can lead to inconsistent displays of areas, tags, and menu options.

The `prefsV3.xml` file automatically updates some the **Build time** fields. When you update your product version, ensure that you merge the contents of the old `prefsV3.xml` file, rather than simply replacing the previous contents. The list must be consistent with the related tables in the Location Awareness Services for WebSphere Sensor Events database.

Define your preferences in the Preferences Administration GUI by entering information in the following **Build time** fields:

Note: Make sure you define your preferences and set your area, icon, and overview scale values correctly before you create zones.

Build time

Build time preferences are defined only when a new area or subarea is added. These preferences determine how the areas, zones, and resources display in the Spatial Management Client.

- **Area SVG:** Select an area from the menu or create or delete one.
 - Click **New** to create a new area.
 - Click **Cancel** to cancel your action without saving.
 - Click **Delete** to delete an area.

Notes:

- Defining an area with an SVG that is greater than 4 MB is not recommended.
- Deleting an area also deletes its subareas.
- **Area name:** Enter a descriptive name for the area. The area name for a given area cannot be changed.
- **SVG path:** Enter the relative path to the area scalable vector graphic (SVG) file. For example, `./svg/Matrix.svg`.
- **SVG overview path:** Enter the relative path to the area SVG overview file, which is the graphic file used for the overview window in the Spatial Management Client. For example, `./svg/Matrix.svg`.
- **SVG width:** Enter the width of the SVG file in pixels.
- **SVG height:** Enter the height of the SVG file in pixels.

- **Minimum Z:** Enter the minimum Z value, or height, for this area.
- **Maximum Z:** Enter the maximum Z value, or height, for this area.
- **Enter the width the drawing represents:** Enter the value in units that the drawing represents. For example, if the drawing represents an area that is 40 feet wide, enter 40.

Note: If you enter the drawing width, **Area Scale** is automatically filled in with a scale determined by the drawing width.

- **Area scale:** Enter the scale factor to use to scale the coordinate display in feet for the SVG file representation. As you move the cursor over the drawing, X and Y coordinates are visible. Move it over an area of the graphic that you know the coordinates for and adjust this value so the values match the scale of the drawing.

Note: If you enter the area scale value, the drawing width also adjusts. The current width and height are calculated and displayed at the current scale.

- **Overview scale:** Scale the overview window to the size you want it.
- **Parent SVG area name:** If this area is to be used as a subarea, enter the name of the parent area. Otherwise leave this entry blank.

Note: Area names must be unique across the Location Awareness Services for WebSphere Sensor Events installation.

- **X offset value:** Enter the X offset value in units for placement of the subarea within the parent area. Otherwise enter 0.
- **Y offset value:** Enter the Y offset value in units for placement of the subarea within the parent area. Otherwise enter 0.
- **Cartesian coordinate system:** Check this box to turn on the Cartesian coordinate system, which flips the area coordinate system so that the X0 and Y0 coordinates are located in the lower-left corner of the drawing with X positive going right and Y positive going up. This system matches the X0 and Y0 coordinate system of many third-party location event providers.

Installation

The **Installation** parameters on the Preferences Administration GUI are described in more detail in the topic about installing the Spatial Management Client.

The **Host** value is taken from the requesting URL for the Preferences Administration GUI and the **Port** value is set in the `prefsV3.xml` file.

Refresh and clustering options

- **Poll interval:** Enter a value to indicate the rate in milliseconds that tag data is requested from the server.

Note: Changing this value does not affect the frequency at which a tracked item's position is reported to the system. It only affects the frequency with which the GUI is updated.

- **Number of clustering grid rows and columns:** Define the number of grids to use for clustering tags. If you have many tags on the screen, overlapping tags can occur. If clustering is set to a value greater than 0, overlapping tags are shown as a cluster icon. This cluster icon can be clicked to open a window showing all tags in the cluster. Ten of those tags can be individually selected.

The number of grid rows defines how large the grid will be, which is covered by a cluster. For example, a value of 20 means a grid of 20 rows and columns, where all tags in one cell are shown as part of the cluster. A value of 0 turns off clustering, meaning you cannot influence the order of tags from back to front or select individual tags.

- Click **Save Installation Changes** to save your changes to the preferences. These preference settings will apply each time the user logs in to the Spatial Management Client.

Making changes

If you edit an attribute related to an area, the **Save Modified Area** button appears. If you press this button after filling in all of the changed attributes, you have the option to update the SVG file for the area.

Tip: If you change the size of the SVG, scaling, offsets, or the cartesian attributes, your old zone definitions, definitions made with the administrative interface for the event provider, or devices might be affected. Be sure to plan these values carefully and redefine all other affected definitions.

Starting the Spatial Management Client (administration)

This topic provides steps for starting the Spatial Management Client if you are an administrator.

About this task

Complete the following steps to start the Spatial Management Client:

1. Start the Spatial Management Client by typing the following URL in a browser: `http://fully_qualified_host_name/Tracking GUI/AtlasAdmin.html`. If you are not an administrator, see “Starting the Spatial Management Client” on page 81.
2. Enter your user name, and password if security is enabled, and click **OK**. Your individual preferences are displayed. You can save your preferences for each area you view by clicking **Save** under **DEFAULT VIEW**. Setting preferences prevents rescaling and repositioning each time you view an area of interest.
3. In **AREA**, select the area that you want to monitor from the drop-down list.
4. In **TAGS**, select the class of tags that you want to monitor. For example, select **Asset** to view all hardware that has been defined or **All** to view all tags.
5. In **ZONES**, under **Visible**, select the category of zones that you want to view.
6. In **ALERTS**, turn the alert sound on or off and choose whether to hide or view all alerts. You can also click **Acknowledge All Alerts** to acknowledge and turn off all current alerts.
7. In **DEFAULT VIEW**, click **Save** to save the current pan and zoom settings. You can customize the view and scale of the area without having to repeat the process every time you log in to the Spatial Management Client.

The **OVERVIEW** window provides a view of the entire area. Drag the blue box around the overview window and notice that the main graphic window of the Spatial Management Client reflects the highlighted area. The box size is controlled by the zoom slider and zoom box controls below. The upper-left corner of the blue box and the upper-left corner of the main graphic window are the same point.

See “Spatial Management Client (administration)” on page 31 for more information.

8. To start monitoring tags in the GUI, start the tag processing servlet:

- a. Open the WebSphere Application Server administrative console and click **Control Processing**.
- b. Select the WebSphere Application Server that is related to your installation and click **Start Selected**.

If you do not start the tag processing servlet, tags are displayed at their last reported location.

Results

In the Spatial Management Client, the defined tags are displayed with the icons you define, either for the item or the class. These icons move on the Spatial Management Client according to the reported coordinates. If you turn alerts on, a red circle highlights the tag icon when an alert related to the tag is reported. You can click the icon and see the alert details and acknowledge the alert. The circle goes away when you acknowledge the alert.

In some cases the tags fade, which means that there is no current position information available about the tag. Location Awareness Services for WebSphere Sensor Events assumes that the tag remains at the last reported position. Use the `InactivityDelay` system property to set the length of time after which a tag starts to fade. To avoid moving tags away from the last reported position, set this parameter to a high value. See “System Properties” on page 62 for a complete list of system properties.

Defining zones

This topic describes how to define zones for Location Awareness Services for WebSphere Sensor Events.

Use the “Spatial Management Client (administration)” to define zones for Location Awareness Services for WebSphere Sensor Events.

Spatial Management Client (administration)

This topic describes the administration version of the Spatial Management Client.

The Spatial Management Client provides a state of the art visual interface which shows the location of tags in real time, allowing an authorized user to monitor employees, contractors, and visitors in hazardous areas, to respond immediately to emergencies, and to locate high-value assets. With the administrative version of the GUI, you can also create or delete zones for special monitoring.

Notes: For optimal GUI performance:

- Use only Internet Explorer 6.0 with the Adobe Scalable Vector Graphics (SVG) Viewer for your browser.
- Maximize the Spatial Management Client for the best results.
- Restart the GUI whenever you change the screen resolution.
- Do not use browser functions. For example, using the **Back**, **Forward**, and **Refresh** buttons in the browser can lead to inconsistent displays of areas, tags, and menu options.

For keyboard accessibility of the GUI, edit the Mouse settings.

1. Select **Start** → **Settings** → **Control Panel** → **Accessibility Options**.
2. On the Mouse tab, select **Use MouseKeys**. This option allows you to control the pointer with the numeric keypad on your keyboard.

The Spatial Management Client retrieves all tags for an area in the following cases:

- When an area is opened
- When the class filter is changed
- Every n polling intervals. The value of n is set according to the `<DisplayRefreshCounter>` parameter in the `prefsV3.xml` file. If this parameter is not present in the `prefsV3.xml` file or it is set to 0, then a full redraw is not scheduled on the Spatial Management Client.

In all other cases, tags are only refreshed when they change their position or they change their alert state.

If you experience problems with the Spatial Management Client, refer to the troubleshooting tips in the product documentation for possible solutions.

- **AREA**

Select the area that you want to monitor from the drop-down list.

- **TAGS**

Select the class of tags that you want to monitor. For example, select **Asset** to view all hardware that has been defined.

- **ZONES**

Visible: Select the category of zones to view.

- **ALERTS**

Sound: Turn the alert sound **On** or **Off**.

Hide: To view all alerts, select **No**. To hide all alerts, select **Yes**.

Tag filter: Filter the tags displayed. The options are **2d/2d**, **p-data**, **inactive**, **alerts only**, and **all**.

Note: These tag filters do not affect the zone or area tag count. They only affect what you can see on the map. For example, if there are three tags in zone Z and one of them has no accurate location information (it has only proximity data) and you filter the tags by p-data, only one tag remains visible on the map, but the tag count for zone Z still shows 3. When you set a filter, an error message appears to remind you of this discrepancy in the tag count.

- **DEFAULT VIEW**

Click **Save** to save the current scaling, positioning, and menu settings to your user preferences. You can customize the view and scale of the drawing without having to repeat the process every time you start the Spatial Management Client and log in with your user ID.

When you click **Save**, the values for the currently selected area are saved. You can press this button for each area. Then, when you switch to an area that has a saved value, the saved setting information is used. The area where you last pressed **Save** is the area that is shown first when you access the interface.

You can also save selected tag labels using the **Save** function. These saved tag labels do not have to be area-specific.

- **Draw Trajectory**

Click **Draw Trajectory** to enable the display of a tag's trajectory for certain time period.

- A start time and an end time are required.
- Fill in the value for either the TagId or IconLabel. At least one of these values must be filled. If both are set, then the TagId is used. If both are set, the tagid is used).

- **Number of Points** - This value depends on the load of your system and network because the number of hops can be limited. The recommended default value is 2000 tag hops, but you should be sure to configure this value to accurately apply to the quantity of your data. If there are more hops in the selected area in the timeframe given, only every *n*th point is displayed, so that it fits in the specified number. For example, if you have 9000 entries in your database and specify the **Number of Points** value as 2000, then you would see every fifth hop in the tag. This means that you could lose information since the display is truncated to every fifth hop, instead of more frequently, such as every second hop. You will be informed of this truncation of information.
- **Timestamp-Interval** - the value you specify produces a timestamp at the position of the tag every *n* points. For example, if you draw 100 points and you specify 10 for the timestamp interval, at every 10th point a timestamp will be written, equalling a total of 10 timestamps.
- **OVERVIEW**

This window provides a view of the entire area. Click and then drag the highlighted area in the overview window and notice that the main graphic window of the Spatial Management Client reflects the highlighted area. The box size is controlled by the zoom slider and zoom box controls below. The upper-left corner of the box and the upper-left corner of the main graphic window are the same point.
- **Cluster view**

When several tags are close together and cannot be distinguished from each other, the cluster icon displays to indicate that there are several icons overlaying each other. Icons might overlay because the icons are too large, the current zoom is not close enough, or the tags are reported to have the same coordinates. To correct overlaying tags, try one of the following:

 - Downscale the size of the icons until they do not overlay each other.

To configure the size of the icons that display in the cluster view of the main window:

 - Press the Ctrl key plus the space bar to display the Tag Zoom Control window. Then click **Up** to enlarge the icons or **Down** to shrink the icons. Icons resize immediately.
 - To close the dialog window, close the window or press the Ctrl key plus the space bar again. You can save the configured icon size with your user preferences.
 - Zoom closer to the icons until you can distinguish them from each other.
 - Click the cluster icon to get a list of icons within the cluster. A window opens to display all the icons of the cluster and the information for each tag according to the current configuration (for example, labels, X and Y coordinates, and alerts). To see more information about a tag, click the appropriate icon and the information appears in the detail view while the cluster view window closes.
- **Zoom selection rectangle**

Click on the dotted rectangle (zoom selection rectangle) and move the pointer to the main graphic window where you can click and drag to create a zoom selection rectangle. When you release the mouse, the window zooms into the selected area.
- **Zoom slider**

Use the slider to enlarge or shrink the current image in the main window. You can drag the slider button, click on the hashed lines, or click the magnifying glass icon to change the zoom.

Note: When you have highly magnified an area, the blue box in the overview window might not be able to represent the area and it becomes a small black rectangle and no longer zooms. You can still drag the box to pan another area.

- **Current[®] tag count by zone/area**

The count table is a draggable window (click and press Shift to drag) that provides a list of areas, subareas, and zones and the number of tags currently in them. Only those zones that match the type of zones set to visible in the **ZONES** drop down menu are displayed.

- Click the area or zone name to display a current tag count window that lists the number of tags in the area and zone. All subareas and zones are listed under the area with which they are associated.
- When the tag count window is open, you also see a button for **Automatic Refresh On/Off**. By default the automatic refresh is on. The poll interval parameter is used for this refresh cycle. The tag movements in this window are independent of the ones in the main window, so there can be differences between how the tag movements are displayed.
- Click **Hide** to hide the area or zone or **Show** to display the area or zone on the main window.

Note: Only content filters, such as filtering for all the tags in the Person class, affect the zone count. Technical filters for details about the tags (2d/3d, p-data, and so on) apply to the visibility of the tags on the map, but they do not affect the zone counts.

- **Area List View**

Click this button to open a new window that displays the zones within the selected area and the number of tags within the zones. Click a zone to expand details about the tags within the zone. You can open multiple area list views at one time.

The tags shown within the zones will be filtered based on any search criteria you specify in the main view and you can click the pause button in the area list view to pause and view the tag information at a specific instance. You can also open the area list view when you are replaying data.

If you want to view an area list view for another area, you must open another instance of the Spatial Management Client.

Note: Filtering for a single tag does not apply to the area list view.

- **Search**

Click this button to search by class, group, or tag properties, or a combination of them.

Note: The search is *not* case sensitive. Also, the **AND** and **OR** only apply to the filter attributes within the class, group, and tag criteria. When you search by a combination of class, group, and tag criteria, they are always combined by **AND**.

- **Class Properties**

Select a class or classes to search for. Enter your search criteria. Click **AND** if all the search criteria you enter must be found or click **OR** to display search results for any search criteria that you enter.

- **Group Properties**

Select the group to search for. Click **AND** if all the search criteria you enter must be found or click **OR** to display search results for any search criteria that you enter.

- **Tag Properties**

To search for a specific tag, click **Tag** and enter the search criteria. Click **AND** if all the search criteria you enter must be found or click **OR** to display search results for any search criteria that you enter.

Click **Search** to search for the specified criteria or click **Reset** to clear all entries and perform a new search. Search results are displayed in a table or list format. When you select a tag in the table or list, the tag will be highlighted by a circle in the Spatial Management Client. If the tag is located in a different area, the area will open in the Spatial Management Client. Click **Save** to save the results to a file or close the window to exit without saving.

- **Replay**

Click this button to replay tag movements and events that occurred during a specific time frame.

A window displays. Enter the start and end date and time for the period of time you want to replay and click **Enable Replay Console**.

Select the area for which you want to display tag movements and events. Then click **Play** in the replay dialog to the right of the main window to watch the tag movements and events that occurred in the area during the specified time frame. Click **Pause** to pause events and **Resume** to resume playing them. Click **Exit** to close the replay dialog and to return to the current area and time.

Note: Times are relative to times on the database server. The machines that host the database server and WebSphere Application Server must be set to the same time zone.

Up to five users can use the replay function at any given time. You can track replay accounts using the Replay Accounts Administration portlet in the WebSphere Application Server administrative console.

Note: When you are using the replay function, you should see the tag count window and the tags in the area; however, for performance optimization reasons, there may be times that the number of tags visible on the screen and the number in the tag count window do not match.

- For the tags visible on the screen, only the tags seen by the location event provider after replay starts are drawn.
- For the tag count, the number of tags in a specific area or zone are counted. This count also includes tags that have entered the zone before replay starts but are not responsive after replay starts.

If you see this inconsistency in the number of tags, and you need to see the complete list of tags in an area at a specific point in time, use the **Search** or the **Show all tags** options.

- **Group Color On/Off**

Click this button to turn group color on or off. The color associated with the group in the Groups Manager portlet is seen as a colored rectangle behind the tag icon. Group color is off by default.

- **Acknowledge All Alerts**

Click this button to acknowledge and turn off all current alerts.

- **Reporting**

Click this button to see a list of defined reports that have been administered in the Reports Administration portlet. Select the **Display** link beside the report in the resulting list that you would like to view. Each report has a set of filter criteria. Click **Reload** to regenerate the list of reports.

See the Reports Operation documentation for more information.

- **Show all tags**

This option lists all tags that are currently in the area in a table similar to the **Search** results window. Selected filters for the area and tags do not apply.

Tags

For tags displayed on the Spatial Management Client, use the following features:

- **Tag Details:** Click a tag to display details about the tag including its tag ID, coordinates, and the class it belongs to. If there is an alert associated with the tag, you can acknowledge it by clicking **Acknowledge Alert**.
- **Label:** Hold down the Ctrl key and click a tag to display the Label window. Select the information to be displayed for the tag when you hover over it. For example, select **Label** to display the label text defined for the item, select **Tag ID** to display the tag ID, or select **X**, **Y**, or **Z** to display location coordinates for the tag.
- **Select Commands:** Hold down the Alt key and click a tag to display the Select Commands window. You can select from the following commands:
 - **Delete Tag** - Removes the tag from the area, leaving the item definition untouched. This action is relevant in scenarios where tags leaving the area cannot be monitored at all times by gates or exit zones. For example, if a tag has left the area, but this has not been recognized by the event provider, you can manually clean up the area by selecting the tag and deleting it.
 - **Show this Tag only** - Filters to show only the selected tag. This is a special tag filter and cannot be saved as a preference. Changing the tag filter or changing the area will take the tag filter away.
 - **Draw Tag's Trajectory** - Starting at the point you select this, a line for the current path of the selected tag is drawn until you select **Stop Drawing Tag's Trajectory**. If you change the area and then come back to the view for the selected tag, you should still see the trajectory line. This action is only possible for one tag. When you select another tag for trajectory, the line for the previously selected tag is removed.
 - **Stop Drawing Tag's Trajectory** - Stops drawing the line for trajectory, if **Draw Tag's Trajectory** was selected. Otherwise, choosing this command has no effect on the tag.

Zones

Note: In the Spatial Management Client, the term *barrier zone* is a synonym for a *boundary zone*, which is defined in the **Boundary Zones** portlet.

For zones displayed on the Spatial Management Client, use the following features:

- **Zone details:** Click a zone to display details about the zone including name, function, coordinates, and number of tags in the zone.

This feature also allows following actions for a zone:

 - **Hide zone:** If you select this, the zone is hidden (but tags are still displayed). To show the zone again, use the **Current tag count by zone/area** window.

- **Creating a zone:** Create a zone in the Spatial Management Client by holding down the Shift key and clicking on the main window to enter coordinates for the zone. The first click is represented by a small green dot; the second and subsequent clicks connect lines that outline the area. After all lines are selected, the area of the zone is automatically shaded. Release the Shift key and click once more to finish creating the zone.

Note: When you are pressing the Shift key, you can also click on tag icons to select the coordinate as a point in the zone.

In the Zone Creation window, enter the following information about the zone:

- **Choose zone type:** Select the type of zone you are creating.
- **Name:** Enter a unique name for the zone.

Note: If you enter the name of an existing zone, you overwrite the existing zone. Make sure your name is unique.

- **Min Z:** Enter the minimum height of the zone.
 - **Max Z:** Enter the maximum height of the zone.
 - **Modifying a zone:** You can modify an existing zone in the Spatial Management Client by holding down the Shift key and clicking on the main window to enter new coordinates for the zone. The first click is represented by a small green dot and the second and subsequent clicks will be connected by lines and the area of the zone is shaded. Release the **Shift** key and click once more to finish creating the zone.
- In the Zone Creation window, enter the following information about the zone:
- **Choose zone type:** Select the type of zone you are creating.
 - **Name:** Enter the name of the existing zone you are modifying.
 - **Min Z:** Enter the minimum height of the zone.
 - **Max Z:** Enter the maximum height of the zone.
 - **Delete a zone:** Hold down the Alt key and click a zone to delete it.

Restrictions on new areas and zones

This topic lists restrictions that exist for new areas and zones.

When you define a new area or zone, existing tags and tag to zone relationships will not be displayed or logged for the new area or zone. For example, you will need to recreate boundary zones and gates and review any historical data you are interested in. New alerts and tag data will be read and analyzed; however, old events will not be reanalyzed to determine whether a tag is within a newly defined area or zone.

For example, consider the following scenarios:

- If three-dimensional tag coordinates are read from the tag, the tag will be displayed and counted in the new area or zone.
- If only one device reads the tag, indicating its presence in the zone, the tag will not be shown or counted in the new area or zone.
- If no information is received from the tag but it was last seen in a location that would be inside the new area or zone, the tag will not be shown or counted in the area or zone.
- If no information is received from the tag, but it had passed a gate or barrier zone previously that you have now defined for the new area or zone, then the tag will not be shown or counted in the new zone.

To summarize, old location events that were received by devices that are now in a newly defined area or zone are ignored by the new area or zone. Only new events will be analyzed and displayed for the new area or zone. This also applies to zone-related rules. Since rules evaluate zone entry and exit event, they are not triggered until a tag enters or leaves a zone.

Furthermore, area and zone names must be unique across the Location Awareness Services for WebSphere Sensor Events installation.

Defining the topology

This topic lists the portlets you can use to define the Location Awareness Services for WebSphere Sensor Events topology (event providers, devices, gates, registration units, and boundary zones).

Event providers provide Location Awareness Services for WebSphere Sensor Events with position data for the tags. Use the following portlet to define these providers and relate them to specific areas:

- “Event Provider”

Log in to WebSphere Application Server administrative console and click **Topology** → **Event Provider** to access this page.

A device relates to a hub, and a hub has a coordinate system. A hub can relate to multiple areas. If a tag or device position is within the area and the hub relates to this area, then the tag or device can be seen in the area. Use the following portlet to define devices and assign them to a hub:

- “Devices” on page 42

Log in to WebSphere Application Server administrative console and click **Topology** → **Location Devices** to access this page.

Gates provide access control for the entryways and exits of a zone. See “Monitoring the entry and exit in an area (gates)” on page 9. Use the following portlet to define these gates:

- “Gate Manager” on page 43

Log in to the WebSphere Application Server administrative console and click **Topology** → **Gate Manager** to access this page.

Registration units read tag IDs and make them available to you for item definition. Use the following portlet to define these units:

- “Registration Units” on page 44

Log in to the WebSphere Application Server administrative console and click **Topology** → **Registration Units** to access this page.

Boundary zones provide access control for areas that are not fully covered by devices. See “Monitoring the entrance and exit of zones that are not fully covered by devices” on page 11. Use the following portlet to define these boundary zones:

- “Boundary Zones” on page 44

Log in to the WebSphere Application Server administrative console and click **Topology** → **Boundary Zones** to access this page.

Event Provider

Use this page to define the event providers for your areas and zones.

Note: Currently location event providers are the only type of supported event provider. Location event providers provide Location Awareness Services for WebSphere Sensor Events with tag location data.

Click **Add** to define a new event provider or click **Delete** to delete an existing provider. Click **Edit** to edit details for an existing event provider.

Add new event provider

Complete the following fields to define a new event provider.

Note: You need to have defined an area before you can relate a event provider to it.

- **Hub Base Parameters:** These parameters are used to define the event provider.
 - **Name*:** Enter a name for the event provider.
 - **ID:** This field is read-only. It is the internal ID used by the simulator configuration properties or by the location events coming from the adapter framework.
 - **Description:** Enter a description of the event provider.
 - **Related App Server ID:** Enter the IP address for the related WebSphere Application Server. This field is required if you choose a connectivity type other than JMS Gateway.

- **Connectivity:**

- **Connection Type*:** Select the type of connection to make to the event provider: JMS Gateway, Socket Gateway, and LAS Socket.
- **Parameter:** This value is pre-populated based on your selection of a connection type:

Note: Brackets [] indicate optional key value pairs.

- **JMS Gateway**

```
IPAddress=<hub IP address>;Port=<hub listener port>;
```

- **Socket Gateway**

```
IPAddress=<hub IP address>;[Port=<hub listener port>;]  
[UseUDP=<true|FALSE>;][BufferSize=<buffer size in bytes>;]  
[BinaryData=<true|FALSE>;][HeartbeatFrequency=<seconds between heartbeats>;]  
[DelayBetweenCommands=<seconds between commands>;]
```

- **LAS Socket**

```
IPAddress=<hub IP address>;[Port=<hub listener port>;]  
[UseUDP=<true|FALSE>;][BufferSize=<buffer size in bytes>;]  
[BinaryData=<true|FALSE>;][HeartbeatFrequency=<seconds between heartbeats>;]  
[DelayBetweenCommands=<seconds between commands>;]
```

- **Input Event Conversion:** If your connection type is something other than JMS Gateway, then you can choose an input event conversion method from the list of available methods. If JMS Gateway is selected, the **Input Event Conversion** fields are not available. Input event conversion methods transform provider-specific events into Location Awareness Services for WebSphere Sensor Events internal events. Different event providers require different conversion methods.
 - **Implementation Name:** Select the name of the event conversion implementation to use. The other fields are pre-populated when you select an implementation name. The default implementation name is LAS for Location Awareness Services for WebSphere Sensor Events.
 - LAS
 - Data Capture

- **Implementation Class:** This value is populated based on your selection of an implementation name. This is the Java™ implementation class used for conversion.
- **Parameters:** This value is pre-populated based on your selection of an implementation name; however, you have to instantiate the template by replacing placeholder values with valid values.

All event converters shipped with Location Awareness Services for WebSphere Sensor Events support the optional parameter `IDPrefix=string`. This parameter can be used for making tag IDs unique across a multi-event provider installation. The provider tag IDs are prefixed by the value specified in the `IDPrefix` parameter throughout Location Awareness Services for WebSphere Sensor Events.

The `com.ibm.atlas.event.conversion.LASEventConverter` implementation class supports the following parameters:

- `ignoreTagIDs=name of file containing tag IDs` - For this parameter, enter the name of a file that contains tag IDs that should be ignored.
- `providerLocale=ISO-639 code` - This parameter is relevant if the actual locale is different from the default locale and, for example, numeric values need to be converted.

Parameters in square brackets (`[parameter]`) in the parameter template are optional. In general, parameters are keyword-value pairs separated by semi-colons (;).

Note: Whenever you add real values to the parameter template, remember to remove the square brackets.

Note: Contact your IBM Services representative in order to use a custom implementation.

- **Transformation options:** These parameters are used to convert the coordinates returned by the event provider into appropriate coordinates for the area, and therefore to transform the area displayed on the Spatial Management Client. You can shift or displace the area, change its scale, rotate it, or juxtapose its position.
 - **X Offset:** Enter a value to offset the area on the X-axis. The offset value can be negative or positive and is 0.0 by default, indicating no offset.
 - **Y Offset:** Enter a value to offset the area on the Y-axis. The offset value can be negative or positive and is 0.0 by default, indicating no offset.
 - **Z Offset:** Enter a value to offset the area on the Z-axis. The offset value can be negative or positive and is 0.0 by default, indicating no offset.
 - **Scaling:** Enter a value greater than 0 to change the scale of the area. The default value is 1.0, or no scaling.
 - **Horizontal Rotation:** Enter a value between 0 and 360 to rotate the area. This value specifies an angle in degrees and has a default value of 0, or no rotation.
 - **X-Y Permutation:** Select this box to switch the X and Y coordinates.
- **Smoothing Algorithm:** No default smoothing algorithm is selected when you create a new provider. Instead, the **Implementation Name** field is pre-filled with a value of **None**, indicating that no smoothing is applied.
 - **Implementation Name:** Select the name of the smoothing algorithm implementation to use. The other fields are pre-populated when you select an implementation name. For example, you can choose between the following types:
 - **None** - The default value; no smoothing is applied.

- **WeightedUpdate** - The new position of a tag is calculated based on its previous position maintained by Location Awareness Services for WebSphere Sensor Events, and the new position is reported by the external event provider. The new position is calculated as follows: $\text{newX} = \text{averagingFactor} * \text{currentX} + (1.0 - \text{averagingFactor}) * \text{oldX}$, and similarly for the other coordinates.
- **MovingAverage** - The new tag position provides position estimates computed according to a moving average algorithm. It accepts as properties the length of the time series on which to operate and the weights for each element in the series. The sum of the weights must be equal to one.
- **HammingWindow** - This is a special version of the MovingAverage smoothing algorithm, where the length of the time series on which to operate is specified. The weighting of the positions is done according to the HammingWindow algorithm.
- **Implementation Class:** This value is populated based on your selection of an implementation name.
- **Parameters:** This value is pre-populated based on your selection of an implementation name; however, you can modify the value. These are the customization parameters for the smoothing algorithm. Specify them in the *keyword=value;keyword=value* format.
 - **WeightedUpdate:** $\text{AveragingFactor}=0..1$, where *0..1* is replaced with a decimal *i* between zero (0) and one (1), for example, 0.5.
 - **MovingAverage:** $\text{TimeSeriesLength}=n; \text{Weights}=0.x, 0.x, 0.x$, where *n* is replaced by a number greater than one (1), and *0.x, 0.x, 0.x* is replaced by values between zero (0) and one (1) which sum up to one (1). You can also specify $\text{Weights}=\text{equal}$, where each position in the series is counted as the same length.
 - **HammingWindow:** $\text{Hamming}=\text{true}; \text{TimeSeriesLength}=n$, where *n* is replaced by a number greater than zero (0).
- **Start Options:** If the **Auto-start** box is selected, communication with the RTLS system behind the event provider definition is started automatically as soon as WebSphere Application Server is started.

Note: If communication with the RTLS system is through an external adapter, this adapter has to be up and running at this time. Otherwise, communication with the RTLS system cannot start.

- **Associated Areas:** Select one or more areas you want to associate with the event provider. If no area is selected, the location events coming from this event provider will not result in any booking of tags to an area.

After you switch to this area in the Spatial Management Client, you might still see tags moving around – even after you have removed the association between an area and a event provider instance. This can happen even if there is no longer an association between the area and a event provider instance. An internal cleanup is made when you delete the area; however, this strange effect cannot be suppressed due to the manner in which Location Awareness Services for WebSphere Sensor Events internally maintains tag-to-area associations.

Click **Save** to save your settings or click **Cancel** to exit without saving your changes. Click **Reload** to refresh the options that are available from the menus and to reset the fields to their original state.

Note: Fields marked with an asterisk (*) are required. All other fields are optional.

Devices

Use this page to define your devices and assign them to a hub (event provider).

Devices can be either readers or a device group to which you can associate several readers. Devices must be defined here if they play a specific role. For example, if a reader represents part of a gate or if a device group represents part of a boundary zone.

A device relates to a hub, and a hub has a coordinate system. A hub can relate to multiple areas. If a tag or device position is within the area and the hub relates to this area, then the tag or device can be seen in the area.

Click **Add** to define a new device or click **Delete** to delete an existing device. Click **Edit** to edit details for an existing device.

Note: Before you can add a new device, you must have defined an event provider in the **Topology > Event Provider** page.

Add a new device or edit an existing device

Complete the following fields to define a new device or to update an existing one:

- **Name*:** Enter a unique meaningful name for the device.
- **Definition:** Enter a description of the device.
- **ID:** Enter a unique ID for each event provider for the device. An ID can be a number greater than 0. Do not use 0.
- **Type*:** Choose either **Device Group** or **Reader** for the device type. If you choose **Device Group**, you can associate other devices with this group.
- **Hub*:** Select the type of event provider to use for the device.
- **Device Location:** Choose one of the following:
 - **No Location** - No position for the device is defined.
 - **Static Location** - Position is defined by X, Y, and Z coordinates. The coordinates are for the hub to which the device belongs. Static locations are recommended for fixed devices.
 - **Dynamic Location** - Position is related to a tag ID. Dynamic locations are recommended for devices associated with a mobile and active tag.

Tag IDs are attributes of items, and they can change dynamically. If you relate a device to a tag ID, then you need to make sure that the tag ID is valid. In addition, if a tag ID is removed from the system, there is no alert to let you know that a device is still associated with that tag ID.
- **Associated Devices:** If you are in the process of defining a device group, you can assign other already defined readers to this group. Use the arrows to associate devices with your device group or to remove associated readers from the group.
- **Assignable Devices:** These are devices that can be added to the collection of devices that makes up a device group. Only devices belonging to the same hub (event provider) can be added. Sample devices, such as readers, and previously defined device groups can be added.
- **Assigned Devices:** These are devices that currently belong to the device group.

Click **Save** to save your settings or click **Cancel** to exit without saving your changes.

Note: Fields marked with an asterisk (*) are required. All other fields are optional.

Gate Manager

Use this page to define gates.

A *gate* allows you to have only one device that specifically monitors the entry to or exit from a zone.

When monitoring zones in areas, define the gate twice: once for the zone and once for the area. Otherwise, Location Awareness Services for WebSphere Sensor Events cannot correctly monitor tag counts for the zone and area.

Click **Add** to define a new gate, click an existing gate to edit it, or select a gate and click **Delete** to delete it.

Add new gate

Complete the following fields to create a gate:

Note: Before you can add a new gate, you must have defined all devices for the associated event provider from the **Topology > Devices** page.

- **Name*:** Enter a unique, meaningful name for the gate.
- **Description:** Enter a description of the gate.
- **Area*:** Select the area to associate with the gate.

Note: For scenarios where proximity data (p-data) is used for tag positions, be sure that the gate you define belongs to the area itself, and not to a zone within the area. Otherwise, the tags may not be visible in the Spatial Management Client.

- **Zone:** Select the zone to associate with the gate. If no zone is selected, the definition applies to the whole area.
- **Hub*:** Select the event provider to associate with the gate.
- **Device*:** Select the name of the device you have already defined using the **Devices** portlet.
- **Role*:** Select the role of the gate.
 - Select **IN** to specify that the associated device monitors tags entering the gate. When the device sees a tag in the associated zone, it considers the tag to be inside the zone. The coordinates of the tag are those reported by the event provider. An event or alert is logged to indicate that the tag entered the zone.
 - Select **OUT** to specify that the associated device monitors tags exiting the gate. When the device sees a tag, it considers the tag to be outside of the zone being monitored. An event or alert is logged to report that the tag left the zone.
 - Select **IN/OUT** to specify that a tag is:
 - Logged in to the zone associated with the device as long as it is "seen" by this device only
 - Logged out if it is not seen by the device anymore (after some delay) or if it is seen by any other devices

Click **Save** to save your settings or click **Cancel** to exit without saving the changes. Click **Reload** to clear the fields or click **Back** to go back to the previous step.

Note: Fields marked with an asterisk (*) are required. All other fields are optional.

Registration Units

Use this page to define the registration units for your areas and zones.

You can designate a registration unit to provide a way to register tag IDs with Location Awareness Services for WebSphere Sensor Events when you create items. For example, you can define an event provider as a registration unit and then its signals can be used to read tag IDs into Location Awareness Services for WebSphere Sensor Events when defining items; you do not have to enter the tag IDs manually.

Prerequisite: When defining an event provider as a registration unit, you must have defined it already in the **Event Provider** portlet before you define it as a registration unit.

If you define a device group or a single device as a registration unit, the remaining devices of the event provider can be used for regular monitoring. Otherwise, if you designate the entire event provider as a registration unit, do not use it for real-time tag reporting.

Click **Add** to define a new registration unit or click **Delete** to delete an existing registration unit. Click **Edit** to edit details for an existing registration unit.

Add new registration unit

Complete the following fields to create a new registration unit:

- **Unit Name*:** Enter a unique, meaningful name for your registration unit.
- **Description:** Enter a description for the registration unit.
- **Hub*:** Choose an event provider from the list of defined providers.
- **Device:** Choose the associated device (which can be a device group or a simple device) from the list, if applicable. Devices are listed only if they have been previously defined.

Click **Save** to save your settings or click **Cancel** to exit without saving the changes. Click **Reload** to refresh the options available from the menus and to reset the fields to their original state.

Note: Fields marked with an asterisk (*) are required. All other fields are optional.

Boundary Zones

Use this page to define boundary zones for critical zones in your area.

You use boundary zones to monitor areas that are not fully covered by devices. For example, inner and outer boundary zones can be set up at the entrance and exit of areas that are not fully covered by devices. They provide position data on tags entering and leaving those areas. If an item is detected in the outer zone and then in the inner zone, and eventually disappears or cannot be located, Location Awareness Services for WebSphere Sensor Events assumes that the item is now within the area protected by the two zones. As a result of this function, inner and outer boundaries can be used to implement a light barrier.

Click **Add** to define a new boundary zone.

Add new boundary

Adding a new boundary definition consists of two steps: first, you define the zone that makes up the boundary, and then you define the zone within the boundary. The latter zone is called a *related zone*. Complete the following fields to create a new boundary zone.

Notes:

- Create all boundary zones and related zones in the Spatial Management Client before you define them here. In the Spatial Management Client, the term *barrier zone* is a synonym for a *boundary zone*.
- Do not define a zone as a boundary zone of itself.
- **Area:** Select the area in which you are defining the boundary zone.
- **Zone:** Select the zone you are defining as a boundary zone. It must be in the area you selected.
- **Boundary Type:** Select the type of boundary zone you are defining:
 - **Inner** - A zone where tags are considered to be in the target zone, even if not visible. Use an **Outer** boundary zone with this type of zone.
 - **Outer** - A zone where tags are considered to be out of the target zone. Use an **Inner** boundary zone with this type of zone.
 - **Single** - A zone where tags are considered to be in the target zone, even if not visible. Do *not* use an **Outer** boundary zone with this type of zone.
- **Related Area:** Select the area of the related zone.
- **Related Zone:** Select the related zone.

Once all the required input has been entered, you can click **Save** to save your settings. The **Save** button is not visible until all required input has been entered. Click **Cancel** to exit without saving your changes. Click **Reload** to refresh the options that are available from the menus and to reset the fields to their original state.

Planning for classes and items

This topic describes how to plan effectively for Location Awareness Services for WebSphere Sensor Events classes and items.

To plan effectively for Location Awareness Services for WebSphere Sensor Events classes and items, consider some basic rules and concepts behind the Location Awareness Services for WebSphere Sensor Events classes as described below.

Class structure

When defining classes and items, start by defining your class hierarchy along with attributes and properties. Then, you can associate items with the classes. Location Awareness Services for WebSphere Sensor Events classes make up a hierarchical tree, so remember the following:

- It is not recommended to add a class without key properties because then you cannot use the AtlasIntegrator application to maintain the items and several item-related Web services will not function.
- When you delete a class, it also deletes all subclasses and items belonging to the deleted class or subclass.
- When you add a subclass, it inherits all properties from the parent class. These inherited properties cannot be changed at the subclass level.

- When maintaining class properties, keep the following points in mind:
 - Make sure you identify whether subclasses have been defined for the class and whether items were defined for the class or subclass.
 - Make sure you use unique property names throughout the class hierarchy. For example, if the class *Employee* has a property named *BadgeNumber* and if the subclass, *SecurityPersonnel*, represents guards who have special badges in addition to the normal employee badge, give the special badge number a unique property name such as *SecurityBadgeNumber*.
 - Changes to key properties are restricted when the class or subclass has items defined. Therefore, the following restrictions apply:
 - Adding key properties is allowed only if no items are defined for the class or any of its subclasses.
 - Deleting key properties is allowed only if no items are defined for the class or any of its subclasses.
 - Changes to key properties are usually allowed only if they are less restrictive.
 - Renaming key properties is allowed.
 - You can only change a key property type from any value to a string. The values are kept.
 - Changes to other properties are restricted when the class or subclass has items defined. Therefore, the following restrictions apply:
 - Adding other properties is allowed only if no items are defined for the class or any of its subclasses.
 - Deleting other properties is allowed only if no items are defined for the class or any of its subclasses.
 - Changes to other properties are usually allowed only if they are less restrictive.
 - Renaming other properties is allowed.
 - You can only change a property type from any value to a string. The values are kept.
 - Changing a property from mandatory to optional is allowed, but you cannot make an optional property mandatory.
- Minimize the number of levels in your class hierarchy. Too many levels can make the display unusable and can decrease performance.

Class name length

Depending on font size and screen resolution, long class names might be truncated in the Spatial Management Client and Location Awareness Services for WebSphere Sensor Events Administrative Console.

Subclasses are shown with an indentation that depends on the level in the class hierarchy. In order to display the full names, use the following guidelines:

- Top-level classes should not be longer than 15 to 20 characters (depending on your resolution).
- The name length should decrease by about 20 percent per level.

Defining classes, items, and groups

This topic lists the portlets you use to define classes, items, and groups.

About this task

Use the following portlets to define classes, items, and groups:

- “Classes/Items Manager”
Log in to the WebSphere Application Server administrative console and click **Tag Registration** → **Classes/Items Manager** to access this page.
- “Groups Manager” on page 51
Log in to the WebSphere Application Server administrative console and click **Tag Registration** → **Groups Manager** to access this page.

Classes/Items Manager

Use this page to define classes and individual items for a class, for example, Asset and Person.

You can define classes or subclasses depending on your needs and then define individual items for those classes.

Click **Add Child Class** to define a new class; or click an existing class to edit the fields or to define new items for the class.

Note: If items are associated with the class or any of its subclasses, you can change the **Description**, **Icon Link**, **Update Interval**, **Stationary**, **Container**, **Edge Length**, and **Smoothing Algorithm** attributes. All other attributes and properties are read-only and cannot be changed. The Smoothing Algorithm is not inherited; changing it has no effect on subclasses or items belonging to those subclasses.

Therefore, to change the class, complete the following steps:

1. Delete the items associated with the class.
2. Change the class.
3. Add the items associated with the class.

When you delete a group, class, or zone, any rules you created that refer to those entities are not automatically deleted. You need to clean up any related rules when you delete a group, class, or zone. If you create a new group, class, or zone with the same name as one you deleted, and you have not cleaned up the old related rules, then the old rules will not apply, even if you intended them to be applicable.

Rules related to items and its properties are always triggered when an item with the properties matches the condition.

Class Details View

Click the **Class Details View** tab to view the details for a class. This view opens automatically when you click **Add Child Class** or click an existing class.

Complete or edit the following fields:

- **Class Name*:** Enter a unique, meaningful name for the class.
- **Description:** Enter a description of the class.
- **Icon Link:** Enter the file name of the graphic icon to display for this class of items. All items in the class are displayed in the Spatial Management Client with the icon.

- **Update Interval:** Enter the number of seconds Location Awareness Services for WebSphere Sensor Events waits before processing location data and updating the location of the icon on the Spatial Management Client. Set this field to a higher number for items that move slowly or not at all to reduce server load. For example, if the tagged item is a mainframe computer, set the field to a higher number because it does not make sense to check its position each second.
- **Stationary:** Check this field if the class is made up of items that should not be removed from a specific location, such as hardware assets.
- **Container:** Check this field to define items in this class as containers. This means they can contain other items.
- **Edge Length:** Enter the size of the container (assume it is a cube). This value allows rules checking in later releases, such as when an item cannot leave a container. You can define an edge length at class-level or an edge length for each item. The class-level edge length is inherited by each item, but if an edge length is specified for an item, it overrides the value set at the class-level
- **Smoothing Algorithm:** No default smoothing algorithm is selected when you create a new provider. Instead, the **Implementation Name** field is pre-filled with a value of **None**, indicating that no smoothing is applied.
 - **Implementation Name:** Select the name of the smoothing algorithm implementation to use. The other fields are pre-populated when you select an implementation. For example, you can choose between the following types:
 - **None** - The default value; no smoothing is applied.
 - **WeightedUpdate** - The new position of a tag is calculated based on its previous position maintained by Location Awareness Services for WebSphere Sensor Events, and the new position is reported by the external event provider. The new position is calculated as follows: $\text{newX} = \text{averagingFactor} * \text{currentX} + (1.0 - \text{averagingFactor}) * \text{oldX}$, and similarly for the other coordinates.
 - **MovingAverage** - The new tag position provides position estimates computed according to a moving average algorithm. It accepts as properties the length of the time series on which to operate and the weights for each element in the series. The sum of the weights must be equal to one.
 - **HammingWindow** - This is a special version of the MovingAverage smoothing algorithm, where the length of the time series on which to operate is specified. The weighting of the positions is done according to the HammingWindow algorithm.
 - **Implementation Class:** This value is populated based on your selection of an implementation name.
 - **Parameter:** This value is pre-populated based on your selection of an implementation name; however, you can modify the value. These are the customization parameters for the smoothing algorithm. Specify them in the *keyword=value;keyword=value* format.
 - **WeightedUpdate:** $\text{AveragingFactor}=0..1$, where *0..1* is replaced with a decimal *i* between zero (0) and one (1), for example, 0.5.
 - **MovingAverage:** $\text{TimeSeriesLength}=n;\text{Weights}=0.x,0.x,0.x$, where *n* is replaced by a number greater than one (1), and *0.x,0.x,0.x* is replaced by values between zero (0) and one (1) which sum up to one (1). You can also specify $\text{Weights}=\text{equal}$, where each position in the series is counted as the same length.
 - **HammingWindow:** $\text{Hamming}=\text{true};\text{TimeSeriesLength}=n$, where *n* is replaced by a number greater than zero (0).

You can define key properties, properties, or child classes for each class you create. From the menu, choose from the following actions:

- **Key Properties**

Key properties are mandatory for a class item. Define key properties so that all members of the class can be clearly identified. For example, a person's social security number is an adequate key property, but a person's first and last names are not adequate key properties, even when used together, because there might be two people using the same first and last names.

Important: Make sure to define key properties with unique names.

- **Name:** Enter a unique name for the property.
- **Type:** Select the type of value that should be entered for the property. For example, you can choose among the following types:
 - **text** - A text field. For example, select this type for a name property.
 - **textarea** - A text field with space for more characters. For example, select this type for an address property.
 - **checkbox** - A check box. For example, select this type for a property where the default is true or false.
 - **integer** - A field that allows only numeric values.
 - **date** - A calendar. Select this type for a property that will always be a date.

Beside each property that you want to delete, click **Mark for Deletion**; then click **Save** to save your changes.

Note: You cannot delete a key property if there are any items, subclasses, or items in a subclass defined.

- **Properties**

Properties can either be optional or mandatory for a class item.

Important: Make sure to define properties with unique names.

- **Name:** Enter a unique name for the property.
- **Type:** Select the type of value that should be entered for the property. For example, you can choose between the following types:
 - **text** - A text field. For example, select this type for a name property.
 - **textarea** - A text field with space for more characters. For example, select this type for an address property.
 - **checkbox** - A check box. For example, select this type for a property where the default is true or false.
 - **integer** - A field that allows only numeric values.
 - **date** - A calendar. Select this type for a property that will always be a date. When entering this property, click **PickDate** to select the date from the calendar or **ClearDate** to clear your selection.
- **Min Occurs:** Enter a value indicating the minimum occurrences of the property. This value should be less than or equal to the **Max Occurs** setting. For example, enter 0 if the property is optional and 1 if it is required.
- **Max Occurs:** This property cannot be modified. The value is 1, indicating that it can occur only one time.
- **Default Value:** Enter a default value for the property. For example, for the Company property, enter the name of your company. This value can be modified when you create an item for the class.

Beside each property that you want to delete, click **Mark for Deletion**; then click **Save** to save your changes.

- **Add key property**
Enter the values for each key property as necessary.
- **Add property**
Enter the values for each property as necessary.
- **Add Child Class**
Enter the values for the class, including defining key properties, properties, and child classes (also called subclasses) for each child class, as necessary.
- **Save**
Click **Save** to save the class or child class you are creating, as well as all key properties and properties defined for the class.
- **Delete**
Click **Delete** to delete the class, its sub classes, and all items in the class. All subclasses and items in the class are deleted.
- **Reload**
Click **Reload** to refresh the options available from the menus and to reset the fields to their original state.

Item View

Click the **Item View** tab to view the items that have been defined for a class. You can add or edit new items for a class, assign items to groups, or delete selected items. If the item is defined as a container class, you can assign other items to the container item.

Items in this view are sorted by tag ID in ascending order. Items that do not have a tag ID are listed at the end.

Select **Add Item** to create a new item. Complete the following fields:

- **Registration Unit:** (Optional) If a registration unit has been defined and you are using it to read tag IDs into Location Awareness Services for WebSphere Sensor Events, select the registration unit. This field is not always available.
- **Tag ID:** Enter the tag ID for the item.
Enter the tag ID manually or use an external device, such as a bar code reader. Additionally, if you defined a registration unit, you can select the appropriate tag ID from the tags that are read by the registration unit.
- **Icon Link:** Enter the name of the graphic icon file. By default, the icon associated with the class displays for the item.
- **Icon Label:** Enter a label to identify the item. The label helps you quickly identify tags in Location Awareness Services for WebSphere Sensor Events alerts, in search results, and on the Spatial Management Client, which allows you to view an icon label beside the tag. If an item is created or modified using import, you can specify rules for automatic label creation, such as building a label consisting of a person's first name, middle name, and last name. If the item is defined as a container class, the value for **Edge Length** is prefilled with the edge length defined for the class and can be modified for the single item.

Complete any additional fields, which vary by class.

Click **Save** to save your settings or click **Cancel** to exit without saving.

After an item is defined, you can complete the following actions:

- **Delete Items**

Under **Choose Action**, select **Delete Items** to delete selected items.

- **Edit Properties**

Click **Edit Properties** to edit an existing item.

- **Edit Groups**

Click **Edit Groups** to assign the item to a group or to remove it from a group.

Note: You can select one or multiple groups, dependent on the HierarchicalGroups system property.

Click **Save** to save your settings or click **Cancel** to exit without saving.

- **Edit Container** (only available if the item is a container)

Click **Edit Container** to assign items to the container. A list of items that can be assigned to the container are listed, as well as a list of any items that have already been assigned to the container, if any. The items are listed by class.

Note: To assign items to the container, select one or more items under containable items. To remove assigned items from the container, select one or more items under direct children.

Click **Save** to save your settings or click **Cancel** to exit without saving.

Filtering the Item View

You can also filter the item view. Enter a string in the text field and then click **Apply Filter**. The items will be filtered according to the string you enter. If any of the property values or tag IDs for an item contains the string, the item will be shown. Click **Clear Filter** to clear the filter criteria.

Note: Fields marked with an asterisk (*) are required. All other fields are optional.

Groups Manager

Use this page to define groups.

Click **Add** to define a new group or click an existing group to edit it. You can also use this page to view items in a group, add an item to a group, and remove an item from a group.

Add new group

Click **Add** and then enter values in the following fields to create a group:

- **Group Name*:** Enter a unique, meaningful name for the group.
- **Description:** Enter a description of the group.
- **Group Color:** Select the color that you want to use to identify the group. You can only assign a group color if the system property HierarchicalGroups is set to Y. The icons of group members are outlined in the selected color in the Spatial Management Client.

Click the arrows to add or remove groups to or from the **Group Members** column. The **Possible Group Members** column lists all defined groups.

Click **Save** to save the group you are creating or click **Delete** to delete the group.

Note: Fields marked with an asterisk (*) are required. All other fields are optional.

View items in a group

Complete the following steps to view items in a group:

1. In the Group Manager portlet, click the group for which you want to modify the item membership.
2. Click **ItemsView**.
3. Optionally, to reduce the items to choose from or to search on specific criteria, enter search criteria in the **Filter Items over properties** field and click **Apply Filter**.
4. In the **Only show items of the following class** field, select the class for which you want to view items or select **All classes**.

Add items to a group

Complete the following steps to add an item to a group:

1. Click the group you want to add items to and then click **ItemsView**.
2. In the **Possible Group Members** column, select the item or items you want to add to the group and click the left arrow button (<<). The item now appears in the **Group Members** column.

Note: If the **Possible Group Members** column is not visible, click **Show possible group members**. To hide the column, click **Hide possible group members**. Any filter criteria you have specified applies to all items on the groups portlet and, therefore, also limits the list of possible group members.

3. When you finish adding items, click **Save**.

Remove an item from a group

Complete the following steps to remove an item from a group:

1. Click the group you want to remove items from and then click **ItemsView**.
2. In the **Group Members** column, select the item or items you want to remove from the group and click the right arrow button (>>). The item is removed from the **Group Members** column.
3. When you finish removing items, click **Save**.

Defining rules

This topic lists the portlet you use to define rules.

Rules define conditions and policies that need to be met. For example, rules can be used to restrict access to certain zones or to limit the amount of time an item stays in a zone. Business rules are implemented based on a generic Complex Event Processing (CEP) engine which facilitates the development of additional rule types. Events (alerts) occur when rules are violated. Events are published and saved in the event database. Subscriber programs can subscribe to Location Awareness Services for WebSphere Sensor Events events. Violations of rules related to items can also be displayed in the Spatial Management Client.

To define rules, use the following portlet:

- “Business Rules” on page 53
Log in to the WebSphere Application Server administrative console and click **Rules/Alerts** → **Business Rules** to access this page.

Business Rules

Use this page to define the business rules for your zones. Rules define the circumstances that trigger alerts.

Note: You need to define a zone before you can define a rule for it.

When you define a rule, you can check for future events, but not for events that have already happened. This means that you cannot define a rule for conditions that have already occurred. For example, if a person is already in Zone A and you then define a rule that items defined in the Person class cannot enter Zone A, the result is no alert. No alert occurs because rule checking is triggered by zone entry and exits events, which did not occur in this situation. The logic is similar with a rule for duration of stay. If a person has already been in Zone A for 30 minutes, and you define a rule that the person cannot stay in Zone A for longer than 10 minutes, no alert occurs.

When setting rules, keep in mind that tag IDs are attributes of items, and they can change dynamically. If you relate a rule to a tag ID, then you need to make sure that the tag ID is valid. In addition, if a tag ID is removed from the system, there is no alert to let you know that the rule associated with that tag ID is not longer valid.

Click the type of business rule you want to work with. Business rule types include:

- **“Duration of Stay in Zone” on page 54** - Define rules indicating how long persons or items can remain in a zone before an alert is triggered.
- **“Items Association” on page 55** - Define rules indicating associations between items.
- **“Man Down Detection” on page 56** - Define rules indicating how long a tag, typically associated with a person, can remain stationary or not blink before an alert is issued. For example, you can specify a rule to detect a person who may have had an accident and cannot move.
- **“Maximum Items per Zone Threshold” on page 57** - Define rules indicating the maximum number of items or persons that can be in a zone at one time.
- **“Visitor Escorting” on page 57** - Define rules indicating how visitors to the location will be escorted. For example, you can specify who will escort visitors or how far away from an employee a visitor can be before an alert is triggered.
- **“Zone Access Restriction” on page 58** - Define rules indicating persons and items that are not allowed to enter zones during specified time frames.
- **“Zone Exit Restriction” on page 59** - Define rules indicating persons and items that are not allowed to leave zones during specified time frames.

A list of rules displays that are of the business type you selected. Click **Add** to define a new rule. Click **Delete** to delete an existing rule or click **Edit** to edit details for an existing rule.

The following fields are available for all business rules except for Visitor Escorting:

- **Activity**
Specify the time frame when the rule should be applied. You can specify to **Always** apply the rule, to apply it **From** a specific date and time (*yyyy/mm/dd hh:mm:ss*) **To** another specific date and time, or to specify a repetitive time frame for the new rule to be applied, for example outside of normal work hours on all weekdays and all day on weekends. You can also choose to **Invert** the time frame, meaning that the rule only applies during times outside the specified time frame.

Note: Times are relative to times on the database server. The machines that host the database server and WebSphere Application Server must be set to the same time zone.

For example, if you check **Monday** and specify 08:00 as the start time and 17:00 as the end time, the rule is active on Mondays between the hours of 8 am and 5 pm. If you choose to **Invert** the rule, then it applies on Mondays except between the hours of 8 am and 5 pm (instead the rule applies from midnight to 8 am and from 5 pm to midnight).

- **Actors**

Specify the class, group, and other filter criteria for items to which the rule applies. You can specify all three, if needed. If you specify a **Class**, then all class-specific **Attributes** are selectable. Otherwise, only the tag ID and label attributes are selectable. If you select criteria for **Inclusion**, the rule applies for all items that match the filter criteria specified. If you select **Exclusion**, the rule applies to all items except those matching the filter criteria. If no **Class**, **Group**, or **Attribute** is specified in both the **Inclusion** and the **Exclusion** sections, the rule applies to every defined item.

For example, if you want the rule to apply to all items in the Person class, except for those in the Security group, specify the **Class** value as **Person** for inclusion and the **Group** value as **Security** for exclusion. You can also exclude the rule from applying to specific people or items by filling in the **Zone** values in the **Exclusion** column.

- **Zones**

Specify the **Zone** and **Zone Type** to which the rule is restricted. Similar to **Actors**, you can specify **Zones** for **Inclusion** and **Exclusion** by filling in the values in the respective columns. In both cases, you can select a single zone or all zones.

Duration of Stay in Zone

This rule type allows you to specify how long specified persons or items that can be in a specific zone at one time before an alert is triggered.

Complete the following fields to create a new rule:

- **Identification**

Enter a **Name** and **Description** for the new rule.

- **Activity**

- **Actors**

- **Zones**

- **Additional Parameters**

Specify the **Maximum duration of stay, in seconds** that an actor can stay in the zone during the specified time. For example, if you specify 120, then if a specified actor stays in the zone for more than 120 seconds, an alert will be triggered.

- **Alert Actions**

Specify the alert action to take when the rule is violated. You can specify a combination of the following actions:

- **Display Alert** displays the issue in the Spatial Management Client.
- **Log Alert** sends the alert to CEI with a `LasDurationOfStay` event type. You can also define a notification channel to call a program as a result of this alert. For example, the alert action could be to send an email.

- **Customized Notification** calls a notification program outside of CEI and requires special customization.

Items Association

This rule type allows creating associations between items. An association is created when two items fulfilling the specified filtering criteria are close to each other for a configurable period of time. Even though the rule can be applied to classes and group of items, associations are always created between two individuals.

Complete the following fields to create a new rule:

- **Identification**

Enter a **Name** and **Description** for the new rule.

- **Base Item**

Specify the tag ID, class, or group of persons or items to which other items are to be associated. You can specify an individual person or item by specifying the **Tag ID** or you can specify a **Class** or **Group** of persons or items to which the rule applies. There can be multiple base items.

Notes:

- The base items must be members of classes that have been specified as containers, and the `DynamicContainerSupportOn` system property must be selected.
- The edge length must be greater than zero.

- **Associated Item**

Specify the tag ID, class, or group of persons or items to be associated with others. You can specify an individual person or item by specifying the **Tag ID** or you can specify a **Class** or **Group** of persons or items to which the rule applies. There can be multiple associated items. The item to be associated must not be a container item.

- **Zone Selection**

Specify the **Zone** and **Zone Type** to which the rule applies.

- **Additional Parameters**

Specify the **Maximum distance, in units** that the base item and the associated item may be apart and still be associated. For example, if you specify 10, then an association only takes place if the associated item is within a radius of 10 from the base item.

Note: Currently, the edge length of the base item (that is in the container class) determines the maximum distance.

Specify the **Minimum time of being together, in seconds**. This means the minimum period of time that both items must be close to each other before an association is created and an alert is triggered. For example, if you specify 120, then two items must be not more than 10 feet away from each other for at least 120 seconds before an alert will be triggered.

- **Alert Actions**

Specify the alert action to take when the rule is violated. You can specify a combination of the following actions:

- **Display Alert** displays the issue in the Spatial Management Client.
- **Log Alert** sends the alert to CEI. You can also define a notification channel to call a program as a result of this alert. For example, the alert action could be to send an email.

- **Customized Notification** calls a notification program outside of CEI and requires special customization.

Tips for using this rule:

- Both the container tag and the associated tag must move into the target zone *after* the zone-related rule has been defined.
- The associated tag must be "inside" the edge length of the container tag longer than the minimum time without interruption.
- The base item must be the last item moving because the association rule is evaluated when the container items move.
- Both the base item and the associated tag must blink continuously in order for the association to occur.

Man Down Detection

This rule type allows detection for a tag that either did not blink or did not move for more than a configurable period of time. Typically, this rule is applied to a tag associated with a person in order to detect a "man down" situation in hazardous zones. For this rule, alerting does not happen immediately because it is dependent on the WatchdogDelay system property.

Note: The MaxUnrecognizedMovement system property is used for specifying which minimum location change is interpreted as a movement. Keep in mind that real-time location systems (RTLS) may report slightly different positions for tags that do not move in reality.

Complete the following fields to create a new rule:

- **Identification**

Enter a **Name** and **Description** for the new rule.

- **Activity**

- **Actors**

- **Zones**

- **Additional Parameters**

Specify the **Down time, in seconds** that is the maximum time a tag may either reside at the same place or be nonresponsive before an alert is issued. For example, if you specify 120, then if a specified actor does not move or his tag does not blink for more than 120 seconds, an alert will be triggered.

- **Alert Actions**

Specify the alert action to take when the rule is violated. You can specify a combination of the following actions:

- **Display Alert** displays the issue in the Spatial Management Client.
- **Log Alert** sends the alert to CEI with a LASManDownDetection event type. You can also define a notification channel to call a program as a result of this alert. For example, the alert action could be to send an email.
- **Customized Notification** calls a notification program outside of CEI and requires special customization.

Maximum Items per Zone Threshold

This rule type allows you to specify the maximum number of persons or items satisfying the specified criteria that can be in a specific zone at one time. When you define a rule, make sure that none of the tags in the rule are already in the target zone. When a rule is defined for a zone, the entry and exit events are counted. The rules engine count begins at 0, so if you have 5 tags in the zone when you create the rule, then the offset is -5.

Note: If the number of tags in a zone exceeds the threshold, then you will receive an alert. If the number of tags in the zone falls below the threshold and then exceeds it again, you will receive another alert.

Complete the following fields to create a new rule:

- **Identification**

Enter a **Name** and **Description** for the new rule.

- **Activity**

- **Actors**

- **Zones**

- **Additional Parameters**

Specify the **Maximum number of actors** that can be in the zone during the specified time. For example, if you specify 10 then if more than 10 actors are in the zone at one time during the specified time frame, an alert will be triggered.

- **Alert Actions**

Specify the alert action to take when the rule is violated. You can specify a combination of the following actions:

- **Log Alert** sends the alert to CEI. You can also define a notification channel to call a program as a result of this alert. For example, the alert action could be to send an email.
- **Customized Notification** calls a notification program outside of CEI and requires special customization.

Visitor Escorting

This rule type allows you to specify rules that enforce the escorting of visitors and that govern how visitors will be escorted at the location. This rule type has less configuration options than the others, and its activity cannot be temporarily restricted.

Complete the following fields to create a new rule:

- **Identification**

Enter a **Name** and **Description** for the new rule.

- **Visitor**

Specify the tag ID, class, or group of persons or items to be escorted. You can specify an individual person or item by specifying the **Tag ID** or you can specify a **Class** or **Group** of persons or items to which the rule applies.

Note: The visitor must belong to container classes, the escort must not contain a container class, and the DynamicContainerSupportOn system property must be selected.

- **Escort**

Specify the tag ID, class, or group of persons or items that will escort the visitor. You can specify an individual person or item by specifying the **Tag ID** or you can specify a **Class** or **Group** of persons or items to which the rule applies. For example, you could specify that a class of persons could escort the visitor.

Note: The visitor must belong to container classes, the escort must not contain a container class, and the `DynamicContainerSupportOn` system property must be selected.

- **Zone Selection**

Specify either the **Zone** or **Zone Type** to which the rule applies. In other words, specify the zones in which the visitor must be escorted.

- **Additional Parameters**

Specify the **Maximum tolerated distance, in units** that the visitor can be away from the escort. For example, if you specify 10, then if the visitor is more than 10 feet away from an escort, an alert will be triggered.

Note: Currently, the edge length of the visitor (who is in the container class) determines the maximum tolerated distance.

Specify the **Maximum tolerated rule violation time, in seconds** that the visitor can be away from an escort before an alert is triggered. For example, if you specify 120, then if the visitor is more than 10 feet away from an escort for more than 120 seconds, an alert will be triggered.

- **Alert Actions**

Specify the alert action to take when the rule is violated. You can specify a combination of the following actions:

- **Display Alert** displays the issue in the Spatial Management Client.
- **Log Alert** sends the alert to CEI. You can also define a notification channel to call a program as a result of this alert. For example, the alert action could be to send an email.
- **Customized Notification** calls a notification program outside of CEI and requires special customization.

Zone Access Restriction

This rule type allows you to specify the times when specific classes or groups of persons or items cannot enter specific zones.

Complete the following fields to create a new rule:

- **Identification**

Enter a **Name** and **Description** for the new rule.

- **Activity**

- **Actors**

- **Zones**

- **Alert Actions**

Specify the alert action to take when the rule is violated. You can specify a combination of the following actions:

- **Display Alert** displays the issue in the Spatial Management Client.
- **Log Alert** sends the alert to CEI. You can also define a notification channel to call a program as a result of this alert. For example, the alert action could be to send an email.

- **Customized Notification** calls a notification program outside of CEI and requires special customization.

Zone Exit Restriction

This rule type allows you to specify the times when specific classes or groups of persons or items must not leave specific zones.

Complete the following fields to create a new rule:

- **Identification**

Enter a **Name** and **Description** for the new rule.

- **Activity**

- **Actors**

- **Zones**

- **Alert Actions**

Specify the alert action to take when the rule is violated. You can specify a combination of the following actions:

- **Display Alert** displays the issue in the Spatial Management Client.
- **Log Alert** sends the alert to CEI. You can also define a notification channel to call a program as a result of this alert. For example, the alert action could be to send an email.
- **Customized Notification** calls a notification program outside of CEI and requires special customization.

Click **Save** to save your settings or click **Cancel** to exit without saving the changes. Click **Delete** to delete the rule.

Note: Fields marked with an asterisk (*) are required. All other fields are optional.

Defining how to handle alerts

This topic helps you define how to handle alerts with Location Awareness Services for WebSphere Sensor Events.

Alerts are generated when a rule has been broken within Location Awareness Services for WebSphere Sensor Events or when there are diagnostic events from an event provider. These alerts are persisted to an event database to which you can subscribe. You define how to react to alerts by defining notification programs and notification channels. An email service is provided as a sample notification program.

Use the following portlets to define how alerts are handled:

- “Notification Program Manager” on page 60
Log in to the WebSphere Application Server administrative console and click **Rules/Alerts** → **Notification Program Manager** to access this page.
- “Notification Channels” on page 60
Log in to the WebSphere Application Server administrative console and click **Rules/Alerts** → **Notification Channels** to access this page.
- “Mail Host Configuration” on page 61
Log in to the WebSphere Application Server administrative console and click **Rules/Alerts** → **Mail Host Configuration** to access this page.
- “Mail Receiver Configuration” on page 61

Log in to the WebSphere Application Server administrative console and click **Rules/Alerts → Mail Receiver Configuration** to access this page.

Notification Program Manager

Use this page to define notification programs.

A notification program is a program or Web service that can be triggered when an event is logged.

Click **Add** to define a new program or click **Delete** to delete an existing program. Click **Edit** to edit details for an existing program.

Add new notification program

Complete the following fields to create a new notification program:

- **Notification Program Name***: Enter the name of the program. For a Web service, enter the URL for the Web service. For a batch program, enter the file name of the batch program.
- **Notification Program Description**: Enter a description of the program.
- **Notification Program Call Type**: Select **Web Service** or **Command**.
- **Notification Program Call Details***: Complete this field depending on the call type. If the program is a command, enter the directory where the program is located. If the program is a Web service, enter the name of the Web service method to be called.

Click **Save** to save your settings or click **Cancel** to exit without saving the changes.

Note: Fields marked with an asterisk (*) are required. All other fields are optional.

Notification Channels

Use this page to define notification channels.

Notification channels define the filter criteria for a given subscriber that should be called for an event. A subscriber is a program subscribing all, or a defined subset of, the events arriving for a given event group. It dispatches the arriving events to Location Awareness Services for WebSphere Sensor Events notification programs, which are programs or Web services that can be triggered when an event is logged.

Click **Add** to define a new channel or click **Delete** to delete an existing program. Click **Edit** to edit details for an existing program.

Add new notification channel

Complete the following fields to create a new notification channel:

Note: A subscriber program must exist before you can add a new notification channel.

- **Subscriber**: Select the subscriber program.
- **Program**: Select a notification program.
- **Description**: Enter a description of the program.

Complete the following fields to create a notification channel matching the filter criteria you selected. Only a positive match for valid criteria will produce valid results.

- **Tag ID:** Enter the tag ID of a person or asset.
- **Tag Label:** Enter the label of the tag.
- **Tag Class:** Enter a class of items to search for.
- **Tag Group:** Enter a group to search for.
- **Zone:** Enter a zone for the events.
- **Event Type:** Select a type of event to search for.
- **Rule Name:** Select a rule name for additional filtering, if you have several rules that lead to the same event. This menu contains most, but not all, rule names. Select **all (*)** to display events for all rules.
- **Acknowledged:** Select **All (*)** to filter all events, **Acknowledged** to filter only the events that have been acknowledged, or **Active** to filter only the events that have not been acknowledged.

Click **Save** to save your settings or click **Cancel** to exit without saving the changes.

Mail Host Configuration

Use this page to define the mail servers for your alerts.

Click **Add** to define a new mail server or click **Delete** to delete an existing server. Click **Edit** to edit details for an existing server.

Add new mail host configuration

Complete the following fields to define a new mail server:

- **Host Address*:** Enter the fully qualified host name or IP address of the mail server.
- **Port*:** Enter the mail server port number.
- **Default Sender*:** Enter the name of the default sender.
- **Default Subject:** Enter a default subject for the alert.

Click **Save** to save your settings or click **Cancel** to exit without saving your changes.

Note: Fields marked with an asterisk (*) are required. All other fields are optional.

Mail Receiver Configuration

Use this page to define the individuals or groups who will receive alert notification. Also specify what types of alerts to send notification of and when to send them.

Click **Add** to define a new mail receiver or click **Delete** to delete an existing receiver. Click **Edit** to edit details for an existing receiver.

Add new mail receiver

Complete the following fields to create a new mail receiver:

Note: Times are relative to times on the database server. The machines that host the database server and WebSphere Application Server must be set to the same time zone.

- **Receiver Name*:** Enter the name of the receiver.
- **Receiver Address*:** Enter the e-mail address of the receiver.

- **Week Days:** Select the days of the week when e-mail should be sent to the receiver.
- **Start Time*:** Enter a start time after which e-mail can be sent to the receiver each day. Use the format hour, minute, and second (HH:mm:ss).
- **End Time*:** Enter an end time after which e-mail should not be sent to the receiver each day. Use the format hour, minute, and second (HH:mm:ss).
- **Alert Type*:** Select the type of alert event to send to the receiver.
- **Mail Host*:** Enter the fully qualified host name of the mail server.

Click **Save** to save your settings or click **Cancel** to exit without saving your changes.

Note: Fields marked with an asterisk (*) are required. All other fields are optional.

Setting system properties

This topic lists the portlet you use to set system properties.

About this task

You can set Location Awareness Services for WebSphere Sensor Events system properties using the following portlet:

- “System Properties”

Log in to the WebSphere Application Server administrative console and click **System Properties** to access this page.

System Properties

Use this page to set system properties. System properties are unique and predefined.

You can edit the properties on this page. Click **Save** to save the changes.

The following system properties are used:

Table 12. Default system properties

Name	Type	Default	Description
LasVersion	String	current version number	Shows the Location Awareness Services for WebSphere Sensor Events version that is installed. This property is read-only.
AllEventsGroup	String	All events	Name of the events group that holds all Location Awareness Services for WebSphere Sensor Events events.
BIRTViewerURL	String	http://localhost:9080/birt-viewer/frameset?__report=	Use this URL to view BIRT reports. If WebSphere Application Server was installed using different ports or if the BIRT engine was installed on a different WebSphere Application Server system, modify the value of this URL to point to the correct location.

Table 12. Default system properties (continued)

Name	Type	Default	Description
BatteryExhaustedAlertAction	String	Event	<p>Alert action if the battery of a tag is completely exhausted. The alert generates an event.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • Event • Display • Event, Display • Display, Event <p>The battery icon displays in the Spatial Management Client.</p>
BatteryLowAlertAction	String	Event	<p>Alert action if the battery of a tag is low. The alert generates an event. When the alert is triggered, you will not receive another alert unless the battery rises above the threshold and then triggers it again.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • Event • Display • Event, Display • Display, Event <p>The battery icon displays in the Spatial Management Client.</p> <p>Independent of the BatteryLowAlertAction property setting, low battery conditions are indicated by a low battery icon that is added to the tag icon.</p>
BatteryThreshold	Integer	1	<p>Battery threshold that triggers an alert when the battery is underrun. The battery status can be:</p> <ul style="list-style-type: none"> • 3, which is full or completely charged • 2, which is high or sufficiently charged • 1, which is low or somewhat charged • 0, which is empty or not charged
ContainerSupportOn	Boolean	The box is not checked, which means No	<p>Turns container processing on during runtime processing, meaning you can add an item to a container. The box can either be checked to mean Yes or not checked to mean No. Use this property in scenarios with containers equipped with active tags and content items without active tags. Once entered to a container, the content items should inherit the container position.</p> <p>Note: This property should not be selected at the same time as DynamicContainerSupportOn. If it is, the ContainerSupportOn property will be ignored.</p>

Table 12. Default system properties (continued)

Name	Type	Default	Description
CurrentSVGDir	String	./svg	Holds the area definitions during replay and normal processing, when running the Spatial Management Client.
DefaultDateFormat	String	MM/dd/yyyy	<p>Format used to display dates and accept date input in Location Awareness Services for WebSphere Sensor Events.</p> <p>By specifying MM/dd/yyyy as the pattern for data pertaining to dates,</p> <ul style="list-style-type: none"> • Input to the GUI, such as 01/12/2008, will be interpreted as January 12, 2008 (not December 1, 2008). • Output from the GUI will be displayed in the specified pattern format, for example 01/12/2008. <p>For detailed information on time format syntax, refer to the Java API for SimpleDateFormat.</p>
DynamicContainerSupportOn	Boolean	The box is checked, which means Yes	<p>Turns on dynamic container processing. The box can either be checked to mean Yes or not checked to mean No. With this flag set, Location Awareness Services for WebSphere Sensor Events is able to detect by position whether a tag is near a container (using its edge length) and will add an item to or remove an item from a container. Use this property if both the container and the content items are equipped with active tags. Each item can be tracked individually. This is also prerequisite for the escorting and association rule.</p> <p>Notes:</p> <ul style="list-style-type: none"> • If you have a lot of container items, this setting will have an impact on performance. Enable this option only if you use this kind of dynamic container assignment often, or if you have defined escorting rules. • When the content item is not seen by the RTLS system, it will not inherit the container position but will stay on the last seen position. To force inheritance of the container position, see ContainerSupportOn. • This property should not be selected at the same time as ContainerSupportOn. If it is, the ContainerSupportOn property will be ignored.

Table 12. Default system properties (continued)

Name	Type	Default	Description
HierarchicalGroups	Boolean	The box is checked, which means Yes	Specifies whether or not to use hierarchical groups. The box can either be checked to mean Yes or not checked to mean No. Note: When you uncheck this box to set this value to No, specifying that you do not want to use hierarchical groups, you cannot switch back to Yes. Also, once you uncheck the box, you may have to wait up to one minute for the changes in the setting to take effect.
InactivityDelay	Integer	60	Time in seconds that Location Awareness Services for WebSphere Sensor Events waits before displaying a tag as inactive in the Spatial Management Client if no new position coordinates are received. Note: The value specified for this property will be added to the value specified for the WatchdogDelay property.
IsStationaryRuleAlertAction	String	None	Defines whether an event is generated if an item that is defined as stationary is moving. Valid values are: <ul style="list-style-type: none"> • Event • Display • Event, Display • Display, Event <p>If a class that is defined as stationary moves twice the amount of feet defined in MaxToleratedMovement, an event is generated once every 5 minutes to avoid flooding stationary events when an item is moving.</p> <p>An event could be similar to the following example: Item with tag [00000017] with label [alabelele], defined as stationary has moved at [Tue Jan 22 22:56:56 CET 2008]. Details: Position [38.30, 38.30, 0.00], Classes: [Asset], Groups: [Laptop]</p>
LasDirectory	String	C://IBMHttpServer//htdocs//en_US	Directory that holds subdirectories, such as archive, search, and SVGs for maps, that are the default values for Location Awareness Services for WebSphere Sensor Events. The subdirectories must contain the complete, running Spatial Management Client, and the specified directory must already exist. Restriction: Do not change the value of this property.
LogHistory	Boolean	The box is not checked, which means No	Specifies whether to save runtime data for the tags. The box can either be checked to mean Yes or not checked to mean No.

Table 12. Default system properties (continued)

Name	Type	Default	Description
LogJSR168Default	Boolean	The box is checked, which means True	<p>Specifies how the logging target for portlet logging. The box can either be checked to mean True or not checked to mean False.</p> <p>The default value is set to True, which means that some logging may go to the SystemOut.log file. Setting the value to False, enforces all logging to use log4j.</p>
Max connection retry	Integer	20	<p>Defines the number of retries before a socket connection to an event provider is lost. A retry is attempted every 3 seconds. The default value is 20. If you specify -1 or if the parameter does not exist, Location Awareness Services for WebSphere Sensor Events tries reconnecting forever. This parameter is valid for all socket connections.</p>
MaxTagSignalAge	Integer	10	<p>Specifies a time frame, in seconds, for resetting the tag smoothing algorithm.</p> <p>With smoothing, several location events are added and new location coordinates, X,Y and Z, are calculated. Typically every 1 or 2 seconds the systems receives a tag location event, but under specific circumstances, such as bad coverage or shadows, the read times between the location events can be extended. This parameter specifies the time to reset the tag smoothing algorithm in order to drop old location events.</p> <p>For example, if this property is set to 10 seconds, and the smoothing algorithm is weighted average with a time series length of 5, then location events 1 through 5 are gathered every second. For every location event the smoothing algorithm calculates the weighted average based on the current amount of gathered events. For location event 2 the algorithm takes location events 1 and 2, for location event 3 it takes events 1, 2 and 3, and so on. If location event 6 takes 11 seconds to be sent to the system, then the smoothing algorithm is reset and location event 6 is processed the same as location event 1.</p>
MaxToleratedMovement	Integer	2	<p>Number of feet an item can move without generating an alert when belonging to a class that is defined as stationary.</p> <p>To avoid a flooding condition of stationary alerts, if an item moves twice the amount of units defined, then a stationary alert will be generated once every 5 minutes.</p>

Table 12. Default system properties (continued)

Name	Type	Default	Description
MaxUnrecognizedMovement	Integer	1	<p>Number of feet an item can move before it is identified as moving. Movements within the specified number of feet are not reported.</p> <p>This property also affects when a tag is identified as having entered or exited a zone. A tag is considered to have entered or exited a zone if its coordinates are inside or outside of the zone by at least 50 percent of this value. For example, if this value is set to 1 and a tag's coordinates indicate that the tag is inside the zone by at least six inches, the tag is considered to have entered the zone.</p>
MinRefreshInterval	Integer	10	<p>Time in seconds after which the tag position is updated. At least every <i>n</i> seconds (as specified with this parameter), the tag position is updated. If you receive updates in shorter intervals, they are used and the tag position is updated more often.</p>
MissingReadsTolerance	Integer	30	<p>Time in seconds during which missing readings from tags are tolerated.</p> <p>The value specified for this property will be added to the value specified for the WatchdogDelay property and a no TagNotResponsiveAlert is generated.</p> <p>The system checks regularly for unresponsive tags, every <i>value of WatchdogDelay</i> seconds. Also, the system is looking for tags that have not been seen in the last <i>value of MissingReadsTolerance</i> seconds. So, it may take up to <i>value of WatchdogDelay</i> plus <i>value of MissingReadsTolerance</i> seconds until this condition is detected. If this function is critical, the value of WatchdogDelay should be smaller than that of MissingReadsTolerance.</p>
NumOfBadMsgIgnored	Integer	10	<p>Time in seconds after which low quality messages are ignored when they follow good quality messages.</p>
NumberOfEventsPerTag	Boolean	50	<p>Defines how many events are listed in the tag details window on the Spatial Management Client.</p>

Table 12. Default system properties (continued)

Name	Type	Default	Description
RTLSProviderAlertAction	String	CEI	<p>Specifies the behavior for diagnostic events coming from event providers. Accepted values are: CEI, CEP, None, Other</p> <ul style="list-style-type: none"> • CEI - The events are routed as is to CEI and show up as diagnostic events, if they are not customized. For example, tagNotResponsive or battery-related events. • CEP - The diagnostic events are routed to the rules engine. This value is only valid if rules for diagnostic events are deployed. • None - The diagnostic events are ignored. This also applies to the rules that depend on diagnostic events such as tagNotResponsive or tagNotMove. For example, the Man Down Detection rule would not work. • Other - All diagnostic events are ignored except for tagNotResponsive or tagNotMove, so the alerts related to those events will function.
RunPerformanceTest	Boolean	The box is not checked, which means No	Specifies whether to check performance options. Use this property when debugging. The box can either be checked to mean Yes or not checked to mean No.
SimulatorFileDirectory	String	C:/IBMAAtlas/Simulator/	<p>Target directory for files with recorded data.</p> <p>This property is available for event providers that are connected with a LAS Socket connection.</p>
SimulatorFileExtension	String	.txt	<p>The file extension of the files with recorded data. A timestamp is used in the file names.</p> <p>This property is available for event providers that are connected with a LAS Socket connection.</p>
SimulatorFileLength	Integer	20	<p>The length defines the size of the simulator file in kilobytes (KB). When the defined size is exceeded, a new simulator file is created with a new suffix. This value can be set to no more than 100.</p> <p>This property is available for event providers that are connected with a LAS Socket connection.</p>
SimulatorFileSwitchInterval	Integer	10800000	<p>Time in milliseconds after which Location Awareness Services for WebSphere Sensor Events switches to a new output file.</p> <p>This property is available for event providers that are connected with a LAS Socket connection.</p>

Table 12. Default system properties (continued)

Name	Type	Default	Description
SimulatorRecordingOn	Boolean	The box is not checked, which means No	Turns recording on or off. The box can either be checked to mean Yes to turn the recording on, or not checked to mean No and recording is not turned on. This property is available for event providers that are connected with a LAS Socket connection.
TagNotResponsiveAlertAction	String	Event	Alert action if a tag is not responsive. The alert generates an event. Valid values are: <ul style="list-style-type: none"> • Event • Display • Event, Display • Display, Event In the Spatial Management Client, the tag icon fades.
UnknownIconLabel	String	Unknown Tag	Label of the unknown tags.
UnknownTagAlertAction	String	Event	Alert action if an unknown tag is found. The alert generates an event. Valid values are: <ul style="list-style-type: none"> • Event • Display • Event, Display • Display, Event In the Spatial Management Client, the unknown tag icon displays.
UnknownTagIcon	String	unknownTag.jpg	Graphical representation of the unknown tags.
WASBootstrapAddress	String	localhost 2809	Defines the bootstrap address for WebSphere Application Server if it is different than the default (such as 2810 in case of multiple servers on the same machine). The bootstrap address is used to retrieve and send alert events.
WatchdogDelay	Integer	60	Time in seconds during which non-zone-related business rules are checked. For example, if this value is set to 60, then tags will be checked every 60 seconds, whether they are responsive or not. Note: If another property that delays checking is set, then action will not be taken on a tag until after the accumulation of delays.

Formatting data types for importing data to Location Awareness Services for WebSphere Sensor Events

This section provides information about data types and values for importing data to Location Awareness Services for WebSphere Sensor Events.

Table 13. Data Types and values

Name (in ClassesItemsManager)	Type	Format	Example
checkbox	boolean	true false	true, false
date	date	MM/dd/YYYY	11/20/2008
text / textarea	string	any string (must not contain "or')	John
integer	integer	any integer	-1

Importing resource data to Location Awareness Services for WebSphere Sensor Events

This topic describes how to import resource data into Location Awareness Services for WebSphere Sensor Events.

Location Awareness Services for WebSphere Sensor Events provides an application that acts as an intermediary component between an enterprise's legacy systems and Location Awareness Services for WebSphere Sensor Events to allow information about tagged items (people or assets) to be imported into Location Awareness Services for WebSphere Sensor Events and, subsequently, to be updated or deleted. The application reads records from comma-separated values (CSV) files that are provided by the existing enterprise application and forms a Java Message Service (JMS) request. The application sends the JMS request to Location Awareness Services for WebSphere Sensor Events through the messaging engine and then logs the responses in a log file.

Planning for importing

When you are importing a large amount of tags in a production or test environment, make sure that the server you are using is capable of monitoring the tags. For example, to monitor more than 200 tags, you need a server with a fast I/O subsystem.

If you are importing a large amount of tags, such as 3000, only use the create action for test scenarios or for an initial load. On a smaller server, update or delete items using the Classes/Items Manager portlet.

If you set the action in the properties file to create instead of createUpdate, then you can import using a x100 server with a standard disk drive with the following statistics:

- The AtlasIntegrator application creates 3000 tags in 5 minutes.
- The Web service creates 3000 tags in 10 minutes.
- If you use the AtlasIntegrator application with the action set to createUpdate, then creating 3000 tags takes 2 to 4 hours.

If you use the AtlasIntegrator application with the action set to createUpdate on a faster I/O subsystem, then you can import with the following statistics:

- Using a M206 server with RAID (4 disks), it takes less than 30 minutes for 3000 tags.
- Using a X3650 server with RAID (8 disks), it takes less than 6 minutes for 3000 tags.

Configuring properties

1. Configure the properties in the following properties files:

Data_Export.properties

Contains properties that identify JMS resources and the location of the CSV and class properties files. A sample is provided in `LAS_HOME\AtlasIntegrator\Data_Export.properties`. A sample CSV file and its associated properties files are also provided in the same location.

Verify that the following properties are set correctly:

- `batchsize`: Enter the size of the JMS batch. For example, a value of 50 means that AtlasIntegrator sends packets of 50 items for import.

Note: Set this number to a low value, such as 20, if you experience timeouts (for example, if you get a message saying that no response was received from WebSphere Application Server).

- `locale_language`: Specify the language corresponding to the data you wish to import. The language value should be a valid ISO language code, such as `en` for English or `de` for German.
- `locale_country`: Specify the country corresponding to the data you wish to import. The country value is a valid ISO country code. These codes are the uppercase, two-letter codes as defined by ISO-3166. For example, `US` for the United States or `DE` for Germany.
- `locale_variant`: Specify the variant. The variant value is vendor or browser-specific code. For example, use `WIN` for Windows®, `MAC` for Macintosh, and `POSIX` for POSIX. If you are unsure about the system you are using, leave this property empty.
- `CSV`: Enter the location of the CSV file.

Note: The following conventions must be followed in the CSV file:

- Each row must contain exactly one data row. You can use the new line character (`\n`) to force a new data row. If you want to include the new line character in a data cell without forcing a new row, enclose the contents of the data cell in double quotation marks (`"item1/nitem2"`).
 - The data cells of a data row must be delimited by a comma. A comma forces a new data cell. If you want to include a comma without forcing a new data cell, enclose the contents of the data cell in double quotation marks (`"item1,item2"`).
 - If you need to use double quotation marks in a data cell without indicating the beginning or end of the data cell contents, enclose the contents of the data cell in double quotation marks (`"Error: "error_message""`).
- `log`: Enter the location of the log file.
 - `hostname`: Enter the fully qualified host name or IP address of the provider of the Service Integration bus.

- **secure:** Specify whether security has been enabled for WebSphere Application Server. The default value is no.

Note: If security has been enabled for WebSphere Application Server, the `Data_Export.bat` file may need to be edited. The default configuration points to the sample key store and trust store files provided with WebSphere Application Server. If you use different key and trust stores or different passwords for these files, edit the `Data_Export.bat` file as needed. The `trace.log` file contains detailed logs about the communication process, including possible security related issues.

- **port:** Enter the `SIB_ENDPOINT_ADDRESS` of your messaging engine. The default value is 7276. If security has been enabled for WebSphere Application Server, enter the `SIB_ENDPOINT_SECURE_ADDRESS` of your messaging engine, which is usually 7286.
- **request_q:** Enter the name of the request queue, such as `AtlasImportRequestQ`.
- **response_q:** Enter the name of the response queue, such as `AtlasImportResponseQ`.
- **propertiesFileLocation:** Enter the directory that contains the `ClassMapping.properties` file and the `class_name.properties` files. Leave this property empty to specify the current directory from which the import application (`Data_Export.bat`) is running.
- **class:** Enter the column in the CSV data file that contains class names. This value must be specified in the `attribN` format, where *N* is the integer representing the column number. For example, if the class names are in column 7, then `class=attrib7`.
- **action:** Enter the action to be performed on the record being sent to Location Awareness Services for WebSphere Sensor Events. Valid values include:
 - **createUpdate:** Create a new record if the tagged item does not already exist in the Location Awareness Services for WebSphere Sensor Events database. Otherwise, update the existing record.
 - **create:** Create a new record if the tagged item does not exist in the Location Awareness Services for WebSphere Sensor Events database. Otherwise, return an error.
 - **update:** Update an existing record in the Location Awareness Services for WebSphere Sensor Events database. If the record does not exist, return an error.
 - **delete:** Delete an existing record from the Location Awareness Services for WebSphere Sensor Events database. If the record does not exist, return an error.

If you do not specify an action, the default action is `createUpdate`.

- **group:** Enter the column in the CSV data file that contains group names. This value must be specified in the `attribN` format, where *N* is the integer representing the column number. For example, if the group names are in column 8, then `group=attrib8`.

Note: If you want to retain the relationship of an item to multiple groups during the import and `HierarchicalGroups` is set to off, you can specify multiple groups names in this column, separating the group names with a pipe symbol (`|`).

- **defaultClass:** Enter the name of the class that new records from the CSV file are added to if the corresponding class name specified in the CSV file is not found in the `ClassMapping.properties` file. For example, if a record in the CSV file contains the class name `RESOURCE SECURITY` and that class name is not found in the `ClassMapping.properties` file, the record is added to the class specified in this property. For example, `defaultClass=Contractor`.
- **defaultGroup:** Enter the name of the group that new records from the CSV file are added to if group information is not specified. For example, if a record in the CSV file does not contain group information, the record is added to the group specified in this property. For example, `defaultGroup=Contractor`.
- **tagId:** Enter the column in the CSV data file that contains tag ID values. This value must be specified in the `attribN` format, where *N* is the integer representing the column number. For example, if the tag IDs are in column 13, then `tagID=attrib13`.
- **mq_response_timeout(secs):** The Location Awareness Services for WebSphere Sensor Events import client sends a JMS request containing a batch of ten records from the CSV data file to Location Awareness Services for WebSphere Sensor Events. Enter the number of seconds that the Location Awareness Services for WebSphere Sensor Events imports client waits for the JMS response. The default value for this property is 60 seconds.

ClassMapping.properties

Provides a mapping from the names in the `class` column of the CSV data file to class names that are defined within the Location Awareness Services for WebSphere Sensor Events database. See “Planning for classes and items” on page 45 for tips on defining the Location Awareness Services for WebSphere Sensor Events class hierarchy. For example, a `ClassMapping.properties` file might read as follows:

```
ACME INC.=Employee
Sunspot Heating and Cooling=Contractor
```

This file indicates that the records with `ACME INC.` in the `class` column are to be assigned to the Location Awareness Services for WebSphere Sensor Events class `Employee` and those records with `Sunspot Heating and Cooling` are to be assigned to the Location Awareness Services for WebSphere Sensor Events class `Contractor`.

class_name.properties

Provides the attribute details about any class. A sample is provided in `LAS_HOME\AtlasIntegrator\Person.properties`. The file name of the class properties file should be the class name. There should be one file for each Location Awareness Services for WebSphere Sensor Events class.

Verify that the following properties are set correctly:

- **label:** Enter the attributes and text strings, separated by a plus sign (+), that automatically fill the tag label field and surround blank spaces with quotation marks. For example, `attrib5+" "+attrib6+" "+attrib4`.
- **icon:** Enter the attributes and text strings, separated by a plus sign (+), that represent the name of the graphic file that will represent the class items in the Spatial Management Client. For example, if the value of `attrib3` is `Susan`, which represents a specific item in the

Person class, the following entry will equate to Susan.png:
attrib3+".png". Surround extensions with quotation marks.

- **attribN**: Enter the name of an attribute and its corresponding Location Awareness Services for WebSphere Sensor Events property name, where *N* corresponds to the column in the CSV file that contains the information. For example, attrib2=First Name indicates that column 2 in the CSV file contains the first name of the item and is mapped to the Location Awareness Services for WebSphere Sensor Events property named First Name.
 - **KeyProperties**: Enter the list of attributes, separated by commas, that represents key properties. For example, attrib5,attrib3.
2. Run the data import application from the *LAS_HOME* directory, specifying your messaging engine user ID and password:

```
Data_Export.bat user_ID password [Data_Export.properties ClassMapping.properties]
```

Tips:

- Because the *Data_Export.properties* and *ClassMapping.properties* files are entered as parameters to the import application, you can replace these file names of these properties with names that are more meaningful to you. This allows you to set up a series of properties files with different names that reflect different tasks or mappings. For example, you could distinguish between the initial import of enterprise data and later maintenance imports.
 - This command assumes that the import application is running from the *AtlasIntegrator* directory. If the *Data_Export.properties* file is not in the same location as the import application, provide the complete directory path. For example:

```
Data_Export.bat user_ID password D:\Properties\Data_Export.properties  
ClassMapping.properties
```
 - If the remaining properties files, such as *ClassMapping.properties* and *class_name.properties*, are not in the same location as the import application, the directory location can be specified in the *propertiesFileLocation* in the *Data_Export.properties* file. For example, if the *ClassMapping.properties* file and the *class_name.properties* files are located in *D:\Properties*, set *propertiesFileLocation=D:\\Properties*. If the *ClassMapping.properties* file and the *class_name.properties* file are located in the same directory as the *Data_Export* application, leave the value for *propertiesFileLocation* empty: *propertiesFileLocation=*.
3. Look at the log file specified in *Data_Export.properties* file to verify that the import application ran successfully.

Note: Look at the *trace.log* file if you are experiencing connection problems.

Using the dispatcher

This topic describes the dispatcher application and how to utilize it.

The dispatcher is a standalone application that acts as an intermediary between large Location Awareness Services for WebSphere Sensor Events event providers, such as hubs that process more than 300 messages per second, and one or more devices. The dispatcher retrieves all location messages from the event providers it is connected to and distributes them to one or more Location Awareness Services

for WebSphere Sensor Events devices. Using the dispatcher enables Location Awareness Services for WebSphere Sensor Events to increase the number of location messages it processes.

The dispatcher is shipped with Location Awareness Services for WebSphere Sensor Events and is located in the `LAS_HOME\samples\AtlasStandaloneDispatcher` directory.

Communicating with event providers

The dispatcher connects to the event providers as a socket server using the IP address and port that are specified in the `dispatcher.bat` file. When the dispatcher establishes a connection, it receives all location messages sent by the provider and distributes them to the connected devices. Each device receives a subset of the location messages from the event provider.

If the dispatcher cannot connect, it tries again every 30 seconds. If an existing connection to an event provider drops, the dispatcher tries to reconnect in time intervals increasing from one to 30 seconds.

Communicating with event devices

The dispatcher communicates with a device as if it were an event provider. The dispatcher waits for a device, which is a client application to the dispatcher, to connect. You specify the number of clients and associated ports that can connect to the dispatcher in the `dispatcher.bat` file. Each client must use an individual port.

When the dispatcher connects to a device, it forwards all location messages assigned to that device. All messages have the same format and content as when they are received from the event providers.

How location messages are assigned to a device

The dispatcher supports two simple algorithms for assigning location messages to a device: modulo (the suggested algorithm) and round robin. You specify the algorithm that you want to use in the `Dispatcher.bat` file.

- **Modulo** - This algorithm uses the last digit or letter of a location message's tag ID. The number of active devices determines which device the message is assigned to.

If there are no active Location Awareness Services for WebSphere Sensor Events devices, the dispatcher discards the location messages from the provider. The dispatcher considers only active devices, and forwards all messages arriving from the event providers to them. If a device cannot keep up with the number of messages provided, the dispatcher queues the outstanding messages.

- **Round robin** - Using this algorithm, if *N* Location Awareness Services for WebSphere Sensor Events devices are connected to the dispatcher, each gets every *N*th location message.
- **Hash map** - Setting this algorithm means that the set of available tag IDs is evenly distributed to the different devices and that messages referring to the same tag ID always go to the same device.

Note: In a production environment, use the modulo or the hash map algorithm. These dispatching algorithms work better with filtering and position smoothing within Location Awareness Services for WebSphere Sensor

Events. You might use the round robin algorithm in a test environment where you use fewer tags to generate test data.

Configuring the dispatcher application

This topic describes how to configure the dispatcher application.

Before you begin

Before using the standalone dispatcher application, make sure that each event provider is defined in the Event Provider portlet.

About this task

This topic explains how to configure Location Awareness Services for WebSphere Sensor Events ports to use the standalone dispatcher. When using Location Awareness Services for WebSphere Sensor Events with the standalone dispatcher, not directly connected to the event provider, you must configure a provider definition for each Location Awareness Services for WebSphere Sensor Events port. These definitions must be identical except for the provider port number, which varies as specified for the dispatcher. All definitions must point to the same provider IP address, and the same areas must be assigned to all provider ports. For more information about the dispatcher, refer to “Using the dispatcher” on page 74.

1. Copy the contents of the `LAS_HOME\samples\AtlasStandaloneDispatcher` directory to a separate directory on your system.
2. Make a backup of the sample `Dispatcher.bat` file.
3. Edit the `Dispatcher.bat` file, providing the following parameters for each event provider:
 - `TagIDPosition` - Specify the position of the tag ID in the input event. The default is 3.
 - `Separator` - This is the separator between units of information in the input event. The default is ; (a semi-colon).
 - `HubIP` - Specify the IP address of the machine hosting the event provider.
 - `HubPort` - Specify the port number that the event provider listens on. Typically, the port is 5117.
 - `AtlasPorts` - Specify a list of the ports that the dispatcher listens to. Separate each port number with a comma.
 - `Logging` - This parameter is optional. Specify on to enable logging or off to disable logging. By default, this parameter is set to off. If logging is enabled, the output is logged in the `SysOut` file. Only enable logging for debugging purposes.
 - `Algorithm` - This parameter is optional. Specify the dispatching algorithm that you want to use. By default, this parameter is set to `MODULO`.

Note: Look over the parameters to ensure that they comply with these guidelines:

- Parameter keywords and the predefined values are case-sensitive.
- Keywords and values cannot contain any blank spaces.

If you start the dispatcher without entering any parameters, a usage message is displayed. You can also enter `?`, `help`, or `h` to display the usage statement. If you enter incomplete or erroneous parameters, you receive an error message.

4. Start the dispatcher application by running the Dispatcher.bat file from a command line.
5. Stop the dispatcher by typing stop or s from a command line.

Backing up and restoring data

This topic describes how you back up historical and event data for Location Awareness Services for WebSphere Sensor Events.

About this task

You are responsible to back up historical and event data for Location Awareness Services for WebSphere Sensor Events. This data is stored in the Location Awareness Services for WebSphere Sensor Events databases and the CEI events database. You can use the database management system (DBMS) to automatically schedule backups, archive, and delete tasks or you can manually back up these databases.

Go online to see the DB2® information center for more information.

Note: Once you make a backup you can replay the transaction logs that were used during the online backup. The data stored in the backup image will be replayed. However, if you wish to replay transactions after you back up databases, ensure that your log files are in the path where they were archived and that the subdirectories are in the data directory, such as in C:\DB2_Archived_Logs\DB2\ATLASDB\NODE0000.

Backing up your databases while online Before you begin

Prior to backing up your system, make sure that:

- The DB2 backup batch files are in the same directory and that the log files are also created in the directory where the batch files are located. The DB2 scripts are shipped with Location Awareness Services for WebSphere Sensor Events in the *LAS_HOME*\DB2\Backup Restore directory.
- The following directories exist:
 - C:\DB2_Database_Backups: Directory where the database backups are stored
 - C:\DB2_Archived_Logs: Directory where DB2 stores the archived log files
 - C:\DB2_LOGTEMP: Directory where temporary log files are created during the roll forward operation of logs stored in the database backup image

About this task

The following instructions describe how to back up your databases when you are online, which allows you to be connected to the database. To perform a backup, do the following:

1. Set up the environment by running the DoSetupBackup.bat script. It calls the setupbackup.bat script.
2. Schedule online backups of the ATLASDB and EVENT databases using the Windows at command and the DoONLINEBackups_Runstats.bat command from the C:\DB2_Backup_Scripts directory:

Note: The DoONLINEBackups_Runstats.bat command also updates statistics for the tables in both databases after the backups are done. The script can

also be run manually from a command prompt in the directory where the file is located and calls the following scripts:

- OnlineBackup_Databases.bat: Creates an offline backup of the ATLASDB and EVENT databases
- ATLASDB_RUNSTATS.bat: Updates the statistics for the ATLASDB database
- EventDB_RUNSTATS.bat: Updates the statistics for the EVENT database

- a. To run a weekly backup at 00:30 on Sunday morning, run the following command:

```
at 00:30 /every:Sunday "C:\DB2_BACKUP_SCRIPTS\DoOnlineBackups_Runstats.bat"
```

- b. To run a backup every day at 00:30, run the following command (on one line):

```
C:\DB2_Backup_Scripts>at 00:30 /every:Sunday,Monday,Tuesday,Wednesday,Thursday,Friday,Saturday "C:\DB2_BACKUP_SCRIPTS\DoOnlineBackups_Runstats.bat"
```

- c. To delete an entry from the at list, run the following commands:

- 1) Run at to list the scheduled tasks:

```
at
```

- 2) Use the ID from the list to delete the entry:

```
at task_ID /DELETE
```

- 3) Run at again to verify that there are no entries in the list:

```
at
```

Backing up your databases while offline

Before you begin

Prior to backing up your system, ensure that:

- All backup scripts are located in the C:\DB2_Backup_Scripts directory. The scripts for offline backups are shipped with ATLAS in the *LAS_HOME*\DB2\Backup Restore\OFFLINE (cold) backups directory.
- No users are active in the system and that WebSphere Application Server is not running.
- The following directories exist:
 - C:\DB2_Database_Backups: Directory where the database backups are stored

About this task

The following instructions describe how to back up your databases when you are offline, which means that users cannot be connected to the database.

Note: To turn off log archiving and allow only offline backups, run the DoTurnOffArchiving.bat script.

To make a backup, do the following:

Schedule offline backups of the ATLASDB and EVENT databases using the Windows at command and the DoBackups_Runstats.bat command from the C:\DB2_Backup_Scripts directory:

Note: The DoBackups_Runstats.bat command also updates statistics for the tables in both databases after the backups are done. The script can also be run manually from a command prompt in the directory where the file is located and calls the following scripts:

- Backup_Databases.bat: Creates an online backup of the ATLASDB and EVENT databases
 - ATLASDB_RUNSTATS.bat: Updates the statistics for the ATLASDB database
 - EventDB_RUNSTATS.bat: Updates the statistics for the EVENT database
1. To run a backup weekly at 00:30 on Sunday morning, run the following command:
`at 00:30 /every:Sunday "C:\DB2_BACKUP_SCRIPTS\DoBackups_Runstats.bat"`
 2. To run a backup every day at 00:30, run the following command (on one line):
`C:\DB2_Backup_Scripts>at 00:30 /every:Sunday,Monday,Tuesday,Wednesday,Thursday, Friday,Saturday "C:\DB2_BACKUP_SCRIPTS\DoBackups_Runstats.bat"`
 3. To delete an entry from the at list, run the following commands:
 - a. Run at to list the scheduled tasks:
`at`
 - b. Use the ID from the list to delete the entry:
`at task_ID /DELETE`
 - c. Run at again to verify there are no entries in the list:
`at`

Restoring databases

Before you begin

Prior to running the script, verify the following:

- The database you are restoring was dropped.
- There is only one backup image file in the path of the database backup. For example, the ATLASDB database should only have one data directory with one backup file in that directory, such as C:\DB2_Database_Backups\ATLASDB.0\DB2\NODE0000\CATN0000\20060330.

About this task

To restore a single database, run the DoRestoreOneDB.bat script from a command line, which calls the RestoreToNewDatabase.bat script.

Scheduling deletion of tag data

This topic describes how to set up a regular schedule for deleting historical tag data.

About this task

Currently Location Awareness Services for WebSphere Sensor Events does not automatically archive historical data. Therefore, it is necessary to delete historical data on a regular basis to increase performance. You can schedule tag data to be deleted on a regular basis using the ATLASDB_DEL_TAG_HISTORY_SCHEDULE.bat script. The script schedules a task to delete all historical data entries that are older than eight hours and for which newer data is available. This allows you to consistently delete historical data.

Make sure you plan carefully for the deletion of historical tags and events. For example, if you have a lot of events in the CEI database within a short period of time, this can lead to performance degradation. You should also use the Web service to delete and archive CEI events as the notification program for the automatic clean up of those events, since they do not need to be kept.

To retain historical data for a longer period of time, you must archive and back up your data separately.

Complete the following steps to set up a regular deletion process for historical data:

1. Create a directory named `C:\tools\history` and copy all files from the `LAS_HOMEDB2\Tools\History` to the new directory. To create the directory in a different location, edit `ATLASDB_DEL_TAG_HISTORY_SCHEDULE.bat` with the new location.
2. Edit `ATLASDB_DEL_TAG_HISTORY.bat` to specify your DB2 installation directory.
3. Edit `ATLASDB_DEL_TAG_HISTORY_SCHEDULE.bat` to enter your DB2 Workgroup Server Edition user ID and password.
4. (Optional) This script deletes all historical data entries that are older than eight hours and for which newer data is available. If you want to change the schedule, for example to delete data every six hours, edit the 8 hour entry in the insert and the delete statements.
5. Run `ATLASDB_DEL_TAG_HISTORY_SCHEDULE.bat` to schedule the deletion task.

Note: After running this script, you must either manually edit each task to change the schedule or you must delete the scheduled tasks and then rerun this script after editing it.

Each time the deletion task runs, it will log results into the `C:\tools\history\ATLASDB_DEL_TAG_HISTORY_DATA.txt` file.

Tracking replay accounts

This topic lists the portlet you use to monitor all configured replay accounts.

Use the following portlet to release, reload, and see an overview of replay accounts:

- “Replay Accounts Administration”

Log in to the WebSphere Application Server administrative console and click **Replay Accounts Administration** to access this page.

Replay Accounts Administration

Use this page to see an overview of all configured replay accounts, along with their status. You can also use this page to release accounts that have not been properly released by their owners.

Select an account and click **Release** to release the account and make it available to a new user.

Click **Reload** to refresh the entire list of accounts, such as when accounts have been used or released.

- **ID** is a read-only value from 1 to 5.
- **Status** can either be **FREE** or **DATA READY**. **FREE** means that the replay account is open. **DATA READY** means that the account is occupied.
- **Owner** is the name of the user that is logged into the Spatial Management Client. If you do not have WebSphere Application Server security enabled, then the Owner could be any logged-in user ID. If WebSphere Application Server security is enabled, then the user ID is defined in the local operating system. You cannot create new IDs or delete existing IDs in this portlet.
- **Requested at** is the date and time at which the account was requested by a client. For example, the date and time at which a replay run was started.

- **Last Client Refresh at** is the date and time at which the last replay data update was delivered to the client.

Operating

Use the Spatial Management Client and the Location Awareness Services for WebSphere Sensor Events Administrative Console to perform daily operation tasks for Location Awareness Services for WebSphere Sensor Events.

Starting the Spatial Management Client

This topic describes how to start the Spatial Management Client.

About this task

Start the Spatial Management Client by completing the following steps:

1. Start the Spatial Management Client by typing the following URL in a browser: `http://fully_qualified_host_name/Tracking GUI/AtlasMonitor.html` If you are an administrator, see “Starting the Spatial Management Client (administration)” on page 30.
2. Enter your user name, and password if security is enabled, and click **OK**. Your individual preferences are displayed. You can save your preferences for each area you view by clicking **Save** under **DEFAULT VIEW**. Setting preferences prevents rescaling and repositioning each time you view an area of interest.
3. In **AREA**, select the area that you want to monitor from the drop-down list.
4. In **TAGS**, select the class of tags that you want to monitor. For example, select **Asset** to view all hardware that has been defined or **All** to view all tags.
5. In **ZONES**, under **Visible**, select the category of zones that you want to view.
6. In **ALERTS**, turn the alert sound on or off and choose whether to hide or view all alerts. You can also click **Acknowledge All Alerts** to acknowledge and turn off all current alerts.
7. In **DEFAULT VIEW**, click **Save** to save the current pan and zoom settings. You can customize the view and scale of the area without having to repeat the process every time you log in to the Spatial Management Client.

The **OVERVIEW** window provides a view of the entire area. Drag the blue box around the overview window and notice that the main graphic window of the Spatial Management Client reflects the highlighted area. The box size is controlled by the zoom slider and zoom box controls below. The upper-left corner of the blue box and the upper-left corner of the main graphic window are the same point.

See “Spatial Management Client” on page 82 for more information.

8. To start monitoring tags in the GUI, start the tag processing servlet:
 - a. Open the WebSphere Application Server administrative console and click **Control Processing**.
 - b. Select the WebSphere Application Server that is related to your installation and click **Start Selected**.

If you do not start the tag processing servlet, tags are displayed at their last reported location.

Results

In the Spatial Management Client, the defined tags are displayed with the icons you define, either for the item or the class. These icons move on the Spatial

Management Client according to the reported coordinates. If you turn alerts on, a red circle highlights the tag icon when an alert related to the tag is reported. You can click the icon and see the alert details and acknowledge the alert. The circle goes away when you acknowledge the alert.

In some cases the tags fade, which means that there is no current position information available about the tag. Location Awareness Services for WebSphere Sensor Events assumes that the tag remains at the last reported position. Use the `InactivityDelay` system property to set the length of time after which a tag starts to fade. To avoid moving tags away from the last reported position, set this parameter to a high value. See “System Properties” on page 62 for a complete list of system properties.

Spatial Management Client

This topic describes the Spatial Management Client.

The Spatial Management Client provides a state of the art visual interface which shows the location of tags in real time, allowing an authorized user to monitor employees, contractors, and visitors in hazardous areas, to respond immediately to emergencies, and to locate high-value assets.

Notes: For optimal GUI performance:

- Use only Internet Explorer 6.0 with the Adobe Scalable Vector Graphics (SVG) Viewer for your browser.
- Maximize the Spatial Management Client for the best results.
- Restart the GUI whenever you change the screen resolution.
- Do not use browser functions. For example, using the **Back**, **Forward**, and **Refresh** buttons in the browser can lead to inconsistent displays of areas, tags, and menu options.

For keyboard accessibility of the GUI, edit the Mouse settings.

1. Select **Start** → **Settings** → **Control Panel** → **Accessibility Options**.
2. On the Mouse tab, select **Use MouseKeys**. This option allows you to control the pointer with the numeric keypad on your keyboard.

The Spatial Management Client retrieves all tags for an area in the following cases:

- When an area is opened
- When the class filter is changed
- Every n polling intervals. The value of n is set according to the `<DisplayRefreshCounter>` parameter in the `prefsV3.xml` file. If this parameter is not present in the `prefsV3.xml` file or it is set to 0, then a full redraw is not scheduled on the Spatial Management Client.

In all other cases, tags are only refreshed when they change their position or they change their alert state.

If you experience problems with the Spatial Management Client, refer to the troubleshooting tips in the product documentation for possible solutions.

- **AREA**

Select the area that you want to monitor from the drop-down list.

- **TAGS**

Select the class of tags that you want to monitor. For example, select **Asset** to view all hardware that has been defined.

- **ZONES**

Visible: Select the category of zones to view.

- **ALERTS**

Sound: Turn the alert sound **On** or **Off**.

Hide: To view all alerts, select **No**. To hide all alerts, select **Yes**.

Tag filter: Filter the tags displayed. The options are **2d/2d**, **p-data**, **inactive**, **alerts only**, and **all**.

Note: These tag filters do not affect the zone or area tag count. They only affect what you can see on the map. For example, if there are three tags in zone Z and one of them has no accurate location information (it has only proximity data) and you filter the tags by p-data, only one tag remains visible on the map, but the tag count for zone Z still shows 3. When you set a filter, an error message appears to remind you of this discrepancy in the tag count.

- **DEFAULT VIEW**

Click **Save** to save the current scaling, positioning, and menu settings to your user preferences. You can customize the view and scale of the drawing without having to repeat the process every time you start the Spatial Management Client and log in with your user ID.

When you click **Save**, the values for the currently selected area are saved. You can press this button for each area. Then, when you switch to an area that has a saved value, the saved setting information is used. The area where you last pressed **Save** is the area that is shown first when you access the interface.

You can also save selected tag labels using the **Save** function. These saved tag labels do not have to be area-specific.

- **Draw Trajectory**

Click **Draw Trajectory** to enable the display of a tag's trajectory for certain time period.

- A start time and an end time are required.
- Fill in the value for either the TagId or IconLabel. At least one of these values must be filled. If both are set, then the TagId is used. If both are set, the tagid is used).
- **Number of Points** - This value depends on the load of your system and network because the number of hops can be limited. The recommended default value is 2000 tag hops, but you should be sure to configure this value to accurately apply to the quantity of your data. If there are more hops in the selected area in the timeframe given, only every *n*th point is displayed, so that it fits in the specified number. For example, if you have 9000 entries in your database and specify the **Number of Points** value as 2000, then you would see every fifth hop in the tag. This means that you could lose information since the display is truncated to every fifth hop, instead of more frequently, such as every second hop. You will be informed of this truncation of information.
- **Timestamp-Interval** - the value you specify produces a timestamp at the position of the tag every *n* points. For example, if you draw 100 points and you specify 10 for the timestamp interval, at every 10th point a timestamp will be written, equalling a total of 10 timestamps.

- **OVERVIEW**

This window provides a view of the entire area. Click and then drag the highlighted area in the overview window and notice that the main graphic window of the Spatial Management Client reflects the highlighted area. The box

size is controlled by the zoom slider and zoom box controls below. The upper-left corner of the box and the upper-left corner of the main graphic window are the same point.

- **Cluster view**

When several tags are close together and cannot be distinguished from each other, the cluster icon displays to indicate that there are several icons overlaying each other. Icons might overlay because the icons are too large, the current zoom is not close enough, or the tags are reported to have the same coordinates. To correct overlaying tags, try one of the following:

- Downscale the size of the icons until they do not overlay each other.
To configure the size of the icons that display in the cluster view of the main window:
 - Press the Ctrl key plus the space bar to display the Tag Zoom Control window. Then click **Up** to enlarge the icons or **Down** to shrink the icons. Icons resize immediately.
 - To close the dialog window, close the window or press the Ctrl key plus the space bar again. You can save the configured icon size with your user preferences.
- Zoom closer to the icons until you can distinguish them from each other.
- Click the cluster icon to get a list of icons within the cluster. A window opens to display all the icons of the cluster and the information for each tag according to the current configuration (for example, labels, X and Y coordinates, and alerts). To see more information about a tag, click the appropriate icon and the information appears in the detail view while the cluster view window closes.

- **Zoom selection rectangle**

Click on the dotted rectangle (zoom selection rectangle) and move the pointer to the main graphic window where you can click and drag to create a zoom selection rectangle. When you release the mouse, the window zooms into the selected area.

- **Zoom slider**

Use the slider to enlarge or shrink the current image in the main window. You can drag the slider button, click on the hashed lines, or click the magnifying glass icon to change the zoom.

Note: When you have highly magnified an area, the blue box in the overview window might not be able to represent the area and it becomes a small black rectangle and no longer zooms. You can still drag the box to pan another area.

- **Current tag count by zone/area**

The count table is a draggable window (click and press Shift to drag) that provides a list of areas, subareas, and zones and the number of tags currently in them. Only those zones that match the type of zones set to visible in the **ZONES** drop down menu are displayed.

- Click the area or zone name to display a current tag count window that lists the number of tags in the area and zone. All subareas and zones are listed under the area with which they are associated.
- When the tag count window is open, you also see a button for **Automatic Refresh On/Off**. By default the automatic refresh is on. The poll interval parameter is used for this refresh cycle. The tag movements in this window are independent of the ones in the main window, so there can be differences between how the tag movements are displayed.

- Click **Hide** to hide the area or zone or **Show** to display the area or zone on the main window.

Note: Only content filters, such as filtering for all the tags in the Person class, affect the zone count. Technical filters for details about the tags (2d/3d, p-data, and so on) apply to the visibility of the tags on the map, but they do not affect the zone counts.

- **Area List View**

Click this button to open a new window that displays the zones within the selected area and the number of tags within the zones. Click a zone to expand details about the tags within the zone. You can open multiple area list views at one time.

The tags shown within the zones will be filtered based on any search criteria you specify in the main view and you can click the pause button in the area list view to pause and view the tag information at a specific instance. You can also open the area list view when you are replaying data.

If you want to view an area list view for another area, you must open another instance of the Spatial Management Client.

Note: Filtering for a single tag does not apply to the area list view.

- **Search**

Click this button to search by class, group, or tag properties, or a combination of them.

Note: The search is *not* case sensitive. Also, the **AND** and **OR** only apply to the filter attributes within the class, group, and tag criteria. When you search by a combination of class, group, and tag criteria, they are always combined by **AND**.

- **Class Properties**

Select a class or classes to search for. Enter your search criteria. Click **AND** if all the search criteria you enter must be found or click **OR** to display search results for any search criteria that you enter.

- **Group Properties**

Select the group to search for. Click **AND** if all the search criteria you enter must be found or click **OR** to display search results for any search criteria that you enter.

- **Tag Properties**

To search for a specific tag, click **Tag** and enter the search criteria. Click **AND** if all the search criteria you enter must be found or click **OR** to display search results for any search criteria that you enter.

Click **Search** to search for the specified criteria or click **Reset** to clear all entries and perform a new search. Search results are displayed in a table or list format. When you select a tag in the table or list, the tag will be highlighted by a circle in the Spatial Management Client. If the tag is located in a different area, the area will open in the Spatial Management Client. Click **Save** to save the results to a file or close the window to exit without saving.

- **Replay**

Click this button to replay tag movements and events that occurred during a specific time frame.

A window displays. Enter the start and end date and time for the period of time you want to replay and click **Enable Replay Console**.

Select the area for which you want to display tag movements and events. Then click **Play** in the replay dialog to the right of the main window to watch the tag movements and events that occurred in the area during the specified time frame. Click **Pause** to pause events and **Resume** to resume playing them. Click **Exit** to close the replay dialog and to return to the current area and time.

Note: Times are relative to times on the database server. The machines that host the database server and WebSphere Application Server must be set to the same time zone.

Up to five users can use the replay function at any given time. You can track replay accounts using the Replay Accounts Administration portlet in the WebSphere Application Server administrative console.

Note: When you are using the replay function, you should see the tag count window and the tags in the area; however, for performance optimization reasons, there may be times that the number of tags visible on the screen and the number in the tag count window do not match.

- For the tags visible on the screen, only the tags seen by the location event provider after replay starts are drawn.
- For the tag count, the number of tags in a specific area or zone are counted. This count also includes tags that have entered the zone before replay starts but are not responsive after replay starts.

If you see this inconsistency in the number of tags, and you need to see the complete list of tags in an area at a specific point in time, use the **Search** or the **Show all tags** options.

- **Group Color On/Off**

Click this button to turn group color on or off. The color associated with the group in the Groups Manager portlet is seen as a colored rectangle behind the tag icon. Group color is off by default.

- **Acknowledge All Alerts**

Click this button to acknowledge and turn off all current alerts.

- **Reporting**

Click this button to see a list of defined reports that have been administered in the Reports Administration portlet. Select the **Display** link beside the report in the resulting list that you would like to view. Each report has a set of filter criteria. Click **Reload** to regenerate the list of reports.

See the Reports Operation documentation for more information.

- **Show all tags**

This option lists all tags that are currently in the area in a table similar to the **Search** results window. Selected filters for the area and tags do not apply.

Tags

For tags displayed on the Spatial Management Client, use the following features:

- **Tag Details:** Click a tag to display details about the tag including its tag ID, coordinates, and the class it belongs to. If there is an alert associated with the tag, you can acknowledge it by clicking **Acknowledge Alert**.
- **Label:** Hold down the Ctrl key and click a tag to display the Label window. Select the information to be displayed for the tag when you hover over it. For

example, select **Label** to display the label text defined for the item, select **Tag ID** to display the tag ID, or select **X, Y, or Z** to display location coordinates for the tag.

- **Select Commands:** Hold down the Alt key and click a tag to display the Select Commands window. You can select from the following commands:
 - **Delete Tag** - Removes the tag from the area, leaving the item definition untouched. This action is relevant in scenarios where tags leaving the area cannot be monitored at all times by gates or exit zones. For example, if a tag has left the area, but this has not been recognized by the event provider, you can manually clean up the area by selecting the tag and deleting it.
 - **Show this Tag only** - Filters to show only the selected tag. This is a special tag filter and cannot be saved as a preference. Changing the tag filter or changing the area will take the tag filter away.
 - **Draw Tag's Trajectory** - Starting at the point you select this, a line for the current path of the selected tag is drawn until you select **Stop Drawing Tag's Trajectory**. If you change the area and then come back to the view for the selected tag, you should still see the trajectory line. This action is only possible for one tag. When you select another tag for trajectory, the line for the previously selected tag is removed.
 - **Stop Drawing Tag's Trajectory** - Stops drawing the line for trajectory, if **Draw Tag's Trajectory** was selected. Otherwise, choosing this command has no effect on the tag.

Zones

Note: In the Spatial Management Client, the term *barrier zone* is a synonym for a *boundary zone*, which is defined in the **Boundary Zones** portlet.

For zones that are displayed on the Spatial Management Client, use the following features:

- **Zone details:** Click a zone to display details about the zone including name, function, coordinates, and number of tags in the zone.

This feature also allows following actions for a zone:

- **Hide zone:** If you select this, the zone is hidden (but tags are still displayed). To show the zone again, use the **Current tag count by zone/area** window.

Starting and stopping tag processing

This topic lists the portlet you use to start or stop the tag processing servlet.

Use the following portlet to start or stop the tag processing servlet:

- “Control Processing”

Log in to the WebSphere Application Server administrative console and click **Control Processing** to access this page.

Control Processing

Use this page to start or stop reporting tags from event providers. All event providers are displayed on this page, along with their status.

Select an event provider and click **Start Selected** to start communication with the provider or click **Stop Selected** to stop communication. If there is only one event provider in the list, the event provider is preselected by default.

Click **Refresh Status** to refresh the status of a single event provider. If the status of an event provider is listed as **Unknown**, then the status could not be calculated. Check the configuration of the specific event provider.

Restriction: The **Refresh Status** link does not return an updated status if you are using a hub simulator as your event provider.

Click **Refresh List** to refresh the entire list of event providers, such as when they have been added or deleted. This action will not refresh the status of the event providers in the list, as indicated by the status changing to "—", which means that status was not calculated.

Configure the port, URL, user ID, and password of the TagProcessingServlet by clicking the wrench symbol in the upper right corner of the portlet. The port and URL fields are mandatory and must not be empty. The user ID and password fields can be empty if the server associated with the target event provider is running without WebSphere Application Server security enabled. The changes apply to the current user only. The user ID and password requested are the user ID and the password of the current user. If those values change, the values have to be changed in this page as well.

Replaying tag movements and events

This topic explains replaying tag movements and events and how to do it.

Location Awareness Services for WebSphere Sensor Events allows you to replay tag movements and events. For example, you might use this feature to replay events that led to a recent alert or you might use it to replay an evacuation drill to identify improvements to procedures or the need for more training. Up to five users can use the replay function at any given time.

Important: Replaying tag movements and events can severely impact performance. To avoid large amounts of historical data, schedule deletion of data after n number of hours. The amount of data stored on your system depends on the number of tags tracked in Location Awareness Services for WebSphere Sensor Events and on the database system.

Replay is based on historical administrative data (such as about zones, classes, rules, and so forth) and historical runtime data (such as tag movements, zone entry events, and rule-based events). Historical data is stored in the Location Awareness Services for WebSphere Sensor Events database and is used to replay specific periods of time. For runtime data to be available for the replay, the LogHistory property must be set to Y during the time frame that will be replayed later. See "System Properties" on page 62.

You can view events that were logged during the period of time you are replaying on the Spatial Management Client or in the "CEI Events" on page 89 portlet.

For details on how to replay events, see the **Replay** section in the "Spatial Management Client" on page 82 topic. For details on tracking replay accounts, see the "Replay Accounts Administration" on page 80 topic.

Handling alerts

This topic contains information about the portlet and tools provided by Location Awareness Services for WebSphere Sensor Events for handling alerts.

Use the following portlet to acknowledge alerts:

- “CEI Events”

Log in to the WebSphere Application Server administrative console and click **Rules/Alerts** → **Cei Events** to access this page.

CEI Events

Use this page to handle events logged by Location Awareness Services for WebSphere Sensor Events.

Location Awareness Services for WebSphere Sensor Events events are logged to the event database and displayed in this portlet. Specify a filter for the events you want to view and then scroll through the events to view them.

Set filter

Complete the following fields to create a filter for events. You do not have to complete all of the fields in order to create a filter, but you should fill in the ones you find necessary to get the desired results. When specifying filter criteria, only positive matches are shown. For example, if you filter by the zone name, ABC, only events containing this zone name are shown. Events without a zone names are excluded.

Note: The filter criteria is specific to the type of event.

- **Tag ID:** Enter the tag ID of a person or asset.
- **Tag Label:** Enter the label of the tag.
- **Tag Class:** Enter a class of items to search for.
- **Tag Group:** Enter a group to search for.
- **Zone:** Enter a zone for which to display events.
- **Event Type:** Select a type of event to search for.
- **Rule Name:** Select a rule name for additional filtering, if you have several rules that lead to the same event. This menu contains most, but not all, rule names. Select **all (*)** to display events for all rules.
- **Acknowledged:** Select **All (*)** to display all events, **Acknowledged** to display only the events that have been acknowledged, or **Active** to display only the events that have not been acknowledged.
- **Event After:** Pick the date to start your search. Click **PickDate** to select the date from the calendar or **ClearDate** to clear your selection. Only events logged after this date are displayed.
- **Event Before:** Pick a date to end your search. Click **PickDate** to select the date from the calendar or **ClearDate** to clear your selection. Only events logged before this date are displayed.

Click **Set Filter** to save your settings or click **Clear Filter** to exit without saving the changes.

Display view

The events that match your filter criteria are then displayed. Scroll through the pages of events by clicking **First**, **Previous**, **Next**, and **Last**. Select all events by clicking the **X**. The maximum number of events displayed is 200. The maximum number of events displayed on each page is 10.

On this page you can do the following:

- Select **Delete** to delete the selected event or **Delete All** to delete all events. Deleting all events is intensive and can affect performance.
- Select **Archive** to save the selected event to an archive or **Archive All** to archive all events. Archiving all events is intensive and can affect performance.
- Select **Mark as acknowledged** to indicate that an event has been completed or **Acknowledge All** to acknowledge all events. Acknowledging all events is intensive and can affect performance.
- Click **Set Filter** to create a filter for events that display on this page. When there are many events, this feature enables you to display only those that interest you.
- Click **Reload** to refresh the events displayed for your filter.
- Click **Details** next to each event to display the date, type, and detailed message for the event.

Searching tags

This topic identifies the portlet and other search mechanisms that are available in Location Awareness Services for WebSphere Sensor Events for searching tags.

Use the following portlet to search for events:

- “Search Tags”

Log in to the WebSphere Application Server administrative console and click **Search** → **Search Tags** to access this page.

You can also use the search feature provided with the Spatial Management Client by opening on of the following URLs and clicking the **Search** option:

`http://fully_qualified_host_name/Tracking GUI/AtlasAdmin.html`

`http://fully_qualified_host_name/Tracking GUI/AtlasMonitor.html`

Note: You must access the search function through one of those two URLs. You cannot access the search URL directly.

You can search by class, group, or tag properties, or a combination of them. If you search by class, group, and tag properties, only those tags that match the combined search criteria are displayed.

Search Tags

Use this page to search all existing tags that are active.

You can search by class, group, or tag properties or a combination of them. If you search by class, group, and tag properties, only those tags that match the combined search criteria are displayed.

Search results are displayed in a table or list format. Click **Save** to save the results to a file or close the window to exit without saving.

If you save the results to a file, be sure that the file path ends with `\\` or `\`. For example, when saving the results, enter `C:\\Program Files\\` instead of `C:\\Program Files`.

Note: The search is *not* case sensitive. Also, the **AND** and **OR** only apply to the filter attributes within the class, group, and tag criteria. When you search by a combination of class, group, and tag criteria, they are always combined by **AND**.

Restriction: Partial searches are not supported. Search results return only an exact match for your criteria. For example, if you want to search for the last name "MacDonald", a search string such as "Mac" or "Mac%" will not find the tag.

History

Select **History** if you want to search on historical data from a specific date and time you enter. Do not select this field if you want to search on current data.

- **Date:** Click **PickDate** to select the date to search on. You can click **ClearDate** to reset the field.
- **Time:** Enter the time to search on in the format of hour, minute, and second (HH:mm:ss). Valid values for hour are 0-23 and valid values for minute and second are 0-59.

Click **Reload** to load the data for the date and time you selected. Then enter your search criteria.

Class Properties

Select **Class Properties** and select a class or classes to search for. Complete the fields, which vary by class, with your search criteria.

Click **AND relation** if all the search criteria you enter must be found. Click **OR relation** to display search results for attributes within the class criteria.

Group Properties

Select **Group Properties** and then select the group to search for.

Tag Properties

Select **Tag Properties** and complete the following fields to search by class:

- **Tag ID:** Enter the tag ID to search for.
- **Battery:** Enter the status of the battery of the tag.
- **Alert:** Select the type of alert to search for.
 - **True** means that a tag has an alert.
 - **False** means that a tag does not have an alert.
 - **Both** means that it does not matter whether the tag has an alert or not.
- **Area Name:** Select the area where the tag you are searching for is located.

Notes:

- Area names must be unique across the Location Awareness Services for WebSphere Sensor Events installation.
 - If Tag criteria are selected and Area Name is set to NONE, tags that are sending signals but are not in any area are returned. This can happen if a tag is in an area for which no zones are defined, but tag signals could still be received.
- **Icon Label:** Enter the icon label associated with the tag you are searching for.

Click **AND relation** if all the search criteria you enter must be found. Click **OR relation** to display search results for attributes within the class criteria.

Click **Search** to search for the specified criteria or click **Reset** to clear all entries and perform a new search. Click **Reload** to refresh the options and to reset the fields to their original state.

Generating reports

This topic lists the portlets you use to generate and manage reports on data within Location Awareness Services for WebSphere Sensor Events.

For reporting, the browser runs only on the server with WebSphere Application Server by default. To modify the reports URL, see the instructions in post-installation step 24.

Note: It is recommended that you use Microsoft® Internet Explorer as the browser to view reports because there can be issues when you view reports with certain versions of Mozilla Firefox.

Use the following portlets to generate and manage reports:

- **Reports Administration**
Log in to the WebSphere Application Server administrative console and click **Reports** → **Reports Administration** to access this page.
- **Reports Operation**
Log in to the WebSphere Application Server administrative console and click **Reports** → **Reports Operation** to access this page.

Reports Administration

Use this page to create and manage customized reports from data that has been collected.

Based on data that has been collected, you can create customized reports, such as: Battery life reports, Tag count by zone reports, and Area and zone list reports.

Functions that you can use to manage an existing report or add a new report include:

- **Add:** Add a new report.
- **Delete:** Delete a report.
- **Reload:** Reload the data from the database.
- **Edit:** Edit the report details.

Adding a new report

Click **Add** and then complete the following fields to create a new report.

- **Report Name*:** Enter a report name. For example, Battery life.
- **Report File Name*:** Enter a report file name. For example, BatteryLifeReport.rptdesign.
- **Report File Path*:** Enter a file path for the report file. For example, C:\tools\reports\
- **Role Name:** Enter the role name for the report. For example, lasmonitor.
- **Description:** Enter the report description. For example, Reports all tags which are equal or below the system property BatteryThreshold.

Click **Save** to save your report.

Note: Fields marked with an asterisk (*) are required. All other fields are optional.

Reports Operation

Use this page to select and view customized reports.

The reports are listed in table format by **Report Name** and **Description**.

If you enable WebSphere Application Server security, access to the reports are granted on role-based security. When you log in with a user ID that is in a group that has the associated role, then you can view the reports associated with that specific role as well as view all reports that have not been associated with any role.

If you do not enable WebSphere Application Server security, all reports can be viewed regardless of their specified roles because role-based security will not be applied.

Click **Display** to display the selected report or click **Reload** to reload the data from the database.

Developing

Use Web services to customize Location Awareness Services for WebSphere Sensor Events.

Web services

Web services are self-contained, modular applications that can be described, published, located, and invoked over a network. They implement Service Oriented Architecture (SOA), which supports the connecting or sharing of resources and data in a flexible and standardized manner. Services are described and organized to support their dynamic, automated discovery and reuse.

Tip: See the WebSphere Application Server information center for more information about implementing Web service applications.

The following WSDL (Web Services Description Language) files are provided by Location Awareness Services for WebSphere Sensor Events:

- http://host_name:9080/PremisesCEPRuleInstantiationWebServiceEJBHttpRouter/services/RuleInstantiationWS?wsdl
- http://host_name:9080/AtlasImportEJBHttpRouter/services/LasItemRegistration?wsdl
- http://host_name:9080/AtlasImportEJBHttpRouter/services/LasItemDetail?wsdl
- http://host_name:9080/AtlasImportEJBHttpRouter/services/LasItemMetaData?wsdl
- http://host_name:9080/LasEventHandlingEJBHttpRouter/services/LasEventHandling?wsdl
- http://host_name:9080/LasQueryEJBHttpRouter/services/LasQuery?wsdl
- http://host_name:9080/LasCeiMessageWrapper.wsdl
- http://host_name/wsdl/EMailHandler.wsdl

This XML-based language is used to create a description of an underlying application. It is this description that turns an application into a Web service, by acting as the interface between the underlying application and other Web-enabled applications.

The following Web services are provided by Location Awareness Services for WebSphere Sensor Events:

- “LasRuleServices” on page 95
- “LasItemRegistrationServices” on page 103
- “LasItemDetailServices” on page 108
- “LasItemMetaDataServices” on page 110
- “LasEventHandlingServices” on page 112
- “LasQueryServices” on page 120
- “handleEvent” on page 121

Security

Location Awareness Services for WebSphere Sensor Events supports HTTPS transport binding for its Web services.

If security is enabled for WebSphere Application Server, the Web services are available only through HTTPS and a secure port (usually 9443). Location Awareness Services for WebSphere Sensor Events Web services are secured by HTTP Basic Authentication as well. This means that authorization occurs using the user name and password provided in the HTTP headers.

HTTPS

HTTPS is a well-known and often-used mechanism to secure HTTP Internet and intranet communications. HTTPS is based on a Secure Sockets Layer (SSL) or Transport Layer Security (TLS) that runs beneath HTTP. HTTPS encrypts the entire HTTP data packet. It also provides security features including party identification and message integrity. Mutual authentication (where the client authenticates to the server and the server authenticates to the client) is possible. If you intend to access Web services protected by HTTPS, certificate stores must be prepared on the client side.

For more information on these topics, see Using HTTP to transport Web services and Invoking outbound services over HTTPS in the WebSphere Application Server information center.

Localization and input parameters

All the Web services (except for LasRuleServices) have *locale_descriptor* as input parameter.

This parameter allows you to specify:

- Language
- Country
- Vendor specific information (such as operating system)

This descriptor enables the server to parse the values it receives from clients and return localized values (such as item properties). The format for the returned values is in the same locale. If the returned value is a message, then the message is translated according to the language of the specified locale, if a translation is available.

The property values must use strings for the locale provided in the *locale_descriptor* input parameter. This applies to integer and double properties.

Timestamp input parameters (such as *lastUpdateTime*, *start_time*, and *end_time*) are locale independent. This means that they are bound to the server time zone and a specific formatting pattern as returned by the server: *yyyy-mm-dd hh:mm:ss.fffffffff*, where *fffffffff* indicates nanoseconds.

LasRuleServices

The following Web services allow you to create, deploy, update, undeploy, hold, and delete rule instances.

createRuleInstance:

Purpose

This Web service allows you to create a new rule in the Location Awareness Services for WebSphere Sensor Events database, but it will not take affect until it is deployed. To use this method, you must provide the Web Service interface with internal information on zone IDs, class IDs, group IDs, and item IDs. You can obtain this information by browsing the following ATLASDB tables and by invoking another Web service.

- For zone IDs: Browse the ATLASDB.ZONES table for information on zone IDs.
- For class IDs: Browse the ATLASDB.CLASSES table for information on class IDs.
- For group IDs: Browse the ATLASDB.GROUPS table for information on group IDs.
- For item IDs: Invoke the Web service “viewItem” on page 108 of “LasItemDetailServices” on page 108 to obtain the item IDs.

Syntax

```
void createRuleInstance(String type,String name,String description,KeywordValuePair[] attributes,boolean deploy)
```

Input

type: For a description of each of the rule types, see “Business Rules” on page 53. The following is a list of valid rule types:

- Duration of Stay in Zone
- Items Association
- Man Down Detection
- Maximum Items per Zone Threshold
- Visitor Escorting
- Zone Access Restriction
- Zone Exit Restriction

name: The unique name for this rule. The maximum name length size is 64 bytes. Quotation marks cannot be used.

description: A description of the rule.

attributes: Attributes for the rule that are specified by keyword and value pairs. For more information on attributes for specified rule types, see Table 14 on page 96 and Table 15 on page 97.

deploy: Indicates whether the rule is to be deployed. Valid values are true or false.

The following tables contain lists of keywords and example values for the *attributes* variable. For additional information on defining rules, see “Business Rules” on page 53.

Table 14 for **Visitor Escorting** and **Items Association** contains a list of valid keywords and values. It is important to note that all the keywords are required.

Table 14. Attributes for Visitor Escorting and Items Association rules

Keyword	Example value	Value description
zoneType	1	The following are valid values for zoneType: <ul style="list-style-type: none"> • 1 : Indicates that the value for the zone parameter represents the zone ID. • 2 : Indicates that the value of the zone parameter represents a zone class ID.
zone	2	Depending on zoneType, this value represents either the zone ID or a zone class ID.
alertActions	3	This value is a sum of the following possible values: <ul style="list-style-type: none"> • 1 = log alert • 2 = display alert • 4 = custom notification In this example, the resulting value 3 is the sum of the value for a log alert (1) and a display alert (2).
itemType1	3	Keyword itemType1 corresponds to the Visitor. Keyword itemType2 corresponds to the Escort. Possible values can be one of the following: <ul style="list-style-type: none"> • 1 = item ID • 3 = class ID • 4 = group ID
itemType2	1	
item1	1	Depending on itemType, this value represents either an item ID, a class ID, or a group ID.
item2	4343	
additionalParameter1	30	This value is the maximum tolerated distance, in units, that the visitor can be away from the escort. Note: Currently, the edge length of the visitor (who is in the container class) determines the maximum tolerated distance.
additionalParameter2	400	This value is the tolerated rule violation time, in seconds.

For **Duration of Stay in Zone**, **Maximum Items per Zone Threshold**, **Zone Access Restriction**, **Zone Exit Restriction**, and **Man Down Detection**, Table 15 on page 97 contains a list of valid keywords and values for the attributes variable. Only activityPattern is required by all the specified rule types. The additionalParameter1 keyword is required but only valid for **Duration of Stay in Zone** and **Maximum Items per Zone Threshold**.

Table 15. Attributes for Duration of Stay in Zone, Maximum Items per Zone Threshold, Zone Access Restriction, Zone Exit Restriction, and Man Down Detection.

Keyword	Example value	Value description
activityPattern	A:	<p>This keyword is required. The activityPattern specifies the time frame when the rule should be applied. The following is a list of values for the rule's activity pattern:</p> <ul style="list-style-type: none"> • A: – Always active. • D:2008/02/17-19:26:00;2008/02/25-19:26:00; – Discretely active. For this example, the rule is active only for the specified time from February 17, 2008, at 7:26:00 p.m. through February, 25, 2008 at 7:26:00 p.m. • R:2+[08:00:00-09:00:00];3-[08:00:00-09:00:00]; – Repetitively active. For this example, the rule is repetitively active on Tuesdays from 8:00:00 a.m. to 9:00:00 a.m. and on Wednesdays except from 8:00:00 a.m. to 9:00:00 a.m.
class1	1	This value represents the class name to be included or excluded.
exclClass1		Class parameters are used to define the actor for a rule. ¹
group1	3	This value represents the group name to be included or excluded.
exclGroup1		Group parameters are used to define the actor for a rule. ¹
attrName1	TagID, Label	This value represents the name of the attribute used for the inclusion or exclusion filter.
exclattrName1		Attribute name parameters are used to define the actor for a rule. ^{1 2}
attrOperator1	endsWith >= contains	<p>This value represents the operator used for the inclusion or exclusion filter.</p> <p>The supported operator values are:</p> <ul style="list-style-type: none"> • equals • equalsIgnoreCase • unequals • unequalsIgnoreCase • > • >= • < • <= • startsWith • endsWith • contains <p>Attribute operator parameters are used to define the actor for a rule. ^{1 2}</p>
exclattrOperator1		

Table 15. Attributes for Duration of Stay in Zone, Maximum Items per Zone Threshold, Zone Access Restriction, Zone Exit Restriction, and Man Down Detection. (continued)

Keyword	Example value	Value description
attrValue1	133	The attribute value used for the inclusion or exclusion filter.
exclattrValue1		Attribute value parameters are used to define the actor for a rule. ^{1 2}
Zone	2	Represents the zone name to be included or excluded. ³
exclZone		
zoneType	6	Represents the zone class name to be included or excluded. ³
exclZoneType		
alertActions	3	<p>This value is a sum of the following possible values:</p> <ul style="list-style-type: none"> • 1 = log alert • 2 = display alert • 4 = custom notification <p>In this example, the resulting value 3 is the sum of the value for a log alert (1) and a display alert (2).</p>
additionalParameter1	varies depending on the rule	<p>This keyword is required but only valid for the following rule types:</p> <ul style="list-style-type: none"> • Maximum Items per Zone Threshold. Specify the maximum number of actors, such as 30. • Duration of Stay in Zone. Specify the maximum duration of stay in seconds. • Specify the down time in seconds, such as 120. This keyword is only valid for the Man Down Detection rule type.

Output

This Web service returns nothing.

updateRuleInstance:

1. The following applies when defining an actor:

- It is possible to define an actor by inclusion, exclusion, or both.
- Specification of an actor is required.
- Actor specification includes at least class or group or attribute specification, or any combination of them.
- You can specify one attribute as the filter criteria. This can be an attribute independent of class (for example, TagID or Label) or an attribute of the selected class. (You can retrieve this by the metadata of the item.)

2. When filtering by attribute, the name, operator, and value keywords all need to be defined for the rule.

- The following expression describes which keywords must be defined when filtering by attribute:
(attrName1 AND attrOperator1 AND attrValue1) OR
(exclattrName1 AND exclattrOperator1 AND exclattrValue1)
- It is possible to use both the include and the exclude attributes in one rule.
- The name should be a valid attribute name for the specified class. The operator should be valid for the type of attribute.

Purpose

This Web service allows you to update an already existing rule. When a rule instance is updated, the old version is deleted and a new instance is created.

Syntax

void **updateRuleInstance**(**String** *type*,**String** *name*,**String** *description*,**KeywordValuePair[]** *attributes*,**boolean** *deploy*)

Input

type: For a description of each of the rule types, see “Business Rules” on page 53. The following is a list of valid rule types:

- Duration of Stay in Zone
- Items Association
- Man Down Detection
- Maximum Items per Zone Threshold
- Visitor Escorting
- Zone Access Restriction
- Zone Exit Restriction

name: The unique name for this rule. The maximum name length size is 64 bytes. Quotation marks cannot be used.

description: A description of the rule.

attributes: Attributes for the rule that are specified by keyword and value pairs. For more information on attributes for specified rule types, see Table 16 on page 100 and Table 17 on page 101.

deploy: Indicates whether the rule is to be deployed. Valid values are true or false.

The following tables contain lists of keywords and example values for the *attributes* variable. For additional information on defining rules, see “Business Rules” on page 53.

Table 16 on page 100 for **Visitor Escorting** and **Items Association** contains a list of valid keywords and values. It is important to note that all the keywords are required.

3. When filtering by using the zone related keywords, you can only define Zone or zoneType, but not both keywords for the rule.

- The following expression describes restrictions for using the Zone and zoneType keywords: (zoneType XOR zone) OR (exclZoneType XOR exclZone).
- It is possible to use both the include and the exclude zone related keywords in one rule.

Table 16. Attributes for Visitor Escorting and Items Association rules

Keyword	Example value	Value description
zoneType	1	The following are valid values for zoneType: <ul style="list-style-type: none"> • 1 : Indicates that the value for the zone parameter represents the zone ID. • 2 : Indicates that the value of the zone parameter represents a zone class ID.
zone	2	Depending on zoneType, this value represents either the zone ID or a zone class ID.
alertActions	3	This value is a sum of the following possible values: <ul style="list-style-type: none"> • 1 = log alert • 2 = display alert • 4 = custom notification <p>In this example, the resulting value 3 is the sum of the value for a log alert (1) and a display alert (2).</p>
itemType1	3	Keyword itemType1 corresponds to the Visitor. Keyword itemType2 corresponds to the Escort. Possible values can be one of the following: <ul style="list-style-type: none"> • 1 = item ID • 3 = class ID • 4 = group ID
itemType2	1	
item1	1	Depending on itemType, this value represents either an item ID, a class ID, or a group ID.
item2	4343	
additionalParameter1	30	This value is the maximum tolerated distance, in units, that the visitor can be away from the escort. Note: Currently, the edge length of the visitor (who is in the container class) determines the maximum tolerated distance.
additionalParameter2	400	This value is the tolerated rule violation time, in seconds.

For **Duration of Stay in Zone**, **Maximum Items per Zone Threshold**, **Zone Access Restriction**, **Zone Exit Restriction**, and **Man Down Detection**, Table 17 on page 101 contains a list of valid keywords and values for the attributes variable. Only activityPattern is required by all the specified rule types. The additionalParameter1 keyword is required but only valid for **Duration of Stay in Zone** and **Maximum Items per Zone Threshold**.

Table 17. Attributes for Duration of Stay in Zone, Maximum Items per Zone Threshold, Zone Access Restriction, Zone Exit Restriction, and Man Down Detection.

Keyword	Example value	Value description
activityPattern	A:	<p>This keyword is required. The activityPattern specifies the time frame when the rule should be applied. The following is a list of values for the rule's activity pattern:</p> <ul style="list-style-type: none"> • A: – Always active. • D:2008/02/17-19:26:00;2008/02/25-19:26:00; – Discretely active. For this example, the rule is active only for the specified time from February 17, 2008, at 7:26:00 p.m. through February, 25, 2008 at 7:26:00 p.m. • R:2+[08:00:00-09:00:00];3-[08:00:00-09:00:00]; – Repetitively active. For this example, the rule is repetitively active on Tuesdays from 8:00:00 a.m. to 9:00:00 a.m. and on Wednesdays except from 8:00:00 a.m. to 9:00:00 a.m.
class1	1	This value represents the class name to be included or excluded.
exclClass1		Class parameters are used to define the actor for a rule. ⁴
group1	3	This value represents the group name to be included or excluded.
exclGroup1		Group parameters are used to define the actor for a rule. ⁴
attrName1	TagID, Label	This value represents the name of the attribute used for the inclusion or exclusion filter.
exclattrName1		Attribute name parameters are used to define the actor for a rule. ^{4 5}
attrOperator1	endsWith >= contains	<p>This value represents the operator used for the inclusion or exclusion filter.</p> <p>The supported operator values are:</p> <ul style="list-style-type: none"> • equals • equalsIgnoreCase • unequals • unequalsIgnoreCase • > • >= • < • <= • startsWith • endsWith • contains
exclattrOperator1		Attribute operator parameters are used to define the actor for a rule. ^{4 5}

Table 17. Attributes for Duration of Stay in Zone, Maximum Items per Zone Threshold, Zone Access Restriction, Zone Exit Restriction, and Man Down Detection. (continued)

Keyword	Example value	Value description
attrValue1	133	The attribute value used for the inclusion or exclusion filter.
exclattrValue1		Attribute value parameters are used to define the actor for a rule. ^{4 5}
Zone	2	Represents the zone name to be included or excluded. ⁶
exclZone		
zoneType	6	Represents the zone class name to be included or excluded. ⁶
exclZoneType		
alertActions	3	<p>This value is a sum of the following possible values:</p> <ul style="list-style-type: none"> • 1 = log alert • 2 = display alert • 4 = custom notification <p>In this example, the resulting value 3 is the sum of the value for a log alert (1) and a display alert (2).</p>
additionalParameter1	varies depending on the rule	<p>This keyword is required but only valid for the following rule types:</p> <ul style="list-style-type: none"> • Maximum Items per Zone Threshold. Specify the maximum number of actors, such as 30. • Duration of Stay in Zone. Specify the maximum duration of stay in seconds. • Specify the down time in seconds, such as 120. This keyword is only valid for the Man Down Detection rule type.

Output

This Web service returns nothing.

deleteRuleInstance:

4. The following applies when defining an actor:

- It is possible to define an actor by inclusion, exclusion, or both.
- Specification of an actor is required.
- Actor specification includes at least class or group or attribute specification, or any combination of them.
- You can specify one attribute as the filter criteria. This can be an attribute independent of class (for example, TagID or Label) or an attribute of the selected class. (You can retrieve this by the metadata of the item.)

5. When filtering by attribute, the name, operator, and value keywords all need to be defined for the rule.

- The following expression describes which keywords must be defined when filtering by attribute:
`(attrName1 AND attrOperator1 AND attrValue1) OR (exclattrName1 AND exclattrOperator1 AND exclattrValue1)`
- It is possible to use both the include and the exclude attributes in one rule.
- The name should be a valid attribute name for the specified class. The operator should be valid for the type of attribute.

Purpose

This Web service deletes a rule instance, removing it from the database and from the CEP runtime engine.

Syntax

void deleteRuleInstance (String type,String name)

Input

type: For a description of each of the rule types, see “Business Rules” on page 53. The following is a list of valid rule types:

- Duration of Stay in Zone
- Items Association
- Man Down Detection
- Maximum Items per Zone Threshold
- Visitor Escorting
- Zone Access Restriction
- Zone Exit Restriction

name: The unique name for this rule. The maximum name length size is 64 bytes. Quotation marks cannot be used.

LasItemRegistrationServices

Location Awareness Services for WebSphere Sensor Events provides Web services to assist you in defining, updating, and deleting items in your installation.

When using these Web services, you can specify items by supplying the key properties that match the item class or by specifying the item ID. Location Awareness Services for WebSphere Sensor Events only checks for the attributes relevant to the specified class. If additional attributes are provided, they are ignored.

Note:

- Any dates must be supplied in the MM/dd/yyyy (month/day/year) format.
- The timestamp parameter, *LastUpdateTime*, makes sure that you have the most recent version of the item before you attempt to update or delete it. Timestamp input parameters are bound to the server time zone.
- You can only specify one instance of a key property attribute, but there can be multiple instances of other property attributes (according to the class schema).

createItem:

6. When filtering by using the zone related keywords, you can only define Zone or zoneType, but not both keywords for the rule.

- The following expression describes restrictions for using the Zone and zoneType keywords: (zoneType XOR zone) OR (exclZoneType XOR exclZone).
- It is possible to use both the include and the exclude zone related keywords in one rule.

Purpose

This Web service creates new items in the Location Awareness Services for WebSphere Sensor Events database.

Syntax

```
ItemIDWithTimeStamp createItem(className,KeywordValuePair[]  
keyPairs,KeywordValuePair[] optPairs,String[] groups,LocaleDescriptor locale)
```

Input

className: A string that represents name of the class for the new item.

keyPairs: An array from the type, `com.ibm.atlas.adminobjects.lasitemregistration.KeywordValuePair`. This parameter is a keyword-value pair that represents key properties. The combination of values has to be unique. The keyword for each keyword-value pair is the name of a key property, and the value is the value for that key property. You can set these with `setKeyword()` and `setValue()`. The values must be in the correct type and in the correct format. For example, if the property type is an integer, then the value must also be an integer and can contain any number between 0 and 9.

optPairs: An array from type, `com.ibm.atlas.adminobjects.lasitemregistration.KeywordValuePair`. This parameter is a keyword-value pair that represents optional properties. The keyword for each keyword-value pair is the name of an optional property, and the value is the value for that optional property. You can set these with `setKeyword()` and `setValue()`. If the `MinOccurs` value of the property is 1, then the property is mandatory and you must enter a value for it. Some properties are not listed in metadata of the class but the method, `createItem`, accepts it. These properties are:

- `iconLink`
- `iconLable`
- `tagid`
- `edgeLength` - Accepted if the **Container** property is selected in the **Class Details View** of the Classes/Items Manager portlet in the WebSphere Application Server administrative console.
- `parentTagID` - Accepted if the tag of the parent exists and if the **Container** property in the **Class Details View** of the Classes/Items Manager portlet is marked for a parent item.
- `parentItemID` - Accepted if the parent item exists in the **Class Details View** of the Classes/Items Manager portlet and if the **Container** property is marked for a parent item.

The values must be in the correct type and in the correct format. For example, if a property type is a date, then the value must be in the MM/dd/yyyy format.

groups: An array from the type, `String`, that contains the groups for which the new item is a member. This array can also be null if the item should not be a member in any group.

locale: If you want to set the default value, you have to input null. See "Localization and input parameters" on page 94 for more information about the locale descriptor.

Output

The output from the Web service is an object from the type, `ItemIDWithTimeStamp`. This contains the item ID and the timestamp of the last update for the new item.

updateItem:

Purpose

This Web service updates optional attributes for an item in the Location Awareness Services for WebSphere Sensor Events database. Key properties cannot be changed with this service. You only need to provide the attributes you want to change. To blank out existing attributes, provide blank or null values. A list of groups always replaces the old list of groups.

Syntax

`ItemIDWithTimeStamp updateItem (className,KeywordValuePair[] keyPairs,KeywordValuePair[] optPairs,String[] groups,LocaleDescriptor locale)`

Input

className: A string that represents name of the class for the item that should be changed.

keyPairs: An array from the type, `com.ibm.atlas.adminobjects.lasitemregistration.KeywordValuePair`. This parameter is a keyword-value pair that represents key properties. The combination of values has to be unique. The keyword for each keyword-value pair is the name of a key property, and the value is the value for that key property. You can set these with `setKeyword()` and `setValue()`. The values must be in the correct type and in the correct format. For example, if the property type is an integer, then the value must also be an integer and can contain any number between 0 and 9.

optPairs: An array from type, `com.ibm.atlas.adminobjects.lasitemregistration.KeywordValuePair`. This variable is a keyword-value pair that represents optional properties. The keyword for each keyword-value pair is the name of an optional property, and the value is the value for that optional property. You can set these with `setKeyword()` and `setValue()`. The values must be in the correct type and in the correct format. For example, if a property type is a date, then the value must be in the MM/dd/yyyy format. You only need to provide attributes you want to change.

groups: An array from the type, `String`, that, if not set to null, contains the assigned groups for the item. A new list of groups always replaces the old list of groups. To remove group assignments, set a blank list.

locale: If you want to set the default value, you have to input null. See "Localization and input parameters" on page 94 for more information about the locale descriptor.

Output

The output from the Web service is an object from the type, `ItemIDWithTimeStamp`. This contains the item ID and the timestamp of the last update for the new item.

updateItemById:

Purpose

This Web service allows you to update attributes for an item in the Location Awareness Services for WebSphere Sensor Events database. You only need to provide the attributes you want to change. To blank out existing attributes, provide empty or null values.

Syntax

ItemIDWithTimeStamp **updateItemById**(**int** *itemID*,**KeywordValuePair**[[*keyPairs*,**KeywordValuePair**[[*optPairs*,**String**[[*groups*,**String** *lastUpdateTime*,**LocaleDescriptor** *locale*

Input

itemID: The item ID (integer) of the item to be updated.

keyPairs: An array from the type, `com.ibm.atlas.adminobjects.lasitemregistration.KeywordValuePair`. This parameter is a keyword-value pair that represents key properties. The combination of values has to be unique. The keyword for each keyword-value pair is the name of a key property, and the value is the value for that key property. You can set these with `setKeyword()` and `setValue()`. The values must be in the correct type and in the correct format. For example, if the property type is an integer, then the value must also be an integer and can contain any number between 0 and 9.

optPairs: An array from type, `com.ibm.atlas.adminobjects.lasitemregistration.KeywordValuePair`. This variable is a keyword-value pair that represents optional properties. The keyword for each keyword-value pair is the name of an optional property, and the value is the value for that optional property. You can set these with `setKeyword()` and `setValue()`. The values must be in the correct type and in the correct format. For example, if a property type is a date, then the value must be in the MM/dd/yyyy format. You only need to provide attributes you want to change.

groups: An array from the type, `String`, that, if not set to null, contains the assigned groups for the item. A new list of groups always replaces the old list of groups. To remove group assignments, set a blank list.

lastUpdateTime: This `String` has the format: yyyy-mm-dd hh:mm:ss.ffffffffff. You can get this by creating the object, `LasItem`, using the `viewItemById` method. The object, `LasItem`, offers a method calls `getLastUpdateTime()`.

locale: If you want to set the default value, you have to input null. See "Localization and input parameters" on page 94 for more information about the locale descriptor.

Output

The output from the Web service is an object from the type, `ItemIDWithTimeStamp`. This contains the item ID and the timestamp of the last update for the new item.

deleteItem:

Purpose

This Web service deletes items from the Location Awareness Services for WebSphere Sensor Events database. All dependent records such as group relations or parent item relationships will be deleted as well. The history entries remain.

Syntax

ItemIDWithTimeStamp **deleteItem**(*className*,**KeywordValuePair**[] *keyPairs*,**LocaleDescriptor** *locale*)

Input

className: A string that represents name of the class for the item that should be deleted.

keyPairs: An array from the type, `com.ibm.atlas.adminobjects.lasitemregistration.KeywordValuePair`. This parameter is a keyword-value pair that represents key properties. The combination of values has to be unique. The keyword for each keyword-value pair is the name of a key property, and the value is the value for that key property. You can set these with `setKeyword()` and `setValue()`. The values must be in the correct type and in the correct format. For example, if the property type is an integer, then the value must also be an integer and can contain any number between 0 and 9.

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

deleteItemById: Purpose

This Web service allows you to delete an item in the Location Awareness Services for WebSphere Sensor Events database. All dependent records, such as group relations or parent item relationships, are deleted as well. History entries remain.

Syntax

ItemIDWithTimeStamp **deleteItemById**(**int** *itemID*,**String** *lastUpdateTime*,**LocaleDescriptor** *locale*)

Input

itemID: The item ID (integer) of the item to be deleted. The item ID must match an existing item in Location Awareness Services for WebSphere Sensor Events.

lastUpdateTime: This String has the format: `yyyy-mm-dd hh:mm:ss.ffffffffff`. You can get this by creating the object, `LasItem`, using the `viewItemById` method. The object, `LasItem`, offers a method calls `getLastUpdateTime()`.

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Output

The output from the Web service is the item ID.

registerItem:

Purpose

This Web service can register an item to a tag.

Syntax

ItemIDWithTimeStamp **registerItem**(int *itemID*,String *tagID*,String *lastUpdateTime*,**LocaleDescriptor** *locale*)

Input

itemID: The item ID is from type int (integer). If an item is not yet registered to tag, you can register the item using its ID. The item ID is an attribute of the item.

tagID: A string that has to be unique. To unregister a tag ID, enter blanks or “ ” as the tag ID.

lastUpdateTime: If you want to set the default value, you have to input null. If you want to use an item from a specific time, you have to set the lastUpdateTime. It is an attribute of an item.

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Output

The output from the Web service is an object from the type, *ItemIDWithTimeStamp*. This contains the item ID and the timestamp of the last update for the new item.

LasItemDetailServices

The following Web Services allow you to obtain information about items.

viewItem:

Purpose

This Web service returns all tag attributes for an item, including tag attributes. If a timestamp is entered, it returns the historical attributes. It does not return the current position or outstanding events.

Syntax

LasItem **viewItem**(String *className*,**KeywordValuePairs**[] *keyPairs*,**LocaleDescriptor** *locale*,String *timeStamp*)

Input

className: A string that represents name of the class for the item.

keyPairs: This variable is a keyword-value pair that represents key properties. The keyword for each keyword-value pair is the name of a key property, and the value is the value for that key property.

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

timeStamp: If you want to set the default value, input null. If you want to get a tag from a specific time, set the *timeStamp* variable in the following format:
yyyy-mm-dd hh:mm:ss.ffffffffff

Output

You can query the following details from the returned object, *LasItem*:

- Class name
- Item ID
- The time of the last update
- Assigned groups
- Key attributes
- Optional attributes

viewItemById:

Purpose

This Web service returns the current or historical details of a given item. The condition for this Web service is the method, *getLasItemDetail*, which returns a *LasItemDetail* and contains all the information for an item.

Syntax

LasItem **viewItemById**(String *itemID*,LocaleDescriptor *locale*,String *timeStamp*)

Input

itemID: The item ID of the item to be viewed. You can get the item ID using the *getItemID()* method of the *TagDetail*.

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

timeStamp: If you want to set the default value, input null. If you want to get a tag from a specific time, set the *timeStamp* variable in the following format:
yyyy-mm-dd hh:mm:ss.ffffffffff

Output

You can query the following details from the returned object, *LasItem*:

- Class name
- Item ID
- The time of the last update
- Assigned groups

- Key attributes
- Optional attributes

LasItemMetaDataServices

The following Web service allows you to query Location Awareness Services for WebSphere Sensor Events metadata.

getItemClassDefinitions:

Purpose

Use this Web service to retrieve the metadata of available classes.

Syntax

LasItemClass[] **getItemClassDefinitions(LocaleDescriptor locale)**

Input

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Output

The output from the Web service is an array from the *LasItemClass* type, which contains all available classes and the metadata for each one. The available metadata includes:

- Class name
- Parent class name (can be empty)
- Key properties
 - Name
 - Type
- Optional properties
 - Name
 - Type
 - Minimum occurrences (Min occurs) - if the minimum is 1, then this property is mandatory
 - Maximum occurrences (Max occurs)

getGroupDefinitions:

Purpose

Use this Web Service method to retrieve the metadata of available groups.

Syntax

GroupDefinition[] **getGroupDefinitions(LocaleDescriptor locale)**

Input

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Output

The output from the Web service is an array from the GroupDefinition type, which contains all available classes and the metadata for each one. The available metadata includes:

- create (creation date)
- description
- group ID
- name
- parent groups
- rgb (group color)
- schema
- status

getZoneDefinitions:

Purpose

Use this Web Service method to retrieve detailed information about zones.

Syntax

ZoneClassDefinition[] **getZoneDefinitions(LocaleDescriptor locale)**

Input

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Output

The output from the Web service is an array from the ZoneDefinition type, which contains all available classes and the metadata for each one. The available metadata includes:

- Zone name
- Area name
- Coordinates
- Creation date
- MaxZ
- MinZ
- Zone properties (an array from the ZonePropertyInfo type)
 - Attribute
 - Attribute order
 - Creation date
 - Type
 - Value
 - Zone class ID
 - Zone properties position

getZoneClassDefinitions:

Purpose

Use this Web Service method to retrieve detailed information about zone classes.

Syntax

ZoneClassDefinition[] **getZoneClassDefinitions**(**LocaleDescriptor** *locale*)

Input

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Output

The output from the Web service is an array from the *ZoneClassDefinition* type, which contains all available classes and the metadata for each one. The available metadata includes:

- Name
- Description
- The time of the last update
- Schema
- Parent zone class

LasEventHandlingServices

The following Web services allow you handle events.

issueEvent:

Purpose

This Web service issues an event, such as *LasTagNotResponsive*.

Syntax

void **issueEvent**(**KeywordValuePair[]** *keyPairs*,**LocaleDescriptor** *locale*)

Input

keyPairs: An array of properties describing the event. For example:

```
"AlertType", "LasTagNotResponsive"
"MessageType", "LasTagNotResponsive"
"TagId", "00000007"
"ZoneName", "Zonename"
"EventTime", "12:12:30"
"TagLabel", "taglabel"
"TagClass", "Person"
"TagGroup", "Security"
```

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Output

This Web service returns nothing.

acknowledgeEvent:

Purpose

This Web service acknowledges a concrete event by using its global instance ID.

Syntax

void acknowledgeEvent(String globalInstanceID,String user,LocaleDescriptor locale)

Input

globalInstanceID: The global instance ID is part of the properties of an event. For more information about how you can query it, see “getEventXMLForFilter” on page 115 and “handleEvent” on page 121.

user: This string specifies the user who acknowledges the event. This value can be null to set the default value.

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Output

This Web service returns nothing.

acknowledgeEventForTag:

Purpose

This service acknowledges all active events for a given tag ID.

Syntax

void acknowledgeEventForTag(String tag_ID ,String user,LocaleDescriptor locale)

Input

tagID: A string that has to be unique.

user: This string specifies the user who acknowledges the event. This value can be null to set the default value.

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Output

This Web service returns nothing.

acknowledgeEventForFilter:

Purpose

This Web Service acknowledges active events using filtering criteria. Every event that fits the given criteria is acknowledged.

Syntax

void **acknowledgeEventForFilter**(**KeywordValuePair[]** *kv_filterCriteria*,**String** *user*,**LocaleDescriptor** *locale*)

Input

kv_filterCriteria: The first argument is an array from the type, `com.ibm.atlas.adminobjects.laseventhandling.KeywordValuePair`. The possible criteria are:

Table 18. Possible criteria for *kv_filterCriteria*

Keyword	Value
tagid	String For example: "0000007"
taglabel	String For example: „Tag 0000007"
tagclass	String For example: „Person"
taggroup	String For example: „Laptop"
handled	String yes (acknowledge) or no (active)
eventafter	SimpleDateFormat in the format: „yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"
eventbefore	SimpleDateFormat in format: „yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"
eventtype	String, which is one of the existing event types For example, "LasZoneEntry"

If you define more than one criterion, only events which meet all of the criteria are returned. Wildcards are not supported.

user: This string specifies the user who acknowledges the event. This value can be null to set the default value.

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Output

This Web service returns nothing.

getEventXMLForFilter:

Purpose

This Web service returns all events which meet the given criteria.

Syntax

KeywordValuePair[] **getEventXMLForFilter**(**KeywordValuePair[]**
kv_filterCriteria,**LocaleDescriptor** *locale*)

Input

kv_filterCriteria: The first argument is an array from the type, `com.ibm.atlas.adminobjects.laseventhandling.KeywordValuePair`. The possible criteria are:

Table 19. Possible criteria for *kv_filterCriteria*

Keyword	Value
tagid	String For example: "0000007"
taglabel	String For example: „Tag 0000007"
tagclass	String For example: „Person"
taggroup	String For example: „Laptop"
handled	String yes (acknowledge) or no (active)
eventafter	SimpleDateFormat in the format: „yyyy-MM-dd'T'HH:mm:ss.SSS'Z' "
eventbefore	SimpleDateFormat in format: „yyyy-MM-dd'T'HH:mm:ss.SSS'Z' "
eventtype	String, which is one of the existing event types For example, "LasZoneEntry"

If you define more than one criterion, only events which meet all of the criteria are returned. Wildcards are not supported.

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Ouput

An array from the type, `com.ibm.atlas.adminobjects.laseventhandling.KeywordValuePair`, is returned. To access to the values of each element in the array you can use `getKeyword()` and `getValue()`. `getKeyword()` returns the global instance ID and `getValue()` returns the event in the Common Base Event XML format.

getEventMessageForFilter:

Purpose

This Web service returns all events which meet the given criteria.

Syntax

`KeywordValuePair[] getEventMessageForFilter(KeywordValuePair[] kv_filterCriteria, LocaleDescriptor locale)`

Input

kv_filterCriteria: The first argument is an array from the type, `com.ibm.atlas.adminobjects.laseventhandling.KeywordValuePair`. The possible criteria are:

Table 20. Possible criteria for *kv_filterCriteria*

Keyword	Value
tagid	String For example: "0000007"
taglabel	String For example: „Tag 0000007"
tagclass	String For example: „Person"
taggroup	String For example: „Laptop"
handled	String yes (acknowledge) or no (active)
eventafter	SimpleDateFormat in the format: „yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"
eventbefore	SimpleDateFormat in format: „yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"
eventtype	String, which is one of the existing event types For example, "LasZoneEntry"

If you define more than one criterion, only events which meet all of the criteria are returned. Wildcards are not supported.

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Output

An array from the type, `com.ibm.atlas.adminobjects.laseventhandling.KeywordValuePair`, is returned. To access to the values of each element in the array you can use `getKeyword()` and `getValue()`. `getKeyword()` returns the global instance ID and `getValue()` returns the event as a localized message.

archiveEvent:

Purpose

This Web service archives a concrete event, using its global instance ID, to a named target file.

Syntax

```
void archiveEvent(String globalInstanceID,String target,LocaleDescriptor locale)
```

Input

globalInstanceID: The global instance ID is part of the properties of an event. For more information about how you can query it, see “`getEventXMLForFilter`” on page 115 and “`handleEvent`” on page 121.

target: The target represents a string which contains the name and path of the target text file. The text file is created in a directory on the server. For the *IP address of server* variable, you can use the value of `localhost` to save the file on the server. Specify the path as follows: */////IP address of server//hard disk drive\$/path to file/file name*

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Output

This Web service returns nothing.

archiveEventForFilter:

Purpose

This Web Service archives events to a text file on a server using filtering rules. Every event that fits the given rule is archived.

Syntax

```
void archiveEventForFilter(KeywordValuePair[] kv_filterCriteria,String target,LocaleDescriptor locale)
```

Input

kv_filterCriteria: The first argument is an array from the type, `com.ibm.atlas.adminobjects.laseventhandling.KeywordValuePair`. The possible criteria are:

Table 21. Possible criteria for *kv_filterCriteria*

Keyword	Value
tagid	String For example: "0000007"
taglabel	String For example: „Tag 0000007"
tagclass	String For example: „Person"
taggroup	String For example: „Laptop"
handled	String yes (acknowledge) or no (active)
eventafter	SimpleDateFormat in the format: „yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"
eventbefore	SimpleDateFormat in format: „yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"
eventtype	String, which is one of the existing event types For example, "LasZoneEntry"

If you define more than one criterion, only events which meet all of the criteria are returned. Wildcards are not supported.

target: The target represents a string which contains the name and path of the target text file. The text file is created in a directory on the server. For the *IP address of server* variable, you can use the value of localhost to save the file on the server. Specify the path as follows: *////IP address of server//hard disk drive\$/path to file/file name*

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Output

This Web service returns nothing.

deleteEvent: Purpose

This Web service deletes a concrete event by using its global instance ID.

Syntax

```
void deleteEvent(String globalInstanceID,LocaleDescriptor locale)
```

Input

globalInstanceID: The global instance ID is part of the properties of an event. For more information about how you can query it, see “getEventXMLForFilter” on page 115 and “handleEvent” on page 121.

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Output

This Web service returns nothing.

deleteEventForFilter: **Purpose**

This Web Service deletes events using filtering rules. Every event that fits the given rule is deleted.

Syntax

void deleteEventForFilter(KeywordValuePair[] kv_filterCriteria,LocaleDescriptor locale)

Input

kv_filterCriteria: The first argument is an array from the type, `com.ibm.atlas.adminobjects.laseventhandling.KeywordValuePair`. The possible criteria are:

Table 22. Possible criteria for *kv_filterCriteria*

Keyword	Value
tagid	String For example: "0000007"
taglabel	String For example: „Tag 0000007"
tagclass	String For example: „Person"
taggroup	String For example: „Laptop"
handled	String yes (acknowledge) or no (active)
eventafter	SimpleDateFormat in the format: „yyyy-MM-dd'T'HH:mm:ss.SSS'Z' "
eventbefore	SimpleDateFormat in format: „yyyy-MM-dd'T'HH:mm:ss.SSS'Z' "

Table 22. Possible criteria for `kv_filterCriteria` (continued)

Keyword	Value
eventtype	String, which is one of the existing event types For example, "LasZoneEntry"

If you define more than one criterion, only events which meet all of the criteria are returned. Wildcards are not supported.

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Output

This Web service returns nothing.

LasQueryServices

The following Web service allows you to query tag data.

getTagDetail:

Purpose

This Web Service returns the current or historical details of a given tag.

Syntax

TagDetail **getTagDetail**(String *tagID*,String *timeStamp*,LocaleDescriptor *locale*)

Input

tagID: The first argument of the method is a string that represents the tag ID. In this case, you get the tag ID using `AtlasCeiEvent` because `AtlasCeiEvent` contains information about the event, and the tag activates the event.

timeStamp: If you want to set the default value, input null. If you want to get a tag from a specific time, set the `timeStamp` variable in the following format:
yyyy-mm-dd hh:mm:ss.ffffffffff

locale: If you want to set the default value, you have to input null. See “Localization and input parameters” on page 94 for more information about the locale descriptor.

Output

This Web service returns an object from the type, `TagDetail`. From that object, you can get the following information about the tag:

- Battery
- Class name
- Icon link
- Item ID
- Item label
- Time last seen

- Parent ID
- Tag ID
- Coordinates
- Events
- Zone times

handleEvent

Purpose

This Web service is a sample Web service that triggers an e-mail based on an event. This service is called when it is configured as a subscriber for events and if the configured event filter matches the incoming event.

Syntax

handleEvent(*string*,*serialized_common_base_event*,**LocaleDescriptor** *locale*)

Output

Information about the event is returned.

Use cases

This section contains possible use cases for Location Awareness Services for WebSphere Sensor Events.

SOA integration

In this release, Location Awareness Services for WebSphere Sensor Events aligns with the Service Oriented Architecture (SOA) approach by enabling alert-driven business processes.

Location Awareness Services for WebSphere Sensor Events is able to call Web services and trigger a business process in case of a business alert. To set this up, customers must register a Web service and define proprietary filter criteria for the Web service to be called. Then when Location Awareness Services for WebSphere Sensor Events issues an alert via CEI, a MDB listener is called, which checks whether the filter criteria for a Web service are met and calls the Web service with the common base event containing the details of the alert.

Additionally, other existing interfaces are available for use with the Web interfaces. These interfaces include:

- WebSphere MQ to import data and location events.
- Servlet interfaces to maintain and query areas and zones, monitor tag details, acknowledge alerts, and retrieve metadata for items.

Location Awareness Services for WebSphere Sensor Events provides sample Web services and customers can also create their own. The following scenarios provide examples of how you might use the provided Web services in a warehouse environment to implement business processes.

Integrating supply chain management

The following scenarios describe how you might use Location Awareness Services for WebSphere Sensor Events to integrate supply chain management business processes in a warehouse environment to track the arrival, storage, and decommissioning of goods.

This scenario consists of the following phases:

1. When new goods are ordered, the arriving pallet of goods must be placed in a specific zone. The position of the pallet is stored in Location Awareness Services for WebSphere Sensor Events.
2. When the zone is full of goods, a sub business process is started to move the pallets of goods to another storage location.
3. When the pallets of goods leave the warehouse, the contents are unregistered. In order to track a pallet of goods, the position of the pallet must be printed on the order.

The following scenarios include examples of how you might use the “Web services” on page 93 that are supplied with Location Awareness Services for WebSphere Sensor Events in the business processes.

Arriving goods

When an order arrives through the electronically available shipment manifest, the pallet tags that will arrive are registered in Location Awareness Services for WebSphere Sensor Events. You can do this using the `LasItemRegistrationServices - createItem` Web service.

Rules that govern the arrival and storage of the goods must already exist. If they do not, you can define rules using the `LasRuleServices - createRuleInstance` Web service. For example, you can define what types of goods must be placed in what zone. You can also specify that goods of a specific type must not enter or leave specific zones or that pallets in specific zones cannot contain more than a specified number of tags of a specific type.

When the pallet passes the entry gate or dock receiving door, any defined business processes are triggered in WebSphere Sensor Events, Data Capture and Delivery, and Location Awareness Services for WebSphere Sensor Events.

For example, when a forklift picks up a pallet with a certain tag ID, WebSphere Sensor Events can send a pickup event to Location Awareness Services for WebSphere Sensor Events using a WebSphere MQ request. During the move, rules correlate the pallet tag ID and its properties to the forklift’s position. (For example, if the item has already been defined as part of a group, such as “flammable group”, and the forklift moves to a restricted area where the pallet is not allowed to go because of its flammable content, an alert is triggered based on the defined business rules for zone entry or exit.) When the forklift releases the pallet, a message can be sent to Location Awareness Services for WebSphere Sensor Events specifying the position of the pallet.

Zone is full

Location Awareness Services for WebSphere Sensor Events evaluates business rules constantly. If more pallets are stored in a zone than is allowed by the business rules, an alert is issued that can trigger other business processes. You can do this using the `subscriberService` Web service. For example, the business process might

cause the pallets stored in the zone to be emptied.

Decommissioning goods

If a pallet is scheduled to be picked up at the position where it is currently stored and moved elsewhere, the supply chain management business process issues a request to Location Awareness Services for WebSphere Sensor Events for the location of the pallet tag ID (you can use the `LasQueryServices - getTagDetails` Web service) and also sends information such as the pallet tag ID and location to the pick up team. When the pallet passes the dock door on the way out, the supply chain management business processes trigger the appropriate business services and sends a message to Location Awareness Services for WebSphere Sensor Events indicating that the pallet tag ID has left the area or zone. If the pallet is leaving the premises WebSphere Sensor Events can decommission the pallet's tag ID by using the `ItemRegisterService` Web service.

It is also possible to request the duration time per pallet in a specific zone by requesting a report. You can use the `LasQueryServices - getTimeReportByTag` Web service.

Granting access to visitors

The following scenarios describe how you might use Location Awareness Services for WebSphere Sensor Events to track the movements of visitors on the premises.

In this scenario a contractor is ordered to temporarily work in special areas or zones of a company's premises. In order to track the contractor, the following actions must be taken:

- Register the contractor in the system.
- Grant access to the contractor to enter specific zones.
- Specify the zones where the contractor must be escorted by someone else.
- Specify the items, such as work-specific tools, that must remain in the work zone.

At the end of the temporary work assignment, the following actions must be taken:

- Unregister the contractor.
- Check the duration that the contractor remains in the work zone.

The following scenario includes examples of how you might use the "Web services" on page 93 that are supplied with Location Awareness Services for WebSphere Sensor Events in the business processes.

For example, when contractors arrive at the company's premises, they receive a tag ID and predefined business rules are activated. These business rules identify areas and zones of the company's premises where contractors are allowed to enter or where they require escorting. Contractor-specific rules can be created as part of the business process as needed. You can define business rules using the `LasRuleServices - createRuleInstance` Web service.

The contractors are registered in Location Awareness Services for WebSphere Sensor Events, for example using the `LasItemRegistrationServices - createItem` Web service.

When the contractors enter the premises or the zones defined for escorting, an escort can be informed automatically using the `SubscriberService` Web service of a contractor's arrival.

When the contractors enter the work zone, Location Awareness Services for WebSphere Sensor Events can verify that the relevant work tools are available for every worker in the zone. Otherwise they can be dispatched using the SubscriberService Web service. The number of tools can also be counted using the LasQueryServices - getTimeReportByZone Web service.

When the job is done and the contract ends, the contractors are unregistered using the LasItemRegistrationServices - registeritem Web service. A tools check can be performed for the work area using the ItemsInZoneService Web service and a final billing for the contractors' working hours is kicked off using the LasQueryServices - getTimeReportByTag Web service. Any contractor-specific rules can be deleted using the LasRuleServices - deleteRuleInstance Web service.

Using containers

Location Awareness Services for WebSphere Sensor Events allows you to use containment relationships to track items. For example, a containment relationship makes it easier to track items being shipped together, such as on a pallet.

About this task

Containment relationships are based on:

- Defined containment classes. You can define containers in Location Awareness Services for WebSphere Sensor Events by explicitly defining specific item classes (new or existing) in the "Classes/Items Manager" on page 47 portlet and specifying them as container classes or by importing containment relationships with a CSV file. Items in container classes can contain other items. For example, the item "palette1" is associated with the class "palettes". The item "palette1" contains item "screwdriver42".

You can dissolve containment relationships by editing the properties of the class to remove the container classification.

- External events that are processed by Location Awareness Services for WebSphere Sensor Events. For example, if the position of item indicates it is inside the container, it is automatically associated with the container. This requires that containers be precisely defined with spatial dimensions and the system property ContainerSupportOn must be checked. Since containers can be mobile, the spatial area occupied by them can change over time. Also, devices, such as complex Data Capture and Delivery devices, can be set up to deliver special events to indicate a container relationship. Containers cannot add other containers dynamically.

Using containers also allows you to visualize the location of all contained items, even if they are not tagged or their tags are not visible to a tag reader. In this case, when the container moves, Location Awareness Services for WebSphere Sensor Events assumes the contained items also move. Also, if a container enters or leaves a zone, Location Awareness Services for WebSphere Sensor Events assumes all contained items also enter or leave the zone and all rules apply to both the container and contained items. Rules can also be defined to prevent items from being removed from a container or added to a container.

It is important to note that the last reported position of a container and the position of its content can differ. This might be the case if the items in the container and the container itself are actively tagged. If contained items are removed from the container's location, you might want to remove the containment relationship and track all items separately. If the tags of the contained items are not visible or are read from a different tag reader than that of the container, the

position coordinates might be read as being located outside of the container. In this case, you might want to define business rules that cause the position of the tagged items in the container to be ignored.

The following restrictions apply to containers and contained items:

- Contained items do not need to be equipped with active tags; however, if contained items are equipped with active tags, the container must also be equipped with an active tag.
- If both containers and contained items are equipped with active tags, the same technology must be used for all tags and the accuracy and send frequency of the tags must be identical.
- Location Awareness Services for WebSphere Sensor Events assumes all containers are cubes; therefore specify a cube size that will most closely resemble the actual size of the container.

Defining a container and assigning items

This topic describes how to define a container class and container. It also describes how to assign items to the container.

About this task

Use the “Classes/Items Manager” on page 47 portlet to create a class of items that can contain other items and containers.

1. Log in to the WebSphere Application Server administrative console and click **Tag Registration** → **Classes/Items Manager** to access this page.
2. Click the **Class Details View** tab.
3. Click **Add Child Class** to define a new class.
4. Fill in the values for the class, making sure to select **Container** to specify the items in the class can contain other items. Also, make sure to enter the correct spatial measurements in **Edge Length** for the size of the containers in the class.
5. Save your settings.
6. Now, add an item to the container class you created.
 - a. Click the **Item View** tab.
 - b. Click **Add New Item** to define a new item in the class you created.
 - c. Fill in the values for the item.

If the edge length of the container is different than the class default, make sure to enter a value for **Edge Length**. If an item already exists and you change the **Edge Length** for the class, the edge length of the existing item is not changed.
 - d. Save your settings.
7. Assign items to the container.
 - a. Click the **Item View** tab.
 - b. Click **Edit Container** next to the new container item you created. A list of items are displayed that can be assigned to the container, as well as a list of any items that are already assigned to the container, if any.
 - c. Select one or multiple items to assign to the container.
 - d. Save your settings and verify the items are now listed as being assigned to the container.

Converting an existing class to a container class

This topic describes how to define an existing class as a container class.

About this task

Use the “Classes/Items Manager” on page 47 portlet to edit an existing class so that it can contain other items and containers.

1. Log in to the WebSphere Application Server administrative console and click **Tag Registration** → **Classes/Items Manager** to access this page.
2. Click the **Class Details View** tab.
3. Click an existing class to modify its properties and specify it as a container class.
4. Select **Container** to specify the class can contain items.
5. Make sure to enter the correct spatial measurements in **Edge Length** for the size of the containers in the class.
6. Save your settings.

Results

Any item defined for the class can now contain other items.

Removing the container property from the class

This topic describes how to specify that items belonging to the class can no longer contain other items.

About this task

Perform the following steps in the “Classes/Items Manager” on page 47 portlet.

1. Log in to the WebSphere Application Server administrative console and click **Tag Registration** → **Classes/Items** to access this page.
2. Click the **Class Details View** tab.
3. Click an existing class to modify its properties and specify that it is no longer a container class.
4. Deselect **Container** to specify the class can no longer contain items.
5. Save your settings.

Results

Any item defined for the class can no longer contain other items. Existing container relationships will be dissolved.

Importing a containment relationship

This topic describes how to import containment relationships with a CSV file.

Before you begin

You can import items into a container using a CSV file. However, before completing these steps, make sure the container item that you are adding items to has already been defined.

1. Configure the properties in `Data_Export.properties`, `ClassMapping.properties`, and `class_name.properties` as specified in “Importing resource data to Location Awareness Services for WebSphere Sensor Events” on page 70. In the `class_name.properties` file, in addition to the other necessary values, set the following values for `attribN`:
 - Enter `attribN=EdgeLength` to map to the column in the CSV file that contains the item’s edge length, if it has been specified.

- Enter `attribN=ContainerTagId` to map to the column of the CSV file that indicates the tag ID of the container item.
 - Enter `attribN=removeFromContainer` to map to the column of the CSV file that indicates whether the item should be removed from the container item. The value in the column must be set to yes for the item to be removed.
2. Then run the data import application, as specified in “Importing resource data to Location Awareness Services for WebSphere Sensor Events” on page 70.

Sending events to establish containment relationships

This topic describes how external events can be sent, such as by a Data Capture and Delivery device, to establish relationships.

Before you begin

In order to send external events that can define a containment relationship, you must have defined an event provider with the proper converter.

About this task

The XML of the event must contain the following information:

- The tag ID of the container is indicated in the `location` attribute of the event.
- The tag ID of the contained item is indicated in the `tagid` attribute of `rfid-tag-data`.

Example

For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<ibmprem:ibm-premises-unified-format xmlns:ibmprem="http://www.ibm.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  dts="2007-01-29T11:18:22" messageId="Event_117008750248226334"
  xsi:schemaLocation="http://www.ibm.com IBMPremisesUnifiedMessageFormat.xsd">
  <event eventId="Event_117008750248226334" location="Erff1"
    type="tag_read_external">
    <argument name="sessionId"
      value="L1501170066396687|1170066406828" />
    <argument name="direction" value="world2forklift" />
    <rfid-tag-data antenna="0" count="1" discovered="1170066396890"
      reader="R12" tagid="331505d7941f7900000003a5" />
  </event>
</ibmprem:ibm-premises-unified-format>
```

Evacuating locations

In case there is an emergency in a location and it is necessary to track the evacuation of employees from endangered zones, Location Awareness Services for WebSphere Sensor Events provides an evacuation view that allows companies to monitor tagged items. The evacuation view displays all the zones in the selected area and the number of tags within the zones.

About this task

To monitor evacuation of a location, follow this procedure.

1. Open the Spatial Management Client.
2. Click **Evacuation View**. The Evacuation View window opens and shows the zones within the selected area.

Note: If you want to view an evacuation view for another area, you can open another instance of the Spatial Management Client.

3. Click a zone to expand details about the tags within the zone. The tags shown within the zones will be filtered based on any search criteria you specify in the main view.

What to do next

You can also open the evacuation view when you are replaying data. This allows you to view evacuation patterns and response times by replaying and pausing data from specific areas and zones.

Troubleshooting

This section includes topics for troubleshooting Location Awareness Services for WebSphere Sensor Events.

In addition to the troubleshooting tips in this section, the following technical resources are available to help you answer questions and resolve problems:

- Location Awareness Services for WebSphere Sensor Events version 6.1.x technotes and APARs (problem reports)
- Searching knowledge bases

Logging

If you are unable to resolve technical problems, this topic describes log files, and the process for gathering log files and sending them to your IBM services representative.

Logging levels

The following table lists the available log levels and their meanings.

Table 23. Available log levels and descriptions

Log level	Description
OFF	No events are logged.
FATAL	Task cannot continue and component cannot function.
ERROR	Task cannot continue, but component can still function.
WARN	Potential error or impending error.
INFO	General information outlining overall task progress.
CONFIG	Configuration change or status.
DETAIL	General information detailing subtask progress.
FINE	Trace information - general trace + method entry / exit / return values.
FINER	Trace information - detailed trace.
FINEST	Trace information - a more detailed trace that includes all the detail that is needed to debug problems.

Table 23. Available log levels and descriptions (continued)

Log level	Description
ALL	All events are logged. Can provide a more detailed trace than FINEST.

A message is logged if the logging request's priority is greater than or equal to the currently assigned priority of the utilized logger. For example, if you have a logger with an assigned log level of INFO, then an ERROR request will pass the approval, but a DETAIL request will be blocked. Level ALL causes a logger to accept all logging request; OFF blocks all requests.

Configuring logging

For Location Awareness Services for WebSphere Sensor Events, logging is performed within the Apache log4j framework that has been designed for simplicity and performance.

Follow these steps for configuring logging:

1. Modify the `WAS_HOME\lib\ext\log4j.properties` file to modify logging for the Location Awareness Services for WebSphere Sensor Events components.
2. Modify the LogJSR168Default system property for portlet logging.
 - a. Open the WebSphere Application Server administrative console.
 - b. Click **System Properties**.
 - c. Set the LogJSR168Default property to false by unchecking the box. This ensures that portlet logging uses log4j instead of writing some messages to the `SystemOut.log` file.
3. Restart WebSphere Application Server.

Log files are stored in the `WAS_HOME\profiles\profile_name\logs` directory.

For more information on how to configure logging, consult the log4j documentation available at: <http://logging.apache.org/log4j/docs/documentation.html>

Gathering logs

1. Collect the following log files (for example, as .zip files):
 - For WebSphere Application Server:
 - `WAS_PROFILE_HOME\logs\General.log`
 - `WAS_PROFILE_HOME\logs\Atlas.log`
 - `WAS_PROFILE_HOME\logs\AtlasException.log`
 - `WAS_PROFILE_HOME\logs\server\SystemOut.log`
 - `WAS_PROFILE_HOME\logs\server\SystemErr.log`
 Additional logs are found in the `WAS_PROFILE_HOME\logs\server1` directory.
 - For the Spatial Management Client: `IHS_HOME\htdocs\en_US\Tracking GUI\logs`
2. Send the files to your IBM services representative.

Turning off logging

Turning off all logging is not recommended. It is strongly recommended that you at least keep the ExceptionLogger turned on; however, if you want to turn logging off, perform the following steps:

- Set the rootLogger to OFF.
- To turn off the com.ibm.atlas logger, set its level to OFF.
- To turn off the com.ibm.atlas.exception.ExceptionLogger, set its level to OFF.

Handling alerts for Location Awareness Services for WebSphere Sensor Events event providers and receivers

This topic describes how to set up Location Awareness Services for WebSphere Sensor Events to handle alerts.

About this task

When alerts are generated for third-party Location Awareness Services for WebSphere Sensor Events event providers and devices, D packet messages are generated, which in turn generate events that are sent to the CEI event database. These messages are classified as type AtlasAloeInfraStructure.

To set up Location Awareness Services for WebSphere Sensor Events to handle this type of alert, do the following:

1. Define a notification channel that refers to the messages of type AtlasAloeInfraStructure.
2. Define a notification program to handle these messages.
3. Define it to use the notification channel and to call the e-mail program, but define a specific e-mail receiver to handle these diagnostic messages by selecting the type AtlasAloeInfraStructure as the alert type.
4. Filter on the details related to the messages of type AtlasAloeInfraStructure. All other D packet messages are ignored.

General troubleshooting tips

This topic describes problems that might occur and provides possible solutions.

- “Something is wrong with Location Awareness Services for WebSphere Sensor Events and I do not understand the problem” on page 131
- “Exceptions in the WAS_PROFILE_HOME\logs directory” on page 131
- “My system did not automatically reconnect after a network failure and I did not receive a fatal error telling me to restart my browser” on page 131
- “The browser window hangs up and then the browser crashes” on page 132
- “I had a browser error, but refreshing the page did not correct the problem” on page 132
- “I had a browser error message, and I selected the attempt to recover option. But it did not correct the problem, and I got the error message again.” on page 132
- “I cannot stop server1 using the GUI menu” on page 132
- “Tag processing does not seem to stop” on page 132
- “Chinese characters are not displayed properly on English Windows 2003 Server operating system” on page 133
- “Event information might contain inconsistent times in event date and message” on page 133

- “Rule violation detected with some delay” on page 133
- “Exception in the SystemOut.log file but no obvious failure” on page 133
- “Not all socket hub connections can start” on page 134
- “SQL error message in LOG_ATLASDB_CreateStProc.txt file after installing Location Awareness Services for WebSphere Sensor Events” on page 134
- “General exception when working with the Control Processing portlet” on page 134

Something is wrong with Location Awareness Services for WebSphere Sensor Events and I do not understand the problem

Verify that all of the path settings in the System Properties portlet are correct.

Exceptions in the `WAS_PROFILE_HOME\logs` directory

Multiple `log4j-1.2.13.jar` files

If you see an exception in the `WAS_PROFILE_HOME\logs` file that looks similar to this example, then a possible cause for this exception is that there is more than one copy of the `log4j-1.2.13.jar` file:

```
[31.10.06 16:43:25:246 CET] 0000000a SystemErr
  R log4j:WARN custom level class [com.ibm.atlas.logging.AtlasLevel]
  does not have a constructor which takes one string parameter
[31.10.06 16:43:25:246 CET] 0000000a SystemErr
  R java.lang.NoSuchMethodException: com.ibm.atlas.logging.AtlasLevel.toLevel
  (java.lang.String, org.apache.log4j.Level)
  at java.lang.Class.getMethod(Class.java:1078)
  at org.apache.log4j.helpers.OptionConverter.toLevel(OptionConverter.java:209)
  at org.apache.log4j.PropertyConfigurator.parseCategory(PropertyConfigurator.java:588)
  at org.apache.log4j.PropertyConfigurator.parseCatsAndRenderers
  (PropertyConfigurator.java:524)
  at org.apache.log4j.PropertyConfigurator.doConfigure(PropertyConfigurator.java:408)
  at org.apache.log4j.PropertyConfigurator.doConfigure(PropertyConfigurator.java:432)
  at org.apache.log4j.helpers.OptionConverter.selectAndConfigure
  (OptionConverter.java:460)
  at org.apache.log4j.LogManager.<clinit>(LogManager.java:113)
  at org.apache.log4j.xml.DOMConfigurator.configure(DOMConfigurator.java:543)
  at com.screamingmedia.openportlet.common.log.Log4jSvr.init(Log4jSvr.java:52)
  at javax.servlet.GenericServlet.init(GenericServlet.java:256)
  at com.ibm.ws.webcontainer.servlet.ServletWrapper.init(ServletWrapper.java:275)
  at com.ibm.ws.webcontainer.servlet.ServletWrapper.initialize(ServletWrapper.java:1400)
  at com.ibm.wsspi.webcontainer.extension.WebExtensionProcessor.createServletWrapper(
  WebExtensionProcessor.java:86)
  at com.ibm.ws.webcontainer.webapp.WebApp.getServletWrapper(WebApp.java:793)
  at com.ibm.ws.webcontainer.webapp.WebApp.initializeTargetMappings(WebApp.java:520)
  at com.ibm.ws.webcontainer.webapp.WebApp.initialize(WebApp.java:409)
  at com.ibm.ws.webcontainer.webapp.WebGroup.addWebApplication(WebGroup.java:115)
  at com.ibm.ws.webcontainer.VirtualHost.addWebApplication(VirtualHost.java:128)
  at com.ibm.ws.webcontainer.WebContainer.addWebApp(WebContainer.java:939)
  at com.ibm.ws.webcontainer.WebContainer.addWebApplication(WebContainer.java:892)
  at com.ibm.ws.runtime.component.WebContainerImpl.install(WebContainerImpl.java:167)
  at com.ibm.ws.runtime.component.WebContainerImpl.start(WebContainerImpl.java:391)
  at com.ibm.ws.runtime.component.ApplicationMgrImpl.start(ApplicationMgrImpl.java:1228)
  at com.ibm.ws.runtime.component.DeployedApplicationImpl.fireDeployedObjectStart
  (DeployedApplicationImpl.java:1067)
```

My system did not automatically reconnect after a network failure and I did not receive a fatal error telling me to restart my browser

It is possible that the network retry values are set for an excessively long period of time. In theory, there are no maximum values for `networkRetryInterval`, `maxNetworkRetries`, or `maxNoResponseDisplayManager`; however, if you set the at numbers that are too high, the recovery system tries for a long time. The values are used in two formulae:

- `networkRetryInterval` x `maxNetworkRetries` = The time spent trying to reconnect to the network before giving up.
- `maxNoResponseDisplayManager` = The number of times the software attempts to read tag data from the server before giving up and sending a fatal error.
This value should be no greater than 60,000 ms (the number of seconds to wait).

Open the `IHS_HOME\htdocs\en_us\Tracking GUI\xml\prefsV3.xml` file with a text editor and reduce the values for the following parameters:

- **`networkRetryInterval ms`** = - The frequency of retry attempts if the network connection fails. The default is 30,000 ms.
- **`maxNetworkRetries attempts`** = - The maximum number of attempts before a fatal error displays. The default is 4.
- **`maxNoResponseDisplayManager attempts`** = - The maximum number of "no response" attempts that the Display Manager will tolerate before checking for a network connection failure. The default is 15.

The browser window hangs up and then the browser crashes

Location Awareness Services for WebSphere Sensor Events may have crashed. Restart the browser.

I had a browser error, but refreshing the page did not correct the problem

The application attempts to perform error recovery but it is not always possible to recover from an error. Restart the browser.

I had a browser error message, and I selected the attempt to recover option. But it did not correct the problem, and I got the error message again.

The application attempts to perform error recovery but it is not always possible to recover from an error. Restart the browser.

I cannot stop server1 using the GUI menu

Try using the command line interface:

1. Navigate to the `WAS_PROFILE_HOME\bin` directory.
2. From a command prompt, issue the following command to stop WebSphere Application Server:

Note: Keep in mind that the user IDs and passwords could be different on your system. You do not have to specify user and password, if WebSphere Application Server security is not enabled.

```
stopServer server1 -username wpsbind -password wpsbind
```

Tag processing does not seem to stop

If you stopped tag processing on the Control Processing portlet in the WebSphere Application Server administrative console and the tags are still moving on the Spatial Management Client or you can see that Location Awareness Services for WebSphere Sensor Events is still retrieving events from the dispatcher, do the following:

1. Stop the dispatcher, if you are using it.

2. Stop tag processing again.
3. To restart tag processing with the dispatcher, start the dispatcher before starting tag processing.

If necessary, repeat the steps.

Chinese characters are not displayed properly on English Windows 2003 Server operating system

Chinese characters (or any other non-standard ASCII characters) can be displayed after installing the corresponding languages.

Event information might contain inconsistent times in event date and message

If the location event information contains inconsistent times in the event date and message, the problem might occur because the DB2 server time and the WebSphere Application Server time are not synchronized. In order to solve this problem, prior to running your configuration, it is recommended that you synchronize these server times because location events use the DB2 server time for event creation, but CEI (Common Event Infrastructure) events use the WebSphere Application Server time for event creation.

The following is an example of event information that contains inconsistent times in the event date and message:

Event Date

Fri Feb 22 **14:12:41** CET 2008

Event Type

LasZoneEntry

Event Message

Tag [00000007] with label [] entered zone
[abc1234567d] at [Fri Feb 22 **11:12:41** CET 2008]
inadmittedly. Details: Classes: [New Class?], Groups:
[Printer?]

Rule violation detected with some delay

If **Duration of Stay in Zone** or **Visitor Escorting** rule violations are detected with some delay, check whether their respective **Maximum duration of stay** or **Maximum tolerated rule violation time** values are less than 30 seconds. If either of these timeout values are less than 30 seconds, change the settings of the scheduler for the **Business Rules** engine. To change the settings of the scheduler for the **Business Rules** engine:

1. From the WebSphere Application Server administrative console, navigate to **Resources** → **Schedulers** → **AMITSCHEDULER**.
2. Set the **Poll interval** parameter to the minimum value for the **Maximum duration of stay** and the **Maximum tolerated rule violation time** parameters in your rule instances.

Exception in the SystemOut.log file but no obvious failure

After you start tag processing and successfully access and use the Spatial Management Client, you could see an exception similar to the following example:

```
[8/13/08 10:30:22:421 EDT] 00000037 SRTServletReq E
SRVE0133E: An error occurred while parsing parameters. java.io.IOException:
Async IO operation failed, reason: RC: 55 The specified network resource or
device is no longer available.
```

```
at com.ibm.io.async.ResultHandler.runEventProcessingLoop(ResultHandler.java:671)
at com.ibm.io.async.ResultHandler$2.run(ResultHandler.java:873)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1473)
```

This is a known limitation in WebSphere Application Server, and requires no action at this time. See the APAR description for this issue at: <http://www.ibm.com/support/docview.wss?uid=swg1PK72336>.

Not all socket hub connections can start

If many of your event providers are defined in Location Awareness Services for WebSphere Sensor Events with socket hub connections, but you can only connect to a few of them, check the work manager settings for the JNDI name, `wm/IBMATlas`. Modify the **Maximum number of threads** value to the number of event providers you have multiplied by three.

SQL error message in LOG_ATLASDB_CreateStProc.txt file after installing Location Awareness Services for WebSphere Sensor Events

If you are installing Location Awareness Services for WebSphere Sensor Events 6.1.0.2 without a previous version of the component installed, you could see an error in the `LAS_HOME\logs\LOG_ATLASDB_CreateStProc.txt` file that is similar to the following:

```
DROP PROCEDURE IBMATLAS.SEARCHDATA (INTEGER, TIMESTAMP)
DB21034E The command was processed as an SQL statement because it was not a
valid Command Line Processor command. During SQL processing it returned:
SQL0458N In a reference to routine "IBMATLAS.SEARCHDATA" by signature, a
matching routine could not be found. SQLSTATE=42883

DROP PROCEDURE IBMATLAS.HEADCOUNTBYZONE (TIMESTAMP, CHAR(1))
DB21034E The command was processed as an SQL statement because it was not a
valid Command Line Processor command. During SQL processing it returned:
SQL0458N In a reference to routine "IBMATLAS.HEADCOUNTBYZONE" by signature, a matching routine
could not be found. SQLSTATE=42883

DROP PROCEDURE IBMATLAS.BATLIFEREPORT ()
DB21034E The command was processed as an SQL statement because it was not a
valid Command Line Processor command. During SQL processing it returned:
SQL0458N In a reference to routine "IBMATLAS.BATLIFEREPORT" by signature, a
matching routine could not be found. SQLSTATE=42883

DROP PROCEDURE IBMATLAS.LASTSEENREPORT (TIMESTAMP, CHAR(1), VARCHAR(256), VARCHAR(256),
VARCHAR(32))
DB21034E The command was processed as an SQL statement because it was not a
valid Command Line Processor command. During SQL processing it returned:
SQL0458N In a reference to routine "IBMATLAS.LASTSEENREPORT" by signature, a matching routine
could not be found. SQLSTATE=42883
```

You can safely ignore this error.

General exception when working with the Control Processing portlet

If you have Location Awareness Services for WebSphere Sensor Events security enabled and you are working with the Control Processing portlet, you could see a general exception. As long as the status of the event providers are changing as expected, then you can safely ignore the exception.

Troubleshooting the Spatial Management Client

This topic describes problems that might occur with the Spatial Management Client and provides possible solutions.

- “The Spatial Management Client does not completely start”
- “Search results are not saved”
- “Personal preferences are not saved”
- “Tags are still visible when they have already left the area, but the tag counts seem to be correct”
- “Error message when using a new area in the Spatial Management Client” on page 136

The Spatial Management Client does not completely start

If you cannot see the map of the current area, and you cannot see Zones in the Zones list (the Zones list basically is empty), then make sure that the Adobe Scalable Vector Graphics (SVG) Viewer plug-in for your browser is installed.

Search results are not saved

If you are using the Spatial Management Client on a Windows operating system and your search results are not properly saving as HTML, enable ActiveX in Internet Explorer:

1. In the browser, navigate to **Tools → Internet Options**.
2. Select the **Security** tab.
3. Click **Custom Level**.
4. Scroll down to **ActiveX controls and plug-ins → Initialize and script ActiveX controls not marked as safe**, and click **Enable** or click **Prompt** if you would like to be prompted with a confirmation window in order to save the search results.
5. Click **Ok**, and then click **Ok** again.

There are also two workarounds you can use if you do not choose to enable ActiveX:

- Use the Search portlet in the WebSphere Application Server administrative console and save the results as HTML.
- Use the Spatial Management Client and save the results in XML format.

Personal preferences are not saved

There are limitations in saving your personal preferences for the Spatial Management Client:

- Selected areas are not saved to your user preference. Instead, the Spatial Management Client always shows the area in the sequence defined in the `prefsV3.xml` file.
- The selected tag filter is not saved as your user preference. Logging in with the same user ID always starts with a tag filter of **All**.

Tags are still visible when they have already left the area, but the tag counts seem to be correct

If you have this issue, you need to add or modify the value of the `<DisplayRefreshCounter>` parameter in the `prefsV3.xml` file. The

<DisplayRefreshCounter> parameter forces a repainting of all the tags in the Spatial Management Client every *nth* poll interval (<pollInterval ms="">).

For example, a setting of <DisplayRefreshCounter>50</DisplayRefreshCounter> means that the Spatial Management Client repaints every 50th poll. If you have a poll interval of 3000 (<pollInterval ms="3000">), then the Spatial Management Client repaints every 150th second.

If you have a lot of tags on the Spatial Management Client, repainting too often increases the load on your system. To avoid overloading, set the parameters to update every 120 seconds, or less often than that.

Error message when using a new area in the Spatial Management Client

If you create a new area using the Preferences Administration GUI, you may receive an error message when you switch to the new area in the Spatial Management Client.

Use these steps to resolve this problem:

1. Close all browser windows after creating the area in the Preferences Administration GUI.
2. Wait for at least 60 seconds before opening the new area in the Spatial Management Client to allow for all caches to refresh on the server side.

When you reopen the Spatial Management Client browser window, you should be able to access the new area.

Messages

This section explains each element in the message line of messages. It also describes the troubleshooting components of message descriptions and presents a list of messages, each of which includes the following descriptive information:

- Message
- Explanation
- Response

The messages are arranged in numeric order, according to the message number.

Message text components

This topic explains the text components of a message.

ATL000E DD MISSING. TERMINATING.

Number/Severity

- ATL0000 is the unique number for this message.
 - E is the severity level code for the message.
- See "Severity code levels for messages" on page 137.

Message text

DD MISSING. TERMINATING. The text explains the reason for the message. It might also include possible causes and system or user actions. In this example, the system is taking the action to terminate the process.

Troubleshooting components of messages

This topic describes the troubleshooting components of a message.

Message

Example: ATL000E CONFIGURATION MISSING. TERMINATING.

Explanation:

Describes what caused the message.

- Examples of an explanation for this message:
Mail host configuration is missing. E-mails cannot be sent.
To send emails, a mail host configuration must be defined. Processing terminates.
- Examples of possible explanations for other messages:
- The name in the field member is not valid. The naming conventions are:
The name must be 1 to 8 alphanumeric characters. Correct for the next run.
- A number parsing exception occurred. This happens if, for example, a letter was entered in a number field. The intended action was not performed.

System action:

Describes what the system does.

System action for this message: Processing terminates.

Examples of system action include:

- Processing terminates.
- Processing continues.

Response:

Describes what you must do to proceed, to recover from the error, or to avoid a problem

- Example of the User Response for this message:
Either delete the notification channel pointing to the email program or configure the mail host.
- Examples of a possible User Response for other messages:
 - Either delete the notification channel pointing to the email program or define a mail receiver. If you do not need email notification for this time and event, ignore the message.
 - Ask if one of your colleagues accidentally deleted this record or look to see if a record that was referenced by this record was deleted

Severity code levels for messages

This topic explains severity code levels used in messages.

Severity code indicators are:

- I (informational)
- W (warning)
- E (error)

I Informational. Provides users with information or feedback about typical events that have occurred or are occurring or requests information from users in situations where the outcome will not be negative, regardless of the response.

Examples:

- The status request is processing.
- The files were successfully transferred.
- Do you want to save the output in file log.txt or in file error.txt?

W Warning. Indicates that potential problem conditions have occurred or could occur, but the program can continue. Warning messages often ask you to make decisions before processing continues.

Examples:

- The resource tahiti.com was not found. Processing will continue.
- A file already exists with the name logfile.txt. Do you want to overwrite this file?

E Error. Indicates problems that require intervention or correction before the program can continue. The typical result of Error messages is that processing terminates.

Examples:

- The file logfile.txt was not found and is required for processing.
- There is no space on the C drive. The file logfile.txt cannot be saved to this drive.

ATL01001E NumberParsingException {0} occurred.

Explanation: A number parsing exception occurred. This happens if, for example, a letter was entered in a number field. The intended action was not performed.

System action: Processing terminates

User response: Verify and correct the values. Then retry the operation.

ATL08001E A RuntimeException {0} occurred.

Explanation: A RuntimeException occurred in the Database Access Layer. The intended action was not performed. This is an internal error.

User response: Retry the operation. If the error persists, report it to your IBM representative.

ATL08002E A general database exception occurred. SQLcode: {0}, SQLstate {1}

Explanation: An unhandled database exception occurred. The intended action was not performed. Review the log files for more detailed information.

User response: Contact your database administrator.

ATL08003E A lock, deadlock, or timeout exception occurred. SQLcode: {0}, SQLstate: {1}, cause: {2}

Explanation: The intended action was not performed due to a lock, deadlock, or timeout. Review the log files for detailed information.

User response: Retry the operation.

ATL08004E Update of an entry on table {0} with key values {1} was based on an outdated version.

Explanation: The performed update operation failed due to an outdated data record. This happened because another process had already updated the same data record.

User response: Reload the data. Then retry the operation.

ATL08005E The insert or update operation violated the foreign key constraint {0}. SQLcode: {1}, SQLstate: {2}

Explanation: The value of the foreign key, which was passed to the abortive insert or update operation, did not match any parent key of the parent table. This happened because another process updated or deleted the referenced data record of the parent table before.

User response: Correct the value and try again.

ATL08006E The insert or update operation violated a primary key, unique constraint, or unique index for column(s) {0} of table {1}. SQLcode: {2}, SQLstate: {3} .

Explanation: The value of the primary key, which was passed to the abortive insert or update operation, violated a primary key, unique constraint, or unique index constraint. This happened either because another process updated or inserted a data row using the values or because an update operation was performed by using values that are already in the database.

User response: Correct the value and try again.

ATL08007E The insert or update operation violated the check constraint {0}. SQLcode: {1}, SQLstate: {2}

Explanation: The performed insert or update operation violated a defined check constraint.

User response: Correct the value and try again.

ATL08008E Assigning a 'NULL' value to a 'NOT NULL' column {0} is not allowed. SQLcode: {1}, SQLstate: {2}

Explanation: The value, which was passed to the abortive insert or update operation, was 'NULL', but the object column was declared as 'NOT NULL' in the table definition.

User response: Correct the value and try again.

ATL08009E Assgined value is too long or too large. SQLcode: {0}, SQLstate: {1}

Explanation: A value, which was passed to the abortive select, insert or update operation, was too long or too large.

User response: Correct the value(s) and try again.

GENERAL A general exception has occurred.

Explanation: An exception has occurred. No additional information is available.

User response: Contact the System Administrator for further information.

DBEXCEPTIONINPORTLET A database exception has occurred {0}.

Explanation: The system tried to access the database and an error occurred.

User response: Contact the System Administrator for further information.

SQLException A SQL exception has occurred.

Explanation: A SQL exception has occurred and is wrapped by an Atlas data base exception.

User response: Contact your database administrator.

RUNTIMEDBEXCEPTION A Runtime Exception has occurred.

Explanation: A Runtime Exception has occurred within the Database Access Layer.

User response: Contact your Database Administrator.

JMSEXCEPTION A JMS Exception has occurred ({0}).

Explanation: Communication with the remote legacy system has failed.

User response: Verify the configuration for the service bus.

GENERALDBEXCEPTION A general database exception has occurred: {0}.

Explanation: A general database exception has occurred. No additional information is available.

User response: Contact the System Administrator or the Database Administrator for further information.

IMPORTEXCEPTION Error in an ATLAS import operation.

Explanation: See message.

User response: Correct errors and try again.

INVALIDINPUTEXCEPTION Invalid input: {0}

Explanation: You have specified a value that is not valid.

User response: Enter the correct value and try again.

REGUNITCONNECTIONEXCEPTION An exception occurred when working with the registration-unit. {0}

Explanation: See message.

User response: Verify that:

- Your registration-unit is running
- The correct ip-address and port are specified in the RegistrationUnits-Portlet.

GENERALCEIEXC An exception related to the Common Event Infrastructure has occurred. It is not possible to send, get or update events.

Explanation: Communication with the event database or the event emitter failed.

User response: Ask your System Administrator to review the Common Event Infrastructure configuration.

CEIEVENTNOSND

Explanation: CEI events cannot be sent.

System action: This can be due to configuration errors in the event database, the CEI server application, or the underlying service bus.

User response: Ask your System Administrator to verify the Common Event Infrastructure configuration.

CEIEVENTNOGET CEI events cannot be retrieved.

Explanation: This can be due to configuration errors in the event database or in the CEI server application.

User response: Ask your System Administrator to verify the Common Event Infrastructure configuration.

CEIEVENTNOCHG CEI events cannot be updated to reflect alert handling.

Explanation: This can be due to configuration errors in the event database, the CEI server application, or the underlying service bus.

User response: Ask your System Administrator to verify the Common Event Infrastructure configuration.

XPATHSELECTOR The channel selector to determine the events of interest is not usable. The notification channel will be ignored.

Explanation: This might be due to a version inconsistency or an incorrect manual edit of the selector.

User response: Delete the notification channel and add it again, using the dialog.

FILEIOEXCEPTION An exception occurred when writing file '{0}'.

Explanation: See message.

User response: Contact your System Administrator.

INTERNAL An internal error has occurred ({0}).

Explanation: The intended action was not performed because of an internal error that is not covered.

User response: Try the operation again. If the error persists, report it to your IBM representative.

KEYPROPERTYNOTDELETABLE You cannot delete key-property {0} because {1} are existing.

Explanation: Objects exist that depend on the key property you want to delete.

User response: First delete the corresponding objects. Then delete the key property.

UPDATEABLERECORDDELETED Your update on {0} could not be saved because {0} was deleted by someone else.

Explanation: While you were editing the record, someone else deleted it.

User response: Speak with your colleagues and ask if someone accidentally deleted your record.

FILENOTFOUND The specified file or path {0} could not be found.

Explanation: Either the entered file or path does not exist or access is denied because of missing authorizations.

User response: Verify that you entered the correct file or path and that you have the required access authorizations.

ZIPEXCEPTION Either the .zip file {0} could not be opened or the specified path {1} could not be found.

Explanation: The entered file path does not exist. Either the .zip file is damaged or access is denied because of missing authorizations.

User response: Verify that you entered the correct file path and retry the operation.

DUPLICATEKEYPROPERTIES An item already exists with the same values in its key properties.

Explanation: An item is already defined that has same values in all of its key properties.

User response: Verify that all key property values for this item are correct.

ATL01002W None of the search criteria were selected.

Explanation: None of the required selection criteria were selected. One of the search criteria checkboxes, Class Properties, Group Properties, or Tag Properties must be selected. .

User response: Select at least one of the search criteria and try the operation again.

ATL15001E Sender address {0} is not a valid internet address.

Explanation: The address must be in the xxx@yyy format.

User response: Correct the email address in the configuration for the mail host.

ATL15002E Mail host configuration is missing. E-mails cannot be sent.

Explanation: To send mails, a mail host configuration must be defined.

User response: Either delete the notification channel pointing to the email program or configure the mail host.

ATL15003E

Explanation: Mail host configuration data not accessible.

System action: The table for the mail host configuration was not accessible.

User response: Inform the System Administrator.

ATL15004E Mail receiver data not accessible.

Explanation: The table for mail receivers was not accessible.

User response: Inform the System Administrator.

ATL15005E

Explanation: One of the receiver addresses is not a valid internet address: {0}

System action: The address must be in the xxx@yyy format.

User response: Correct the receiver email address in the configuration.

ATL15006E Mail could not be sent.

Explanation: The email could not be sent because of the reason specified.

User response: Correct the receiver email address in the configuration.

ATL15007W Mail cannot be sent since no receiver addresses defined.

Explanation: To send mails a mail receiver must be defined for this time and event type.

User response: Either delete the notification channel pointing to the email program or define a mail receiver. If you do not need email notification for this time and event, ignore the message.

ATL15008W No mail receivers available.

Explanation: To send mails, a mail receiver must be defined for this time and event type.

User response: Either delete the notification channel pointing to the email program or define a mail receiver. If you do not need email notification for this time and event, ignore the message.

CANTEDITBECAUSEDELETED You cannot edit the {0} because it was already deleted.

Explanation: The record you want to edit has already been deleted. Either this was done by someone else or it was done indirectly because something was deleted that is referenced by this record; and therefore, this record was also deleted. .

User response: Ask if one of your colleagues accidentally deleted this record or look to see if a record that was referenced by this record was deleted

Glossary

This topic lists terms that are used in this documentation.

Alarm zone

Zone where access restriction rules or similar rules can be triggered when an item (usually person or asset) enters this type of zone. Restriction rules can also be set for other zone types, but they have additional semantics.

Area

A representation of the real physical space within the location to be monitored. Areas have a flat lower and an optionally flat upper boundary and are the container for all zones.

Boundary zone

Zone used to monitor tags entering and leaving another zone.

Container

Items that can contain other items. Tags that are added to a container inherit the position of the container. Also you can define the size of a container using the EDGELENGTH system property, which assumes the container is a cube.

Device

A device is used for an event provider to provide location, notification, or telemetry data. Devices always belong to a hub and can be grouped in device groups.

Event converter

Applications that convert external events into a format that Location Awareness Services for WebSphere Sensor Events can process. Event converters are specific to each event provider and can be provided as plug-ins.

Event database

A database that stores all events that are sent by Location Awareness Services for WebSphere Sensor Events.

Event device

A third party that receives the tag signal and transmits the data to an event provider.

Event group

A group of related messages. A filter is defined to route certain types of alert messages to the related message queue.

Event provider

A third party that monitors areas and feeds Location Awareness Services for WebSphere Sensor Events tag data. Event providers are not part of Location Awareness Services for WebSphere Sensor Events, so they must be defined within Location Awareness Services for WebSphere Sensor Events. They must be configured for an existing area so that Location Awareness Services for WebSphere Sensor Events can track tags within that area.

Exit zone

Zone used to determine whether a tag has exited the area. If the tag passed and no signals can be received thereafter, the item has left the area and there is no reason to be concerned about not receiving a signal.

Gate An entry or exit to or from a zone that is monitored by one device.

Group A container that allows grouping of items from different classes for common rules, searches, or so forth.

Item Entities within a location that can be equipped with tags and whose positions can therefore be tracked. An example is an asset or person.

Item class

Class of items with common attributes. You can define sets of attributes and rules for each item class. For example, you might have the following classes: Person and Asset. Within these classes, you can also have subclasses with extended attributes.

Location

A real physical space that is made up of many areas.

Notification channel

A channel definition that defines for a given subscriber the program or web service that should be called for an event, and the filter criteria under which the program or web service should be called.

Notification program

A program or web service that can be triggered when an event occurs.

Privacy zones

Currently, privacy zones behaves like alarm zones.

Registration unit

An event provider that, as a whole or with a special part of its

infrastructure, reads tag IDs into the Location Awareness Services for WebSphere Sensor Events system for the purpose of defining an item for the first time.

Rule Criteria or circumstances that are defined to trigger an event. For example, rules can be triggered during entry to or exit from a zone and can be specified for a tag ID, class, or group. You can set the following types of rules in Location Awareness Services for WebSphere Sensor Events:

- Zone entry and exit rules
- Tag not responsive rule
- Tag battery low rule
- Proximity rule
- Unknown tag rule

Shadow zone

Zone where the tags might not be visible temporarily because they are out of reach of the tag reader infrastructure or the signals are shielded. Location Awareness Services for WebSphere Sensor Events assumes that a tag continues to be in the shadow zone at the last reported position after it has been seen. No alert is generated if the tag is no longer visible.

Smoothing algorithm

Algorithm used to smooth position estimates, so that tag icons do not move abruptly in the Spatial Management Client.

Subarea

An area that is nested within another area.

Subscriber program

A program that subscribes all, or a defined subset of, events arriving for a given event group. It dispatches the arriving events to the Location Awareness Services for WebSphere Sensor Events notification programs.

Zone Logical section within an area for which rules can be defined. A zone can span multiple subareas of the area to which it is related. It is the unit on which rules can be performed, and on which counts and statistics for a tag entering or leaving can be calculated. Rules can be defined for entering and exiting a zone.

Index

A

- alerts 89
 - define how to handle 59
 - handle 130
- area names 38
- areas
 - define 19
 - description of 8
- areas and subareas
 - define 19
 - naming 19
- automatic reconnect
 - did not work 132

B

- boundary zones
 - add 44
 - define 45
 - definition of 9

C

- class names 46
- class structure 45
- classes
 - add 47
 - define 47
 - delete 50
 - edit 47
- coordinate transformation samples 25
- coordinates, transforming 21

D

- data
 - backing up 77
 - restoring 77
- data types
 - for importing data 70
- databases
 - backing up offline 78
 - backing up online 77
 - restoring 79
- device groups
 - definition of 7
- devices
 - add 42
 - definition of 7
 - delete 42
 - edit 42
- dispatcher
 - configure 76
 - description of 74
 - start 76
 - stop 76

E

- error recovery did not work
 - browser 132
- event providers
 - add 39
 - conversion method 39
 - definition of 7
 - delete 39
 - edit 39
 - smoothing algorithm 40
 - transformation options 40
- events 89
 - acknowledge 90
 - archive 90
 - delete 90
 - display 89
 - filter 89
 - handle 130
 - types 17
- exceptions
 - constructor error 131
 - WAS_PROFILE_HOME\logs 131

F

- formatting data types 70

G

- gates 43
 - add 43
 - definition of 7, 9
 - delete 43
- glossary 141
- graphic of area
 - converting a bitmap format 20
 - converting a vector format 20
 - converting to SVG 19
 - importing into Spatial Management Client 19
- groups
 - add 51
 - add items 52
 - assign color 51
 - define 51
 - definition of 15
 - delete 51
 - remove items 52
 - view items 52

H

- historical data
 - replay 88
 - searching 91

I

- import resource data 70

- importing data
 - data types and values 70
- item classes
 - definition of 14
- items
 - add 50
 - delete 51
 - edit groups 51
 - edit properties 51
 - view 50

K

- key properties
 - add 49
 - delete 49

M

- mail receiver
 - add 61
 - define 61
 - delete 61
 - edit 61
- mail server
 - add 61
 - define 61
- maxNetworkRetries attempts= 132
- maxNoResponseDisplayManager attempts= 132
- message components 136

N

- network failure
 - no automatic reconnect 132
- networkRetryInterval ms= 132
- notification channels 60
 - add 60
- notification programs
 - add 60
 - define 60
 - definition of 18
 - notification channels
 - definition of 18
 - type of 60

P

- preferences for Spatial Management Client 28
- properties
 - add 49
 - delete 50

R

- registration units
 - add 44

- registration units (*continued*)
 - define 44
 - delete 44
 - edit 44
- replay 88
- reports 92
- reports administration
 - add 92
 - define 92
- reports operation
 - define 93
- resource data, import 70
- restrictions
 - class names 46
 - classes 45
 - new areas and zones 37
- rules
 - add 53
 - define 52
 - delete 53
 - edit 53
 - types 15

S

- search 90
 - by class 91
 - by group 91
 - by tag properties 91
- search tags 90
- severity code levels 137
- smoothing algorithm 40
- Spatial Management Client
 - description of 82
 - description of administration
 - version 31
 - setting preferences 28
 - start administration version 30
- Spatial Management GUI
 - start 81
- stationary items 48
- subareas
 - define 19
- subclasses 14
- subscriber programs
 - definition of 18
- system properties
 - add 62
 - delete 62
 - setting 62
 - table of 62

T

- tag processing 87
 - start 87
 - stop 87
- tags, searching 90
- terms 141
- topology 38
- tracking GUI
 - updating 28
- transformation options 40
- transformation, coordinates 21
- type of properties 49

U

- user interfaces 5

Z

- zone names 38
- zone type
 - alarm 8
 - boundary 9
 - exit 9
 - privacy 8
 - shadow 8
- zones
 - boundary 44
 - define 31
 - description of 8
 - monitoring 11
 - sample layout 12

Readers' Comments — We'd Like to Hear from You

Sensor Events

Location Awareness Services for WebSphere Sensor Events

Version 6.2.0

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.
- Send a fax to the following number: 1-800-227-5088 (US and Canada)

If you would like a response from IBM, please fill in the following information:

Name

Address

Company or Organization

Phone No.

E-mail address



Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Information Development
Department 9BSA
P.O. Box 12195
Research Triangle Park, NC 27709-9990



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line



Printed in USA