Deployment Guide

XARRAY AND XSPAN GATEWAYS

DEPLOYMENT AND BEST PRACTICES

Version 2.0

www.impinj.com

This document is to help RAIN RFID specialists plan and implement Impinj gateway deployments. It provides high-level deployment steps, case examples, and best practices for designing and evaluating xArray- and xSpan-based solutions. The guide is created for use by practitioners who have completed RAIN RFID training from Impinj. More detailed information about setting up hardware and preparing or developing custom software is available on the Gateway support page.

TABLE OF CONTENTS

OVERVIEW

A gateway is an integrated RAIN RFID reader system. Unlike traditional RAIN RFID readers such as the Speedway fixed reader, a gateway requires no additional antennas, cables, or antenna connectors. Not only does it remove the hasle of managing RFID accessories, but it also enables more features due to control of the antenna beams. This guide covers the planning and implementation of solutions that use one or more Impinj gateways.

Gateway Models: xArray & xSpan

The xArray gateway combines the award-winning performance of the Speedway reader with a phased array antenna that synthesizes 52 beams, each of which has both horizontal and vertical polarizations. This combination provides a much wider circular area of monitoring for more tag types and orientations than other commercially available RAIN RFID Systems.

The xSpan gateway also employs a Speedway reader, but with a phased array of 13 polarized beams. Its monitoring area is oblong. xSpan arrays are also designed to be mounted sideways to enhance performance in Direction Mode scenarios.

These Impinj gateways can be combined in deployments to extend coverage area and leverage the best features of both models, broadening the functionality of RAIN RFID solutions for a given facility.

Modes and Coverage Areas

The gateways operate in preset configurations called Modes: Inventory Mode for item inventory, Location Mode for item (x, y) location, and Tag Direction Mode for determining tags' direction of travel. Table 1Table 1 shows the various modes and descriptions.

Table 1: Gateway Configuration Modes

MODE	DESCRIPTION
Inventory	A gateway takes an inventory by reading tags within its field of view (FOV). This configuration simply reports tags seen, when, and by which beam.
Location	The Gateway calculates a tag's (x,y) location within 5 feet (1.5 meters). See Location Accuracy in Concepts to Know. This configuration reports tags seen, when, and where they are based on Impinj location algorithms.
Direction	The Gateway detects the direction of a tag as it transitions across the gateway's field of view (FOV). For example, an xArray installed above the intersection of 2 hallways can track any direction of a tagged item traveling in all 4 directions through the intersection. See Direction Mode: Configuration and Testing

Impinj gateways provide a variety of modes and coverage. Table 2 shows the modes and coverage limitations of each gateway.

Table 2: Modes and Coverage Limitations

MODE	XARRAY	XSPAN
Inventory	52 Polarized beams Coverage: up to 1,500 ft² (140 m²) Circular area**	13 Polarized beams Coverage: up to 1,000 ft² (93 m²) Rectangular area**
Location	tag x and y coordinates	Not supported
Direction	4 directions	2 directions

*** Assumes Monza R6 based tags with larger far field antennas (>50mm) and optimal dielectric.*

IMPINJ SOFTWARE RESOURCES

Impinj has a variety of software resources to help you plan, deploy, and test custom solutions based on Impinj Gateways.

This Gateway Deployment Guide was written based on the functionality provided by Octane firmware version 5.10.

Test and Control Software

Impinj ItemTest[A] is a Windows application for Impinj Gateway partners for use in testing and optimization. ItemTest helps you iteratively test and configure Gateways prior to and during deployments to refine performance. ItemTest allows the ability to visualize and tune raw RAIN RFID parameters. Screenshots from ItemTest appear throughout this document, and basic procedures are described.

Impinj ItemSense[B] is software that provides centralized management of RAIN system components, aggregation of raw RAIN data, and transformation of aggregated data into business-driven event data. ItemSense can greatly simplify gateway deployments. While this deployment guide does not provide any guidance as to how to install or configure ItemSense software, it does provide guidance on how to deploy, tune, and test xArray and xSpan hardware used in an ItemSense deployment.

Application Integration Options

Impinj gateways work with the following options for integration:

- Octane Software Development Kit (SDK) with intuitive class libraries for both Java and C# /.NET
- LLRP Tool Kit (LTK) with C, C++, Java, and .Net variants[C] in the form of numerous Impinj extensions offering granular control of the gateway
- On-gateway applications using C/C++ running directly on the gateway by means of the Impinj Embedded Developer's Toolkit (ETK).

Please visit the support portal Gateway section for the latest LTK and SDK releases and related documentation[D].

DEPLOYMENT SUCCESS FACTORS

A successful gateway deployment requires users to understand a range of external factors that can constrain the use case details and success criteria requirements, including but not limited to: the physical environment, tag read sensitivity, gateway mounting variations, and configuration settings. This document provides guidance for these factors to help you deploy your gateway successfully. You should always assess the deployment results through onsite testing.

One key attribute regarding success in a business application is the read probability, which will determine directionality, location, and inventory accuracies. This probability is a result of the factors described in this document.

Prerequisite Knowledge

This guide assumes that you have a solid, practical background in deploying and using RAIN RFID technologies, and have completed the gateway training program offered by Impinj.

At a minimum, you should have a working knowledge of fundamental RAIN concepts such as Sessions, RF modes of operation, and a basic understanding of current RAIN technology limits. Previous experience with Impinj Speedway readers and attendance of an Impinj FastTrack class is useful, though not required.

This document is intended for use by the following people:

- Integrators
- Software Developers
- RFID Systems Engineers
- Facility Managers/IT Managers

Basic Steps for Gateway Deployment

Generally, most large-scale deployments require some empirical testing of use cases to ensure the required outcome is possible within the customer's typical RF environment. This is referred to as a *proof of concept*. In building your proof of concept you will want to:

1. Plan:
   - Identify your tagged items to quantify how easily your items will be read by the Gateway
2. Design:
   - Calculate the likely gateway coverage to plan where to install your gateways
   - Ensure your use case is aligned with the modes supported by each gateway
3. Implement:
   - Install your gateway
4. Optimize:
   - Test your gateway for best performance based on the following modes:
     - Inventory Mode
     - Location Mode
     - Direction Mode

IDENTIFY YOUR TAGGED ITEMS

The tagged item refers to the combination of an item and RAIN RFID tag attached to it. The tagged item RAIN RFID read performance plays a very important role in a successful gateway deployment. Tag sensitivity for a particular material is key because it directly correlates with the read range. (Reference RFID Link Budget Whitepaper[E] for information regarding the actual read distance.)

TIP: If RAIN RFID tags will be purchased or influenced as part of the deployment, consider using only Monza–based tags for best performance. The features and benefits are outlined in the resources provided by our Monza Learning Center[F]. Refer to your tag manufacturer's input when selecting the best fit tag given your item's material and associated far-field use case

Tag Sensitivity

Tag sensitivity is a critical contributor to the performance of RAIN RFID systems, especially those based on always-on, overhead technologies where long range tag reading (greater than 3m) is needed.

Read range and read margin depend almost entirely on tag sensitivity as detailed in our Link Budget Whitepaper[G]; we recommend every gateway installation use the most sensitive and consistently performing tags. The choice of tag often determines the difference between success and failure for far-field applications.

Tag sensitivity depends on tag integrated circuit (IC) sensitivity, inlay design, and dielectric interactions with the item itself. Some useful generic guidance is included below, but reach out to your Impinj account team for further information and guidance:

- Tags using Inlays of at least 70mm long. Example of vendors with this size inlays include Smartrac, AveryDennison, Invengo and others.
- Monza R6 chip enabled inlays show the best and most consistent read ranges in empirical testing due to advanced features such as Auto-Tune and Endura Pads.
- Verify that your tag is tuned for your region. For example, if your region is ETSI (Europe) and your tags are tuned for FCC (America), tag read range can be considerably reduced.

Use Case Examples

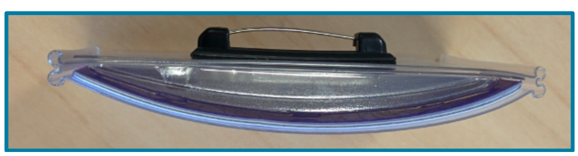
A use case is a specific application of technology; for RAIN RFID, some examples are people tracking, asset tracking, and pallet location, among many others. Generally, a use case defines the tagged item and the intended goal for using RAIN RFID.

People Tracking

When using RAIN RFID to track people, it is important to choose a badge holder that creates space between the tag and the wearer's body will greatly improve tag read performance. This is primarily due to human body impedance.

Generally, the further a tag is from the person's body, the easier it is to read, but a distance of one centimeter provides most of the benefit of this free space. Convex badge holders, such as the example in Figure 1, have good empirical results. If your use case permits it, placing tags on the top of a hard hat or clipped to a collar may assist readability when used with overhead gateway placement. In Impinj's tests, gateway performs well with credit card sized tags.

Figure 1: Convex Badge Holder



Liquid Asset Tracking

Liquid will greatly detune any tag causing it to have a much more limited coverage. If the tagged item contains liquid, using foam padding can minimize detuning. Also, the tag should be placed on the top of the bottle to be "in view" when used with overhead gateway placement. The tags are also placed over the air space in the bottle to distance the tag from the liquid. See Figure 2 for an example of liquid asset tagging.

Figure 2: Foam Tagging for Liquid

Foam padding helps tags on liquid items.

GATEWAY COVERAGE

The xArray Gateway generates up to 52 directed beams. The xSpan Gateway generates up to 13 directed beams. Each beam has two states: a vertically polarized state and a horizontally polarized state.

Coverage Area

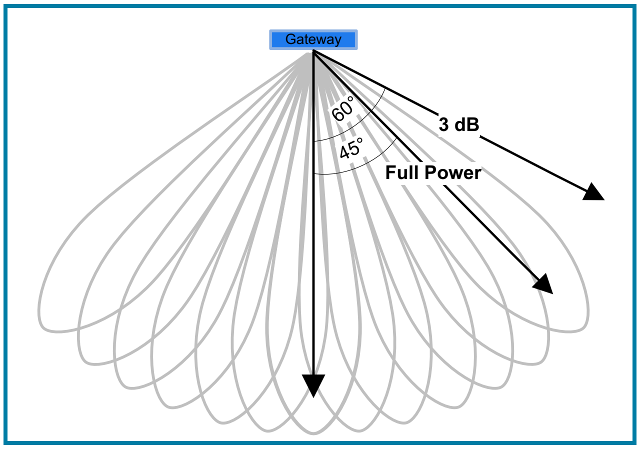
The size of the coverage area is a function of many variables:

- Transmit power from the gateway
- The antenna beam pattern for the gateway
- The distance of the tags from the gateway
- The orientation of each tag
- The dielectric of the material for the tagged item (see Tag Sensitivity)
- The multipath caused by reflective surfaces
- Any object obstructing the RF from the reaching the tags (ex. dry wall, wood shelving)
- As the received power to the tag drops, so does the probability of reading it.

The best approach is to empirically assess your physical environment by evaluating the gateway's coverage area at different heights and make sure that your tags fall within the gateway's coverage. For example, by placing tags on high shelves, you may observe if the proposed gateway placement is able to attain coverage for the highest and lowest shelving (ie, the most challenging coverage areas). These tests may demonstrate if you need additional gateways to improve coverage in the Z-plane.

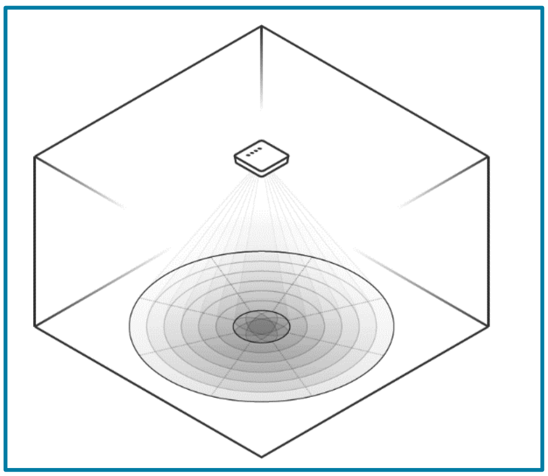
The xArray transmits full power coverage of 90°. Similarly, the xSpan transmits full power coverage of up to 90° along its major axis. The 3 dB beamwidth defines the angle for which the power transmission is within 3 dB of full power. Figure 3 provides a visual sample of full transmit power coverage for either the xArray or xSpan.

Figure 3: Full transmit power coverage area of 90° and 3dB beamwidth at 120°



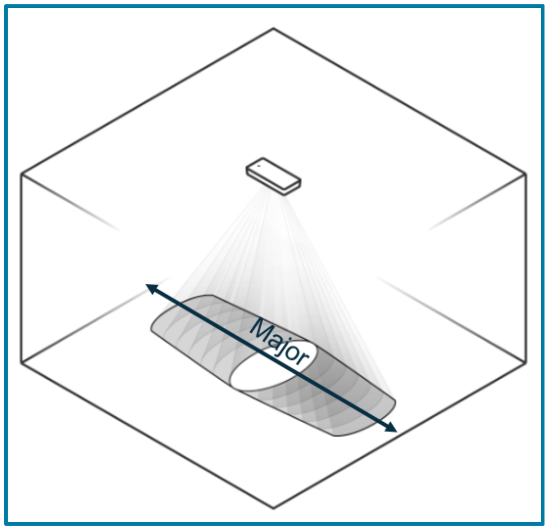
Enabling all 52 antennas of the xArray provides a circular 3 dB beamwidth coverage area of approximately 120°. In other words, the xArray will transmit full power - 3 dB towards a tag that is offset by 60° from directly under it. Figure 4 shows the xArray's circular coverage area.

Figure 4: xArray Beams and Sectors



For the xSpan, enabling all 13 antennas provides an oblong 3 dB beamwidth coverage area approximately 120° major axis and 80° minor axis. Figure 5 demonstrates the shape and size of its coverage area.

Figure 5: xSpan Coverage with Major Axis



Note: Minor axis is perpendicular to major axis

Tags outside the 3 dB beamwidth area can still be read, but with lower probability due to reduced receive power.

In planning your deployment, you will need to decide how much margin to build into your design. The greater the need for a high probability of reading a tag, the more you want to design your system inside of the 3 dB beamwidth area. As an example, many of the location and directionality algorithms need large read counts for items to have higher accuracies. These scenarios require a higher probability read area to enable these features - for example, by mounting the gateways closer together.

Coverage Area Examples in FCC

Table 3 shows the coverage area of the gateway using highly sensitive Monza R6-based 70mm tag inlays in an emulated free space environment. All the equipment used was FCC based including the xArray, xSpan, and tags. Free space environment means that there were no obstructions between the gateway's antenna beam and the tag causing beam deformation or significant multipath. The tagged items in the testing lab environment consist of tags resting on foam, and ideal dielectric. Both the xArray and xSpan were configured to maximize tag read performance, namely:

- ReaderMode="Mode1002:AutosetDenseReaderDeepScan"
- SearchMode= SingleTarget
- Session=2
- TransmitPower=30dBm

TIP: Consider this coverage area data as best case when setting up tests in your deployment environment. You will experience less coverage area when using smaller inlays, less sensitive tagged items, or have significant obstructions in yoru physical environment (metal shelving, ductwork, etc…).

The coverage area has a 90% or greater chance or reading the tags under these RF friendly conditions. The perimeter of the coverage area is at 90% unique tags read and the interior (closer to the gateway) goes up to reading 100% of the tags.

Table 3's dimensions for the xArray's circular coverage area are based on >=90% unique tags reads using 70mm Monza R6 tags in free space. Figure 6 represents the radial coverage area relative to height.

Table 3: xArray Coverage Area Test

XARRAY HEIGHT ABOVE TAGS	RADIUS (FT)	DIAMETER (FT)	AREA (FT^2)	HEIGHT (M)	RADIUS (M)	DIAMETER (M)	AREA (M^2)
1	12.5	25.0	491	0.3	3.8	7.6	46
2	13.8	27.7	601	0.6	4.2	8.4	56
3	15.2	30.3	723	0.9	4.6	9.2	67
4	16.5	33.0	855	1.2	5.0	10.1	79
5	17	34.0	908	1.5	5.2	10.4	84
6	17.5	35.0	962	1.8	5.3	10.7	89
7	18	36.0	1018	2.1	5.5	11.0	95
8	18.6	37.3	1090	2.4	5.7	11.4	101
9	19.3	38.5	1164	2.7	5.9	11.7	108
10	19.9	39.8	1241	3	6.1	12.1	115
11	20.5	41.0	1320	3.4	6.2	12.5	123

Figure 6: xArray Coverage Diagram

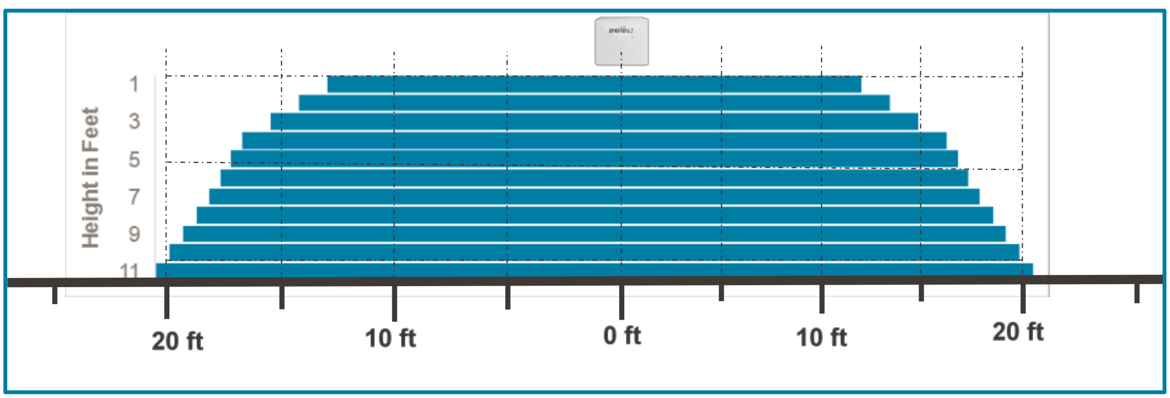


Table 4 shows the dimensions of xSpan's oblong coverage based on >=90% unique tags read using 70mm Monza R6 tags in free space. Figure 7 and Figure 8 represent the coverage area of the xSpan over height along the major and minor axes, respectively.

Table 4: xSpan Major Axis Coverage Diagram

XSPAN HEIGHT ABOVE TAGS	MAJOR AXIS (FT)	MINOR AXIS (FT)	AREA FT²	HEIGHT (M)	LENGTH (M)	WIDTH (M)	AREA M²
1	28	20	560	8.5	8.5	6.1	52
2	29.0	20.8	603	8.8	8.8	6.3	56
3	30.1	21.5	647	9.2	9.2	6.6	60
4	31.1	22.3	694	1.2	9.5	6.8	64
5	31.7	22.6	717	1.5	9.7	6.9	67
6	32.3	23.0	742	1.8	9.8	7.0	69
7	32.9	23.3	767	2.1	10.0	7.1	71
8	34.3	24.4	835	2.4	10.4	7.4	78
9	35.7	25.5	907	2.7	10.9	7.8	84
10	37.0	26.5	982	3	11.3	8.1	91
11	38.4	27.6	1060	3.4	11.7	8.4	98

Figure 7: xSpan Major Axis Coverage Diagram

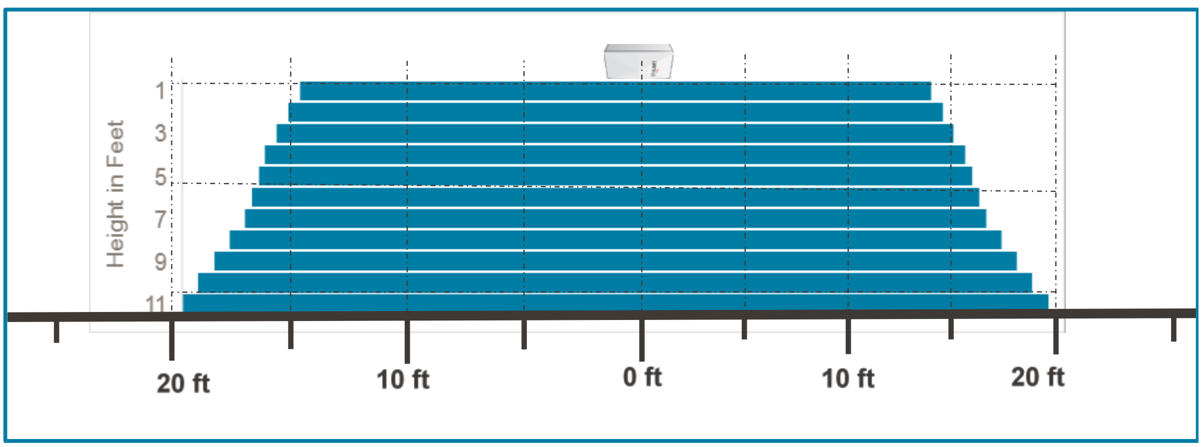
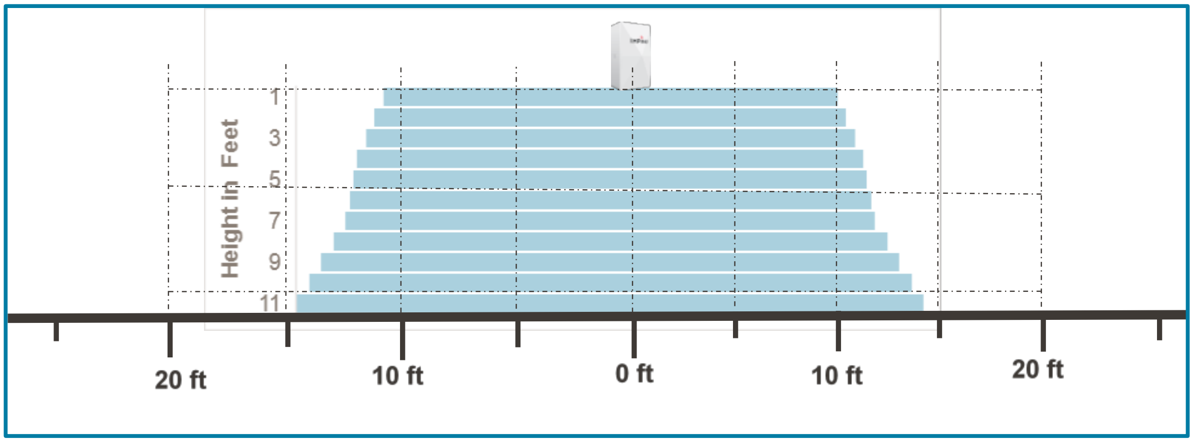


Figure 8: xSpan Minor Axis Coverage Diagram



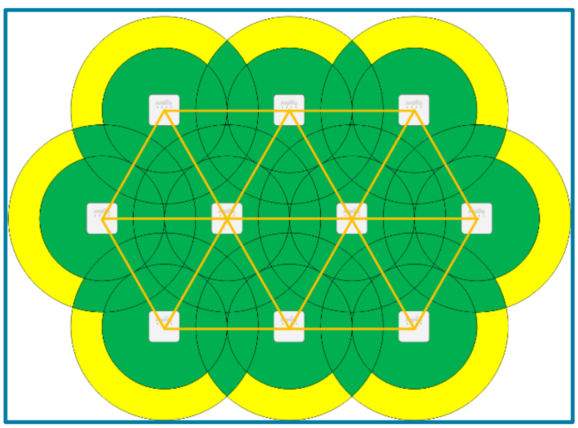
Testing coverage in your environment

Follow the instructions in the Deployment Testing section to verify that your gateways are positioned to meet your requirements. If over deployed, consider spacing the gateways farther apart. Likewise, if the read percentages do not meet requirements then spacing the gateways closer is necessary.

xArray Hexagonal Packing

Gateways are typically installed on a rectangular grid. xArrays with their circular coverage can be installed in a hexagonal pattern. This pattern has shown to improve location accuracy since it increases the likelihood of tag reads from multiple xArrays. The idea is to arrange the xArrays so the circular coverage areas are packed as efficiently as possible. See Figure 9.

Figure 9: xArray hexagonal packing example



Notes about Construction

- Antenna beams use phase manipulation, which means most beams form at approximately 1/3 meter from the surface of the gateway.

- Antenna beams have side lobes that result when synthesizing the more obliquely angled beams. Beams can be steered up to 45 degrees. Beams steered at 45 degees from the boresight have the greatest amount of side lobe. Beams not steered or only slightly steered will not have side lobes. Both the Tag Direction and Location Mode consider this phenomenon in their respective algorithms.
- Beam geometries are not precise (they're propagating radio waves!) so the user must not expect the beams to directly conform to the geometries depicted on the xArray or xSpan antenna charts.
- Gateways are designed for installation at heights ranging from 3m to 4.5m (10ft to 15ft). Don'tassume that automatically installing a gateway at 4.5m gives the best performance. Remember that the gateway emits propagating waves, which are subject to path loss. Go too high and the Gateway's wave may be too attenuated to energize tags. Testing is always the best way to determine the correct height.
- Numerous lab and real world tests have shown an installation height of about 3.5m (11.5ft) to be ideal. It provides the best coverage with the highest tag readability rates.
- When deciding on the placement of overhead gateways, don't base your decisions on the furniture, racks, cabinets, or other large objects below especially if their positions are subject to change.

Gateway Placement Best Practices

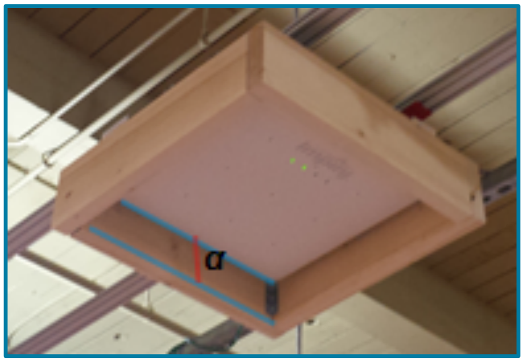
The gateway needs to be installed to maximize coverage within the constraints imposed by the physical characteristics of the deployment environment. Please keep the following in mind when designing your deployment.

- Install the gateway at or below the level of HVAC equipment, pipes, and especially lighting, such as in Figure 10. Not only will such equipment block the gateway beams, but florescent lighting will also create RF noise interference if in the field of view. This rule applies only when the HVAC or lighting is below the gateway, in the area where tags are to be read. For example, if the gateway were 3 meters away from the other equipment, this rule would not apply since the equipment is outside of the field of view.

Figure 10: xArray Installed Slightly Lower Than Vents and Lighting

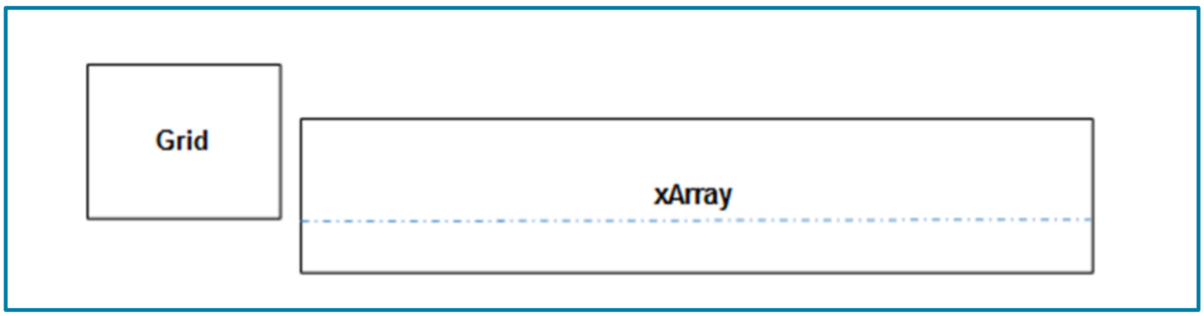
- Installing the gateway (xArray or xSpan) into a coffered ceiling (Figure 11) can reduce tag read performance. The deeper the gateway is mounted into the coffered ceiling, the more impact there is on tag reads. The beams on the outer ring are the most affected.

Figure 11: xArray installed deeply into coffered ceiling panels



- The best way to mount in a grid system is to line up the ground plane of the xArray with the Grid System. The ground plane is 1" from the face of the radome (2 inches from the backplane). Figure 12 features a side view for demonstration.

Figure 12: xArray Ground Plane Aligned with the Grid System



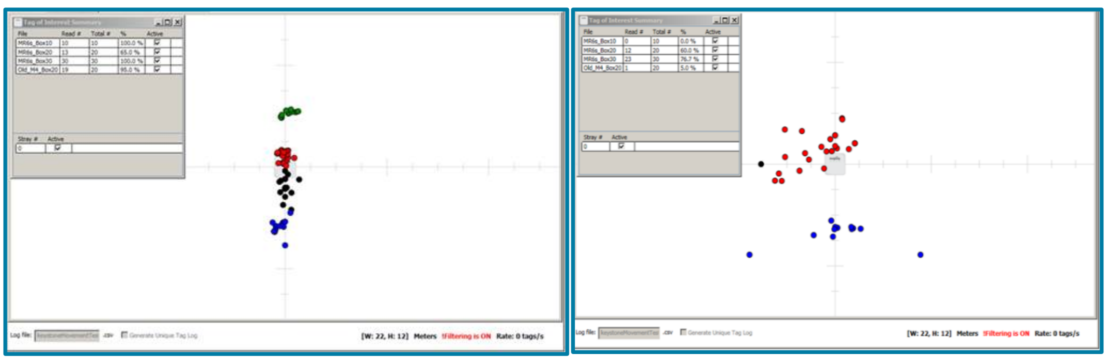
xArray Specific

- Installing the xArray near a wall or in the corner of a room is highly undesirable. The material in walls will create reflection which is especially troublesome for an xArray in Location Mode (because it reduces X and Y coordinate accuracy). Moreover, the corner of a room is difficult to cover efficiently with circular coverage areas. However, although generally inadvisable, there may be certain situations where corner placement is unavoidable. If your scenario seems to call for such placement, you can mitigate the effects of multipath to some extent by selectively turning off antennas that face a nearby wall (or even turning off all antennas except a few that are facing a target tag population).
- For xArray, installing directly above shelving will result in limited tags reads especially on the lower shelves. In Figure 13 we show a set of shelves with tagged items on the bottom shelf, with the reader adjacent to the wall. Figure 14 shows two diagrams: one showing the results of a Location Mode where the items unobstructed (not on a bottom shelf), and the second showing the results with the same items on the shelf as shown. Notice when the shirts are located without any obstruction, they are all read and located with moderate accuracy. However, when the items are on the bottom shelf, they are not all read and their locations are scattered.

Figure 13: An Undesirable Placement

Placement next to wall and directly above shelving makes the bottom shelf items difficult to read for the xArray.

Figure 14: Location Mode for Bottom Shelf Objects

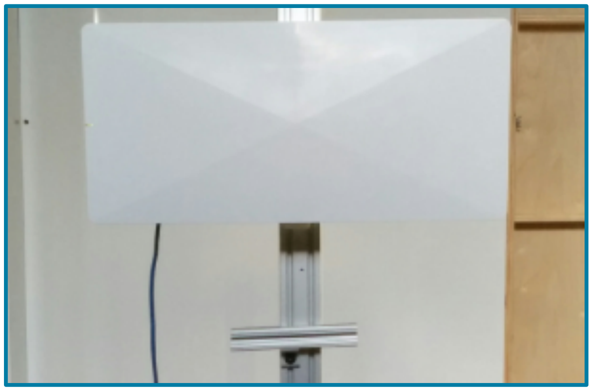


Location Mode for items on bottom shelf when unobstructed (left), vs. when the read is obstructed by shelving (right).

xSpan Specific

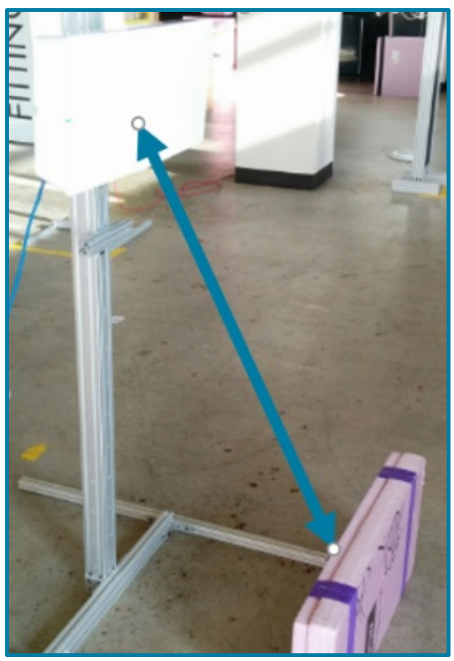
While gateways typically mount overhead keeping them out of the way and out of sight, the xSpan is also designed to mount sideways. To use the xSpan in Direction Mode, horizontally mount it sideways as shown in Figure 15 by aligning the major axis parallel to the ground. This allows the tagged item to transition between sectors as they pass in front of the xSpan.

Figure 15: xSpan Horizontally Mounted on a Pole



As you would for any RAIN RFID application, test the read zone of the xSpan. Verify that the read zone meets your needs. For example, place your tagged items as close as possible to the xSpan while still on the ground so that all the tags still can be read as show in Figure 16.

Figure 16: Horizonally Mounted xSpan Read Zone Test



Tags can still be read at angle/distance shown.

- Vertically mounting the xSpan on a wall will not provide direction of tagged items passing by the xSpan. It will however provide presents of a tagged item in the xSpan's field of view. A vertically mounted xPortal can also be used in this way.
- An xSpan in inventory mode can be mounted overhead or horizontally side-mounted in a small room. In a small room install the xSpan in the center of room when overhead or in middle of the wall. Pick a place that minimizes the number objections between the tagged items and the antenna.

Poor Mounting Conditions

Mounting a gateway under any of the following conditions should be avoided:

- Proximity to fluorescent lights: Ideally place gateways greater than a foot away. The reason for this is that the ballasts can oscillate in the data bands causing RF interference.
- A scenario where there is a metal impediment between the tagged item and the gateway. Given that metal blocks the RFID signals, a tagged item's readability will be greatly reduced if it is blocked by metal.
- Placement directly under or next to a fire suppression nozzle (sprinkler): gateways are not water-resistant.

Placement Parameter Setup for Location Mode

When performing an xArray Location Mode based deployment, the following preparatory steps are necessary:

1. Perform a site visit to find the ideal location to install the xArrays.
2. Plot the xArrays on a floor plan drawing.
3. Define the southwest corner of your facility as (0,0). This will allow you to use positive values for the coordinates of your xArrays throughout the facility.
4. Gather the facility X and Y coordinates of each xArray.
5. Gather the xArray orientations (for simplicity try to install each xArray with the same orientation).
6. Determine the average tag height for each gateway.

Painting the Radome

To paint an xArray or xSpan radome, follow a few simple steps.

1. Remove radome from system.
2. Clean radome with alcohol to remove oils and small particles.
3. For xArray, mask off LED light covers to avoid painting.
4. Paint exterior body of radome with spray paint or paint spray gun. Do not use metallic based paints as they will have RF implications, but water, oil, and latex paints will work.

GATEWAY INSTALLATION

Installing a gateway requires you to mount it in a location with electrical power, and to provide network connectivity to the unit.

Mounting a gateway

Gateways mounted on the ceiling can provide continuous, dependable coverage. Impinj gateways are engineered for maximum versatility in installation.

- We identify mounting options that work for various deployment scenarios on the Third Party xArray Mounting Options[H] page and Third Party xSpan Mounting Options[I].
- See The xArray and xSpan Installation and Operations Guide. Download it from the xArray Partner and Documents folder[J] on the support portal.
- On the support portal you can also watch a video showing installation methods[K].

Network Connectivity

Network connectivity is covered in the xArray and xSpan Installation and Operations Guide. Download it from the xArray Partner and Documents folder[L].

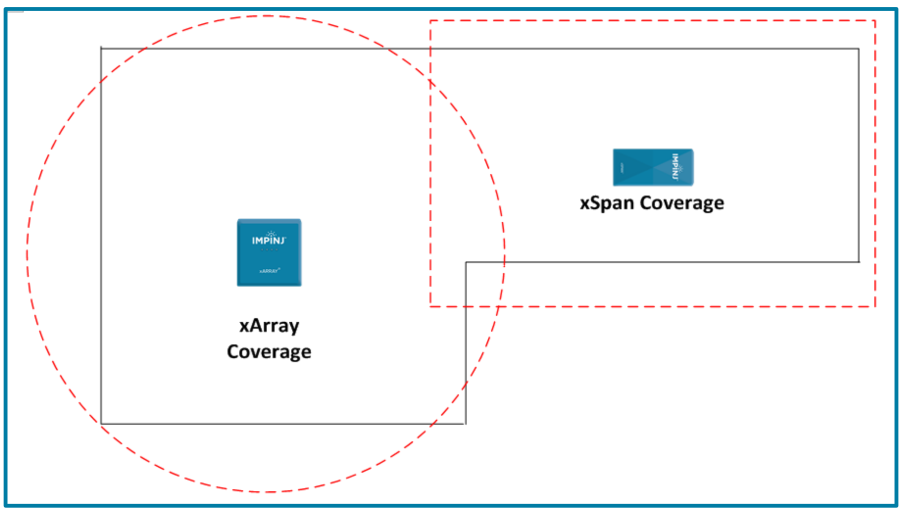
INVENTORY MODE: CONFIGURATION AND TESTING

Concepts to Know

In Inventory Mode, you can think of the xArray as a Speedway with 52 virtual antennas that can be switched through very quickly. Similarly, the xSpan quickly switches through 13 virtual antennas. (See Appendix 1: Gateway Sector and Beam Diagrams.) Gateways in Inventory mode have full control of the antennas. For example, they can set Gen2 parameters such as Session, Search Mode, transmit power, antenna sequence, tag population, and enable and disable beams. (See Table 5 for a list of parameters in Inventory mode.) While Inventory Mode gives users total control over the gateway, it doesn't provide Location or Tag Direction reports.

When running Inventory Mode, xSpan's rectangular coverage and xArray's circular coverage can be combined to fit the facility's needs. An example is shown in Figure 17.

Figure 17: Fitting the Facility with 1 xArray and 1 xSpan



Constraining the Read Zone

In some scenarios, only those tags inside a given area are relevant. If the full deployment also includes tags outside this area, you can constrain the read zone for a gateway to avoid reading tags irrelevant to that scenario. This method can still result in "stray" reads, where unintended reads of nearby tags not directly in the intended field of view exist. The RAIN RFID middleware that receives data from the gateway removes these "stray" reads.

- For the xSpan and xArray, you can control the read zone to some extent by selecting appropriate antenna beams. ItemTest and the Octane SDK can be used to enable and disable antennas.
- Adjusting transmit power is yet another way to limit the read zone. You can adjust the transmit power in ¼ dB steps (ranging from 10 dBm to 30 dBm), in any mode.

Table 5: Configuration Parameters for Inventory Mode

GEN 2 PARAMETERS	VALUES
Antennas	xArray: Optimized (antennas: 45,16,35,6,41,28,47….) xSpan: Optimized (antennas: 1,2,3,4,5,6,7,8,9,10,11,12,13)
Mode	1002: Autopilot Static
Estimated Population	4
Session	2 or 3
Power	30 dBm

The gateway will read the maximum number of unique tags in Inventory Mode when the user has selected SingleTarget and Session 2 or Session 3. Session 2 and 3 with their long persistence will force the closer easier to read tags to stay quiet while the gateway finds the harder to reach tags. In Inventory, the user has full access to all RAIN RFID settings in the gateway. Reader Mode 1002: Autopilot Static provides the most powerful over the air communication with the tag optimized for reading the largest number of unique tags. Using a low estimated population (ex. 4 tags) even in a large tag population will also yield higher unique tag counts.

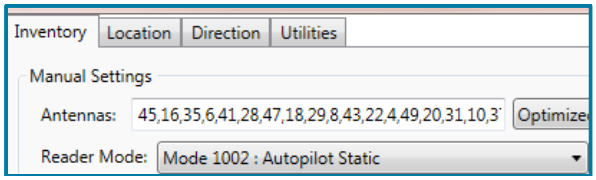
Antennas

This parameter lets you enable and disable antenna beams by sector, concentric ring, or individually.

- The antenna sequence (i.e. the order in which antennas are fired) affects the overall number of tags read. Different antenna sequences may be more or less efficient in suppressing already inventoried tags.
- Alternation allows the gateway to read more unique tags faster because adjacent/sequential beams have some degree of intentional overlap. When using all 52 antennas on xArray we recommend the following firing order, which is the same as "Optimized" in ItemTest. See Figure 18 for the recommended firing order in manual settings, and Figure 19 for the coding in Octane SDK:
  - (Start) 45, 16, 35, 6, 41, 28, 47, 18, 29, 8, 43, 22, 4, 49, 20, 31, 10, 37, 24, 51, 14, 33, 12, 39, 26, 2, 13, 32, 11, 38, 25, 52, 15, 34, 5, 40, 27, 46, 1, 17, 36, 7, 42, 21, 48, 19, 30, 9, 44, 23, 50, 3 (End)
- On xArray sometimes using only 28 antennas reads more unique tags than 52 because of the shorter amount of time required to cycle through all of the antennas. The 28 are:
  - (Start) 45, 16, 35, 14, 49, 36, 2, 31, 18, 29, 48, 19, 46, 3, 33, 52, 15, 50, 13, 32, 1, 51, 30, 17, 47, 20, 34, 4 (End)
- On xSpan the 1,2, 3,…,13 antennas alternate between opposite sectors so sequencing from 1 to 13 will create an efficient antenna sequence.
- The xSpan has ¼ the number of beam as xArray so cycling through the beam sequence is much faster. You will notice the fast inventories from xSpan in a mixed environment of xArrays and xSpans.
- If you need to limit the coverage area, reduce the beam output power. See *Power* parameter.
- Timing information:
  - The maximum time spent on each polarized beam is 250 milliseconds in single target inventory.

- The maximum number of beam polarizations on xArray if all 52 beams are enabled is (2 polarizations * 52 beams) = 104.
- The time depends on the number of tags found on each beam. If there are no tags the gateway will spend only a few milliseconds on each polarized beam.
- See the ConfigureManyAntennas Example in Octane SDK

Figure 18: ItemTest Notes – Inventory Mode Antennas



Click Optimized to generate a field-tested, high-performance alternating antenna sequence. The specific firing order is shown in the antennas: edit box. ItemTest dialog shows Optimized antenna sequence

Figure 19: Octane SDK C# – Inventory Mode Antennas

```
// Use the AntennaUtilities class to help select antennas to enable and disable. Methods
include:

// AntennaUtilities.GetAntennaIdsByRing (xArray Only)
// AntennaUtilities.GetAntennaIdsBySector (xArray/xSpan)
// AntennaUtilities.GetOptimizedAntennaList (xArray/xSpan)

// ********** Optimized Antenna List code  ******************

// Get Optimized list of antennas on an xArray
List<AntennaConfig> listAntennaConfig = settings.Antennas.AntennaConfigs;
listAntennaConfig.Clear();
foreach (ushort antenna in AntennaUtilities.GetOptimizedAntennaList(ReaderModel.XArray))
{
        listAntennaConfig.Add(new AntennaConfig(antenna));
}
```

Default: All Antennas are enabled

Reader Mode

- `ReaderMode.AutoSetDenseReaderDeepScan` (Impinj Numeric Mode ID: 1002) is recommended when inventorying slowly moving tags (<= 0.5 m/s) or not moving at all (e.g. apparel items in a retail store). `ReaderMode.AutoSetStaticFast` (Mode ID = 1003) will provide faster read rates but it is more susceptible to interference from other RF sources, possibly from nearby gateways. (See Figure 20 for settings.)

Figure 20: Reader Mode – Inventory Mode Settings

```
// Reader Mode Dense Reader/Deep Scan – Mode 1002: Autopilot Static
s.ReaderMode = ReaderMode.AutoSetDenseReaderDeepScan;
```

Default: AutoSetDenseReader

Estimated Population

- Keep this number low particularly when running an inventory in Session 2 or 3. Empirical testing shows a value of **1** gives the best results regardless of the number of tags in the field of view. With TagPopulationEstimate set to 1, the gateway beams continuously cycle over the tag population incrementally reading small numbers of tags each time. It is this "searchlight" action that allows the gateway to read both quickly and broadly. (See Figure 21 for settings.)

Figure 21: Estimated Population – Inventory Mode Settings

```
// Set Estimated Tag Population to 1
s.TagPopulationEstimate=1;
```

Default: 32

Session

- Using Session 2 or Session 3 in Single Target permits reading large tag populations quickly. Tag populations exceeding 200 to 300 tags are usually good candidates for Sessions 2 or 3 using Single Target. (See Figure 22 for settings.)
- If the long decay time associated with Session 2 or Session 3 is a potential issue, you might try using Session 1 with or without TagFocus[M]. Impinj Monza-based tags are required for TagFocus. However, both Session 1 and Session 1 with TagFocus may have problems due to the spatial diversity of the beams. In other words, it takes time for the gateway to go back and boost previously inventoried tags as the gateway cycles through the beams. Hence, the efficacy of Session 1 and Session 1 with Tag Focus will depend on the number of tags and the reader mode. The reason for this is that suppressed tags must "hear" the suppression command from the gateway at regular intervals. If the gateway is inventorying large tag populations in an opposite direction from the suppressed tags, those suppressed tags may fail to hear the TagFocus command. They will then revert back to normal Gen2 Session 1 behavior.
- Run Session 1 on handhelds to prevent sharing sessions used by gateways running session 2 or 3. See APPENDIX 2: Operating Handhelds Readers with a Gateway.

Figure 22: Session – Inventory Mode Settings

```
// Configure Each gateway to Session 2 or Session 3.
s.Session = 2;
s.Session = 3;
```

Default: 2

Target

- Use Single Target inventory to read the largest number of unique tags in the shortest amount of time. Use Dual Target to get numerous reads for each tag, resulting in Single Target reading the largest number of unique tags. (See Figure 23 for settings.)

Figure 23: Target – Inventory Mode Settings

```
// Single Target Inventory
s.SearchMode = SearchMode.SingleTarget;
```

Default: Dual Target

Power

Sets the signal strength of the gateway.

- Range: [10dBm – 30 dBm]
- Set output power to *30 dBm* to maximize the coverage area. (See Figure 24 for settings.)

Figure 24: Power – Inventory Mode Settings

```
// 30 dBm for max power.
s.Antennas.GetAntenna(1).TxPowerInDbm = 30;
```

Default: MaxTxPower set to true

Gen2 Filter

You can use Gen 2 Filtering to read only tags that match the filter(s) entered. (See Figure 25 for an example.)

Figure 25: Antennas – Inventory Mode Settings

```
// Define 1 Filter (EPC, TID, Reserved, and User); up to 2 filters
s.Filters.TagFilter1.TagMask = "9999";
s.Filters.TagFilter1.MemoryBank = MemoryBank.Epc;
s.Filters.TagFilter1.BitPointer = BitPointers.Epc;
s.Filters.TagFilter1.BitCount = 16;
s.Filters.Mode = TagFilterMode.OnlyFilter1;
```

Default: No Filter

Data: Tag Reports

Every time an antenna reads a tag a report is generated.

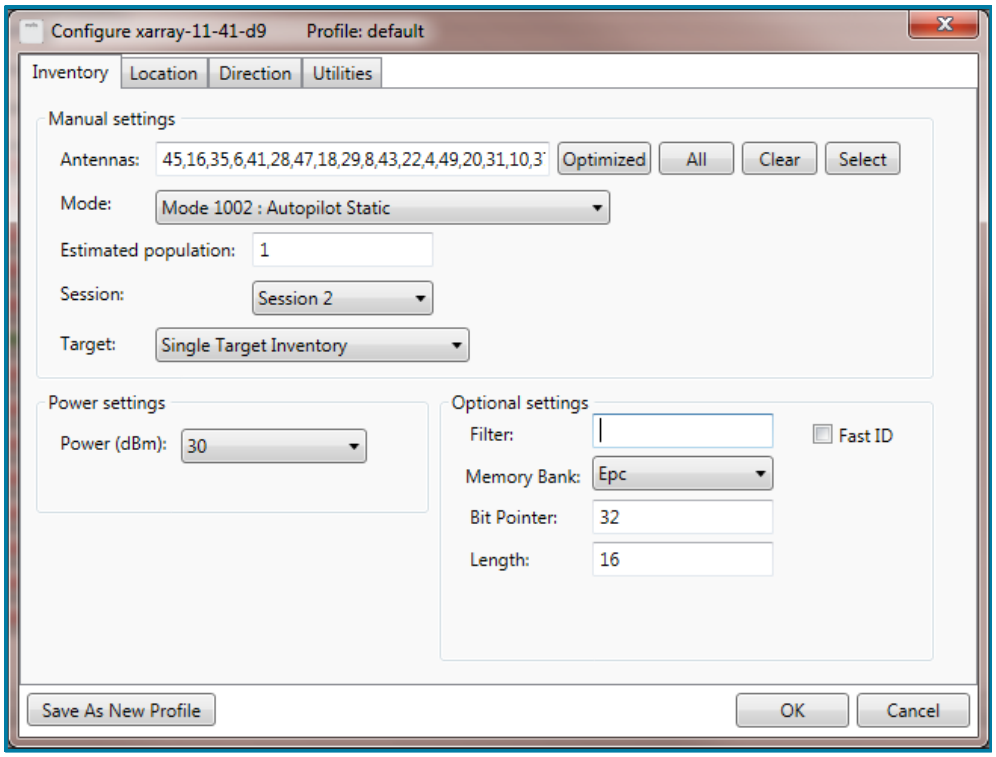
Table 6: Data Tag Reports – Inventory Mode Settings

ITEMTEST NOTES	OCTANE SDK C#
Each received TagReport will update the ItemTest UI tag list.	OnTagsReported handler creates a Tag instance for each tag. Includes EPC, TimeStamp, Channel, AntennaPortNumber, Rssi, … See the SDK for more details.

Deployment Testing

These instructions show how to configure ItemTest to run Inventory modes and validate the coverage of your deployment by measuring the percentage of tags read by the gateway that were defined in your TOI files.

Figure 26: ItemTest Dialog Settings

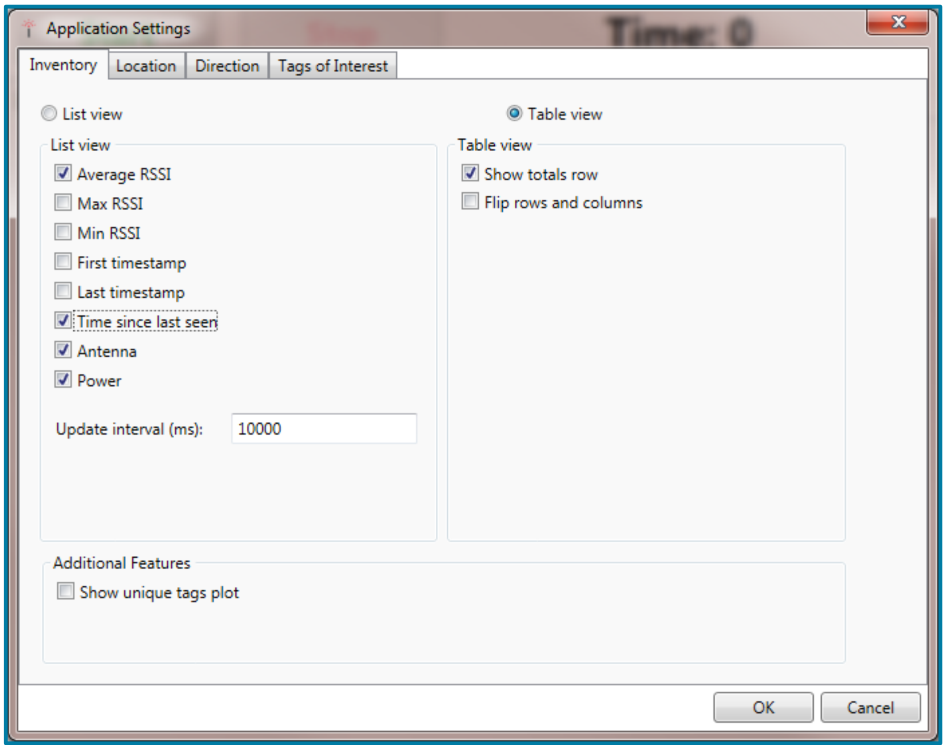


1. On Inventory tab of the Configure screen (Figure 26Figure 27), next to Antennas, click Optimized to use Impinj's recommended beam sequence.
2. Set the following reader parameters:
   - Estimated population = **1**
   - Mode = **AutoPilot Static (1002)**
   - Power(dBm) = **30**

- Session = **2** or **3**
- Target = **Single Target Inventory**

3. Click OK to save the configuration change.
4. On the main ItemTest tab click Application Settings. (Figure 27)

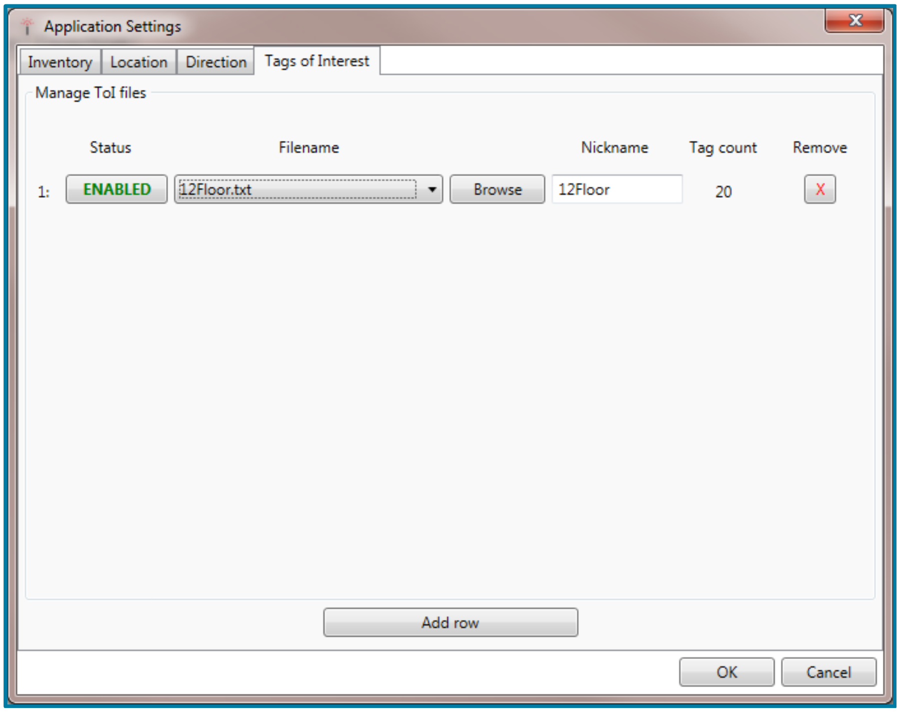
Figure 27: Inventory Application Settings in ItemTest



Enter Caption Text Here

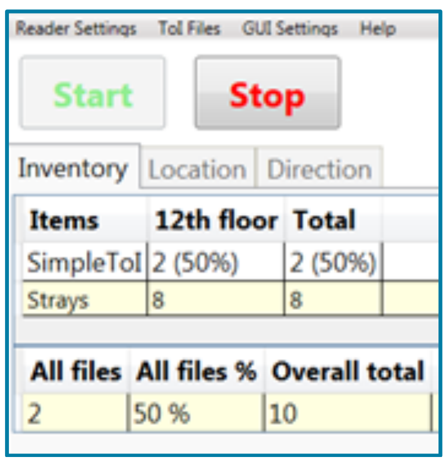
5. Select and enable the TOI file in the Application Settings→ Tags of Interest dialog. (Figure 28)

Figure 28: Application Settings dialog



6. On the Inventory tab, click the **Table View** radio button, and check **show totals row**. (Figure 28Figure 29)

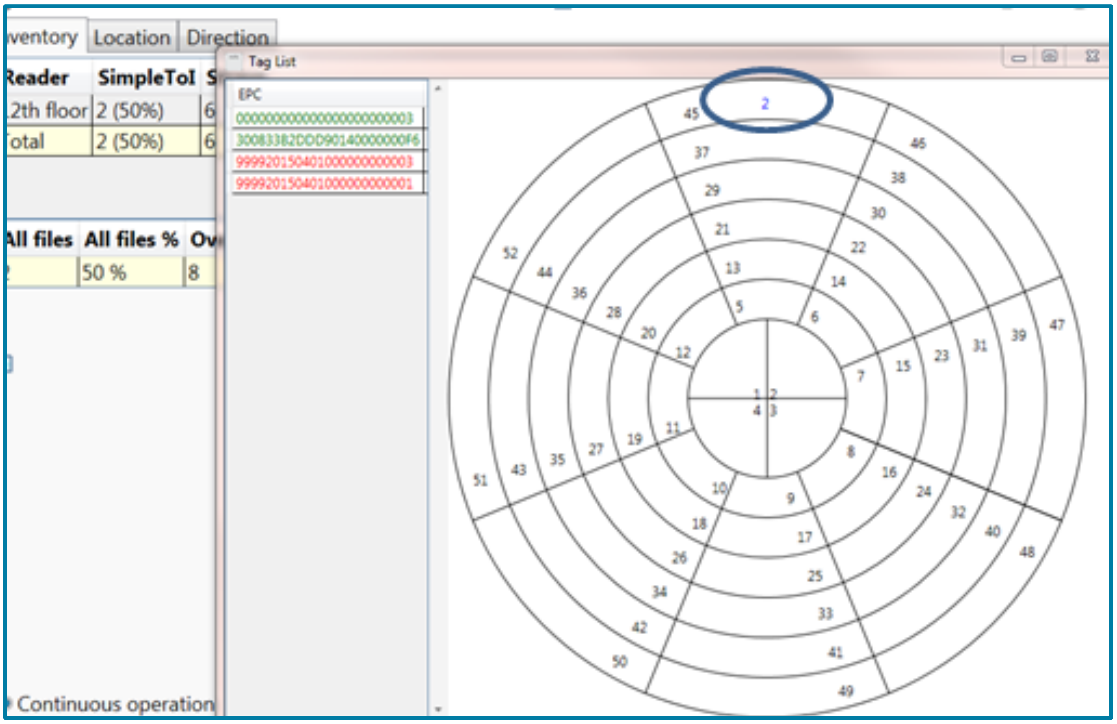
Figure 29: Running Inventory with table view selected and showing the TOI read percentages



7. Run your inventory for at least 5 minutes
8. To simulate the effect of environmental RF changes, move around the space and handle tagged items. Doing so will likely increase read percentages and provide a good prediction of how the system will perform when actually deployed.
9. After the inventory operation (Figure 29) you can look at which antenna beams have read tags of interest. Double-click the column in Table View that displays total tags read to see a visualization

of unique reads per antenna beam. For example, Figure 30 shows which antennas read tags with EPCs labeled “SimpleToI” in the TOI file. Here Antenna 45 read 2 tags.

Figure 30: Tag List Location Diagram



TIP: If you find that most reads are coming from the outer rings, then you will likely experience less accuracy in Location Mode. You should consider adding additional gateways and employing the weighted average algorithm.

To measure the readability at different locations within your deployment perform the following:

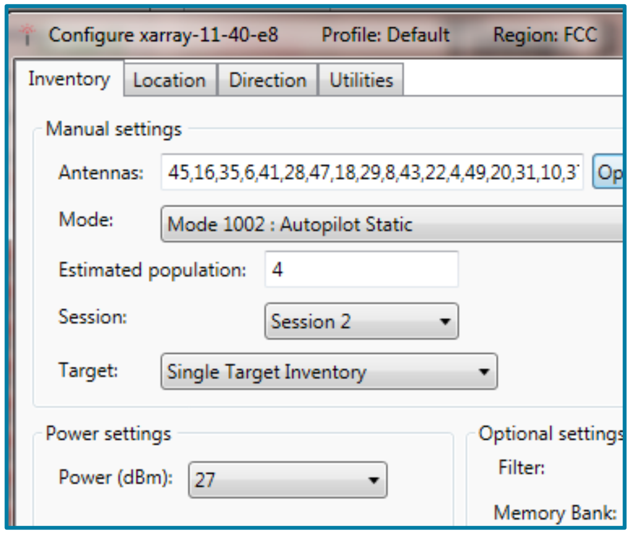
- Identify a small set of tagged items. For example, a stack of tagged jeans.
- Create a TOI file of these Items.
- Place these items in different locations of your facility and run inventories for several minutes to measure the read percentages in each location.
- If you find that some locations have weak coverage, consider moving or adding gateways to those areas.

Inventory Mode - Overdeployment Test

If you are setting the Maximum TxPower and enabling all of your antennas and yet you are consistently reading less than 100% of your tags, you are not over deployed. See Tags of Interest in ItemTest. This assumes that.

On the other hand, if you are consistently reading 100% or very close to 100% of your tags, you may have over-deployed. You can systematically reduce the power supplied to the antennas to determine the degree for which you are over deployed, as shown in Figure 31.

Figure 31: Using ItemTest Dialog to adjust the Transmit Power



The power setting has been reduced from max power at 30dBm to 27 dBm. Keep observing the tag reads as power is reduced to determine the amount of overdeployment.

Show Unique Tags Plot

In ItemTest, use the “Show Unique tags plot” feature to graph the progress of the number of unique tags read to gain a better understanding of how long it takes to inventory your tags. Figure 32 shows the “Show unique tags plot” selected, and Figure 33 shows a Unique Tags plot graph.

Figure 32: "Show unique tags plot" in Application Settings

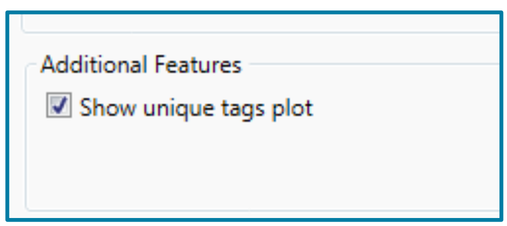
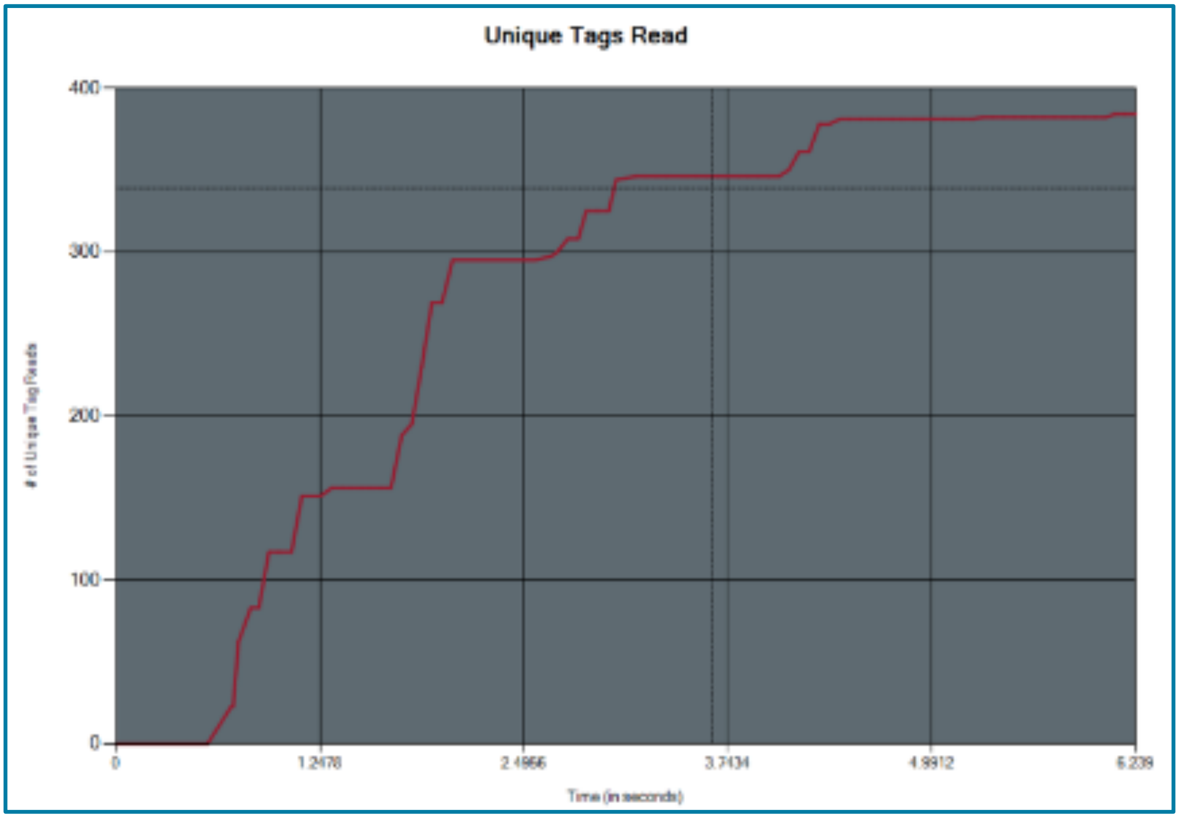


Figure 33: "Unique Tags Read" graph displayed during Inventory

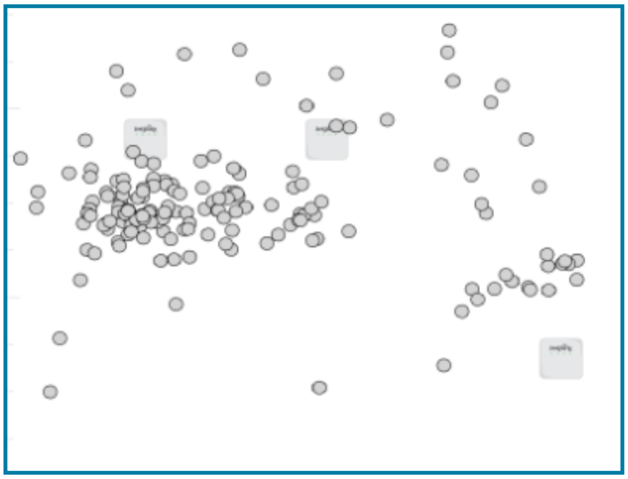


In many retail apparel use cases, for instance, users run xArrays in the Location Mode at night. While a prolonged time to read tags is helpful, it is important to recognize that runs done without items moving, i.e. without the benefit of customers interacting with merchandise, mean unique tag counts likely may be sub-optimal. The reason: motion gives xArrays new opportunities, regardless of how fleeting they may be, to read tags otherwise obstructed, potentially "de-tuned" by proximity to other objects or where the RF energy is nulled.

Inventory Over-Deployment Test Steps in ItemTest:

1. In Reader Configuration dialog (see above), reduce the Power (dBm) by 3 (from 30 to 27).
2. Re-run the Inventory Mode (allow sufficient time to read the tags – at least two minutes).
3. If still seeing 100% of your tags, reduce the Power (dBm) by another 3dBm and re-run the Inventory Mode; repeat until the power is low enough that you are not able read at 100%. Remember that since Single Target and Session 2 are used, you will need to allow 2 minutes between Inventory runs for the tags to have time to de-energize.

Figure 34: Tags plotted per their location



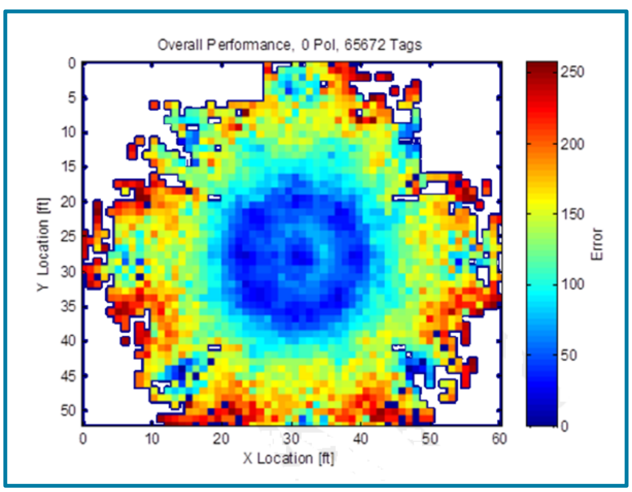
xArray depicted as the white square.

When the xArray is running in the Location Mode, it computes the X and Y coordinates of tags, as shown in Figure 34. Location Mode determines X and Y based on the internal computations and the provided average height of xArray with respect to the tag. The Location Mode is unique to the xArray; there is no comparable mode of operation with traditional RFID readers. When a tag is first read, the xArray gateway will send a TagEntry report. Based on the user-settable UpdateInterval value in Octane SDK, updated location reports will be sent every *n* seconds. Finally, after the tag is no longer being read for a period set by the TagAgeInterval, the xArray will no longer generate update reports and send a TagExit report.

Concepts to Know

NOTE: If the RFID Gen2 concepts presented here are completely new to you and you need more help understanding them, we recommend you participate in our RFID training[N].

Figure 35: Location Accuracy Heatmap



- Location accuracy is best directly under the xArray. The heatmap in Figure 35 shows the highest accuracy in a 12ft radius around the xArray in blue. The farther from the xArray the less accuracy.
- When using the Location Mode, the xArray always inventories tags via Dual Target. Therefore, if a tag is taken from the field of view of the xArray, you cannot know whether that tag's inventory flag is in the A or B state for the relevant session. The implication of this is that if you expect to read that tag via another reader, you should ensure that other reader is inventorying the tag using a session different from its closest xArray neighbor. Alternatively, choose Dual Target on the other reader(s).
- The time to cycle through all antennas (referred to as the cycle time) depends on the number of tags in the field of view, the reader mode, and the number of antennas enabled. If the tag population is sufficiently high, it may take tens of seconds, possibly minutes, to complete a full cycle. Reducing the cycle time is generally desirable in Location Mode when the location of moving tagged items. In other scenarios, reliably reading as many unique tags as possible might be your priority. Table 7 illustrates the tradeoffs between the number of unique tags read and cycle time when configuring different reader modes.

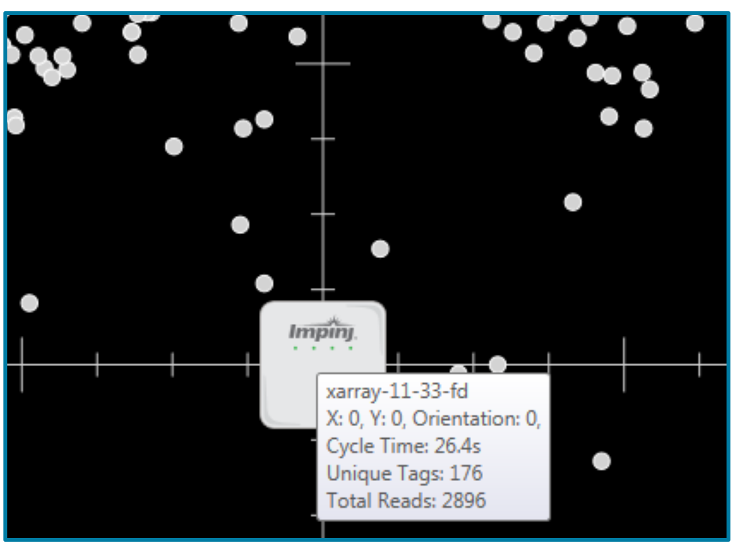
Table 7: Reader Mode, Unique Tags Read vs. Cycle Time

READER MODE	UNIQUE TAGS READ	CYCLE TIME (SEC)	
Max Throughput (0)	198	1.8	
Autopilot Static (1002)	304	22	
Autopilot Static Fast (1003)	274	5	
Autopilot Static Dense Reader (1004)	293	33	

Generally, as Reader Mode cycle time decreases, number of unique tags that are read in a cycle also decreases.

- **ItemTest Note**: To find the cycle time, hover over the xArray icon. An example of hovering is shown in Figure 36 below. A message box displays the cycle time (in this example, 26.4 seconds), the total reads (2896 in the example) and the number of unique tags (176).

Figure 36: Run Statistics from xArray Icon



To display run statistics, hover mouse pointer over xArray icon.

A battery assisted tag (BAP) may give poor results when the xArray is in Location Mode. This is because the tag does not rely on the xArray for its power source and is likely visible to orthogonal and/or opposing beams. Therefore, we do not recommend using BAP tags when in Location Mode.

Location Max Number of Tags

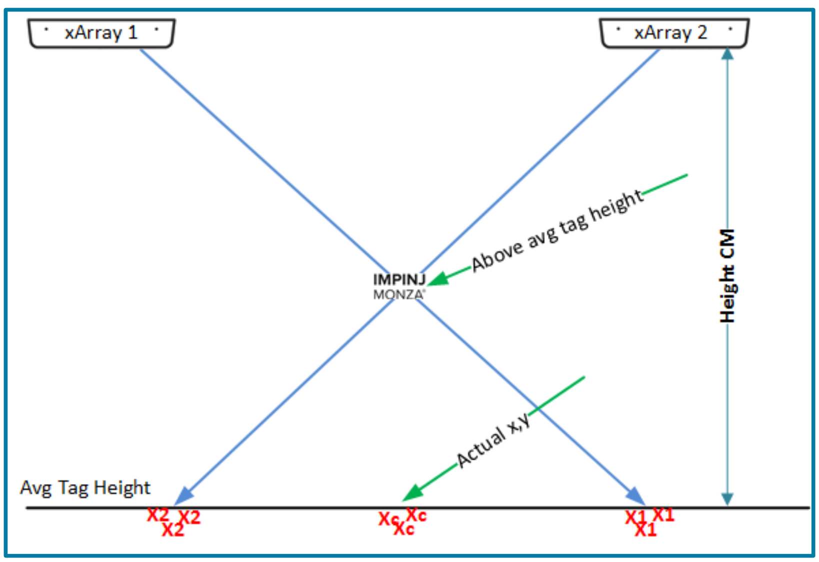
The location engine in the xArray can be computationally intense and does have a limitation as to number of tags that it will track. For example, if Reader mode 1002 AutoPilot Static is used the maximum number of tags for an xArray is 4096 tags. Please see the Readers and Gateways Octane <Version number> Release Notes for specifics on the maximum number of tags.

Weighted Averaging with Multiple xArrays

Besides providing greater RF coverage, using multiple xArrays is an excellent way to increase location accuracy. By inputting tag reports into a simple weighted averaging calculation, the user will likely see more accurate computed locations.

In Figure 37, even though the tag being read is tens of centimeters above the average tag height entered into the xArray's settings, a weighted average with inputs from two xArrays helps to minimize location error. The xArray is continuously computing the x and y location of a tag. In Figure 37, the 3 X1's represent the result of different x, y locations calculated by xArray #1. Similarly, the X2's are location calculations from Array #2. The Xc's are calcuations from averaging the locations of X1 and X2.

Figure 37: Weighted Tag Average Height Measurement, Two xArrays, Centered



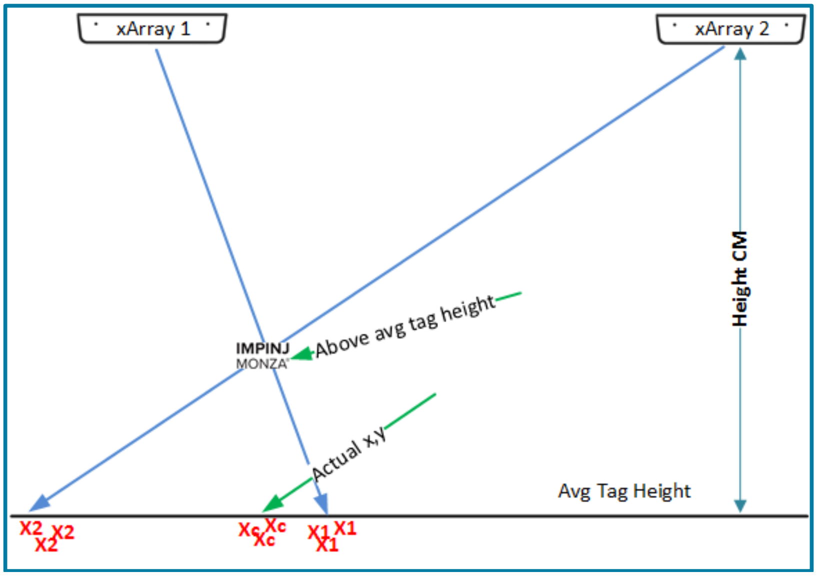
Note that while using a weighted average computation generally improves location accuracy for the majority of tags, there are some instances where it may modestly increase error. Specifically, if a tag is directly under one xArray (xArray #1), and that tag is also being read by a second, more distantly located xArray (xArray #2), reads from xArray #2 will "drag" the tag closer to xArray #2 when factored in the weighted average algorithm. Moreover, RF interference (from absorption and reflection by nearby objects) will increase read errors.

For example, imagine a retail situation where a stack of sweaters is displayed on a table with an xArray directly overhead. If a shopper is bent over the stack, his/her body may prevent RF energy from reaching tags in the sweater stack. If the tags in the stack are simultaneously visible to a more distant xArray, then for the duration of the "body block", the more distant xArray's reads will dominate the weighted average algorithm. Therefore, the tags' computed locations may falsely appear closer to the distant xArray.

Keep in mind that the second xArray in most instances will have a smaller read count, so drift in the algorithm likely is slight. Moreover, it is possible to replace the weighted average algorithm with another algorithm of your own devising to derive Cartesian coordinates, although such customization is out of scope for this guide.

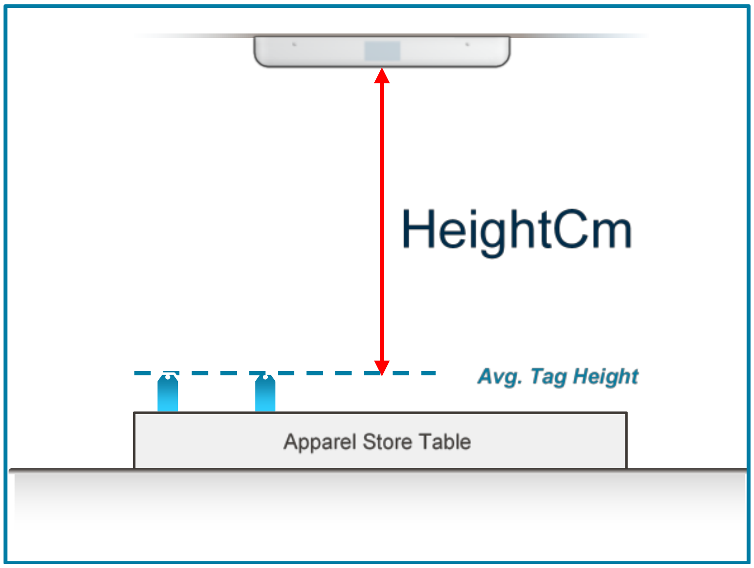
If a tag is directly under one xArray and seen by a second xArray, the second xArray will have a small weighting in the total weighted average calculation, as depicted in Figure 38.

Figure 38: Weighted Tag Average Height Measurement, Near and Far xArray Scenario



Make sure when configuring the xArray you understand that HeightCM is the distance from the xArray to the average tag height. It is not a measure of the height of the xArray to the floor (unless for some unusual reason you are trying to read tags lying on the floor). The distinction is highlighted in Figure 39. Tag heights will usually vary considerably, so you likely have to do a best guess *estimate* of the average tag height.

Figure 39: xArray HeightCM Measurement vs Avg. Tag Height



Configuration

The configuration parameters relevant to Location Mode are listed below, with information provided for setting parameters in ItemTest and via Octane SDK.

Easy Settings Recommendation

The settings in Table 8 can locate both a small or large set of tags.

Table 8: Recommended Easy Settings

SETTINGS	VALUES
Mode	1002: Autopilot Static
Session	2 or 3
Power	30 dBm
Compute Window	60 seconds
Tag Age Interval	120 seconds
Update Interval	10 seconds

Reader Mode

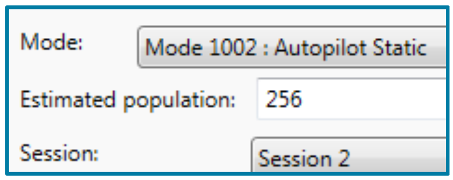
Specify one of several preset reader modes that are optimized for different conditions.

- 1002 is recommended (a.k.a. AutoPilot Static) – This mode consistently finds a greater number of tags (Figure 40). AutoPilot Static finds a greater number of tags and has a faster read rate than

Dense Reader (M=8). Because Mode 1002 will use slower, more sensitive RF modes in an effort to pick up power-challenged tags, multiple xArrays each set to 1002 and inputting to a weighted averaging technique will give more accurate location data with high probability. (See Figure 41 for settings.)

- If you have a small number of tags, limited RF interference, and need to optimize for real-time tracking, then you should use AutoPilot Static 1003 mode. This mode will provide faster (approximately 3X) read rates than 1002 but is more susceptible to interference from other RF sources and finds fewer unique tags.
- The cycle time (i.e. the time taken to move through all of the antenna beams) is dependent on the number of tags visible on a given antenna and the reader mode. The lower the number of tags visible on an antenna, then the lower the amount of time spent on that antenna. Max Throughput, which is the fastest RF mode, should result in the shortest time spent on an antenna.

Figure 40: Configure Dialog



"Mode 1002: Autopilot Static" selected

Figure 41: Reader Mode – Location Mode Parameters

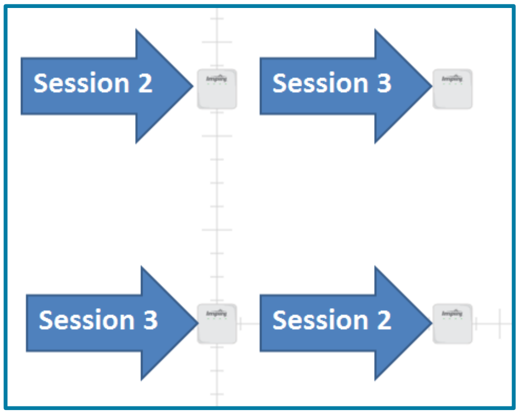
```
settings.ReaderMode = ReaderMode.AutoSetDenseReaderDeepScan;
```

Default: AutoSetDenseReader

Session

- When using Location Mode, you can choose only Session 2 or Session 3. Sessions 0 and 1 are not permitted.
- We suggest alternating adjacent xArrays between Sessions 2 and 3. An example is illustrated in Figure 42. Adjacent xArrays using the same Session can create a condition where one xArray is putting a tag into B state while the other is pushing the same tag back into A state. For an example of programming in Octane SDK, see Figure 43.

Figure 42: Adjacent xArray Alternation



Alternate adjacent xArrays between Sessions 2 and 3
For Octane SDK programming, see Figure (…)

Figure 43: Session – Location Mode Parameters

```
// First xArray:
s.Session = 2;

// Second xArray:
s.Session = 3;

// Third xArray:
s.Session = 2;

// Fourth xArray:
s.Session = 3;
```

Default: Session 2

Gen2 Filter

- Standard Gen2 filtering using C1G2 SELECT Filter as defined in LLRP guides. (Figure 44)

Figure 44: Gen2 Filter – Location Mode Parameters

```
// Define 1 Filter (EPC,TID, Reserved, User); Up to 2 Filters
s.Filters.TagFilter1.TagMask = "9999";
s.Filters.TagFilter1.MemoryBank = MemoryBank.Epc;
s.Filters.TagFilter1.BitPointer = BitPointers.Epc;
s.Filters.TagFilter1.BitCount = 16;
s.Filters.Mode = TagFilterMode.OnlyFilter1;
```

Default: No Filtering

Compute Window

This is the amount of time (i.e. the window) over which tag locations are computed. Parameters can be found on Figure 45.

- Range: [1-65535] seconds
- Default: 10 seconds.
- Tags location will improve with the number of reads. Ideally a tag will have at least **50 - 100 reads** during a single compute window.
- If tags are "static", you should increase the compute window up to 30 or 60 seconds to greatly increase location accuracy.
- If you are reading a small population of "dynamic" tags – tagged items moving around –lower the compute window to 1 or 2 seconds.
- There is an inherent set of tradeoffs between accuracy, the length of the compute window, the velocity of tags, and the number of tag reads for a given tag in each compute window computation.

TIP: Set ComputeWindow >= UpdateInterval. A longer ComputeWindow produces the most accurate X and Y location for static tags, and a shorter update interval tracks movement better for dynamic tags.

Figure 45: Compute Window – Location Mode Parameters

```
s.SpatialConfig.Location.ComputeWindowSeconds = 10;
```

Configurable in ItemTest User Interface
Default: 10 seconds

NOTE: The ItemTest log file for Location Mode shows the tag location and number of reads. If location accuracy for a specific tag is in question, check the number of reads for that tag. Increasing the compute window will give the xArray more time to read a tag thereby increase the tag's location accuracy.

Tag Age Interval

This value answers the question "How long does the gateway keep a tag in its internal database, even after the tag is no longer reading, before issuing its Tag Exit Report?" (Figure 46)

- Make sure the value of TagAgeInterval is greater than the value chosen for the Compute Window.
- Default: 20 seconds.

Figure 46: Tag Age Interval – Location Mode Parameters

```
s.SpatialConfig.Location.TagAgeIntervalSeconds = 20;
```

Configurable in ItemTest User Interface
Default: 20 seconds

Update Interval

This value governs the frequency of location update reports via LLRP. (See Figure 47 for Octane SDK settings.)

- Range: 0-65535

- Note: Locations are specified, perhaps too specifically, in centimeters, yet the xArray is generally accurate to within 1.5 meters. In some scenarios, you might see computed locations that are accurate within a meter. Inter-interval update reports can vary considerably by up to 150cm in either the X or Y dimensions. So even with stationary tags, some degree of reported location change is almost assured.
- Default: Every 5 seconds; The default 5 seconds is good starting point especially when the tag population is not in the hundreds.
- Shortening the update interval (1-2 seconds) will help when tracking a "smaller" set of dynamic/moving tags. This is if the application needs this level of granularity.
- A longer update interval is recommended when tags are static or for large population of tags.
- The default value of 5 seconds is typically adequate for most ItemTest-based experiments and demos.
- Setting the Update Interval to 0 will cause the reader to send updates as fast as possible. As of firmware 5.6, a value of 0 translates to ½ second, as shown in the tables in Location Limitations.

Figure 47: Tag Age Interval – Location Mode Parameters

```
s.SpatialConfig.Location.UpdateIntervalSeconds = 5;
```

Configurable in ItemTest User Interface
Default: 5 seconds

TxPowerInDbm and MaxTxPower

Reducing TxPower will reduce multipath (which causes inconsistent X and Y location coordinates (sometimes referred to as jitter). Keep in mind, lowering the TxPower will also reduce the xArrays ability to read the harder to reach tags. If you are experiencing multipath but are able to read all of your tags at say 25 dBm, lower the transmit power to reduce the multipath.

- Set MaxTxPower to true if the maximum transmit power is desired, false if a custom value is desired. (See Figure 48 for Octane SDK settings.)
- If MaxTxPower is set to false, then a custom power can be used. Provide power in values in increments of 0.25 dBm. (See Figure 49 for an example.)

Figure 48: MaxTxPower – Location Mode Parameters

```
s.SpatialConfig.Location.MaxTxPower = false;
```

Configurable in ItemTest User Interface
Default: True

Figure 49: TxPowerInDbm – Location Mode Parameters

```
s.SpatialConfig.Location.TxPowerInDbm = 25.25;
```

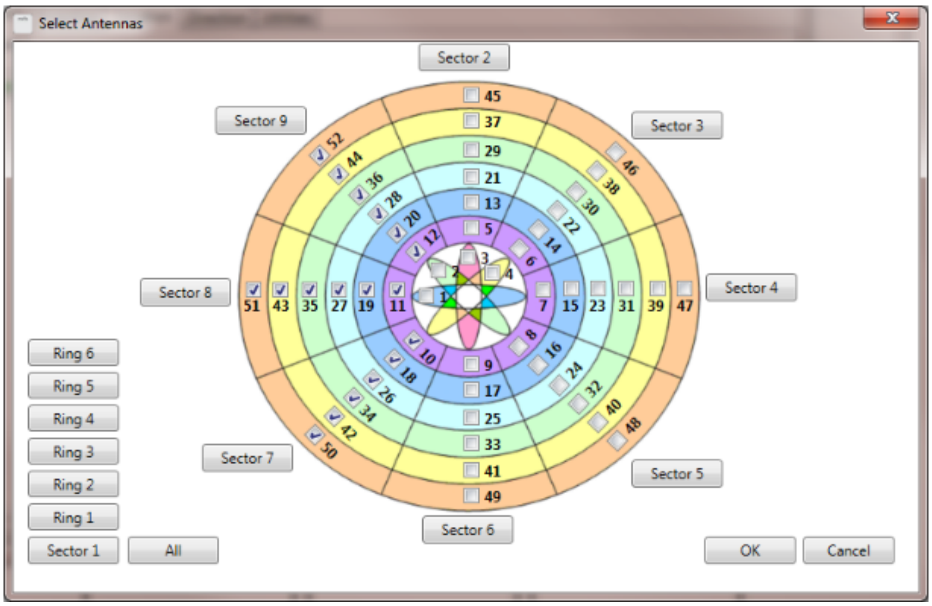
Configurable in ItemTest User Interface
Default: 30dBm

DisabledAntennaList

When a highly reflective surface (such as a metal wall) is in the xArray's field of view, disabling antenna beams can help "reduce" the amount of multipath which causes inconsistent X and Y location coordinates.

- Use the DisabledAntennaList to select antenna beams that are subject to multipath or are pointing in a direction that you don't want to read tags. The Select Antenna View is shown in Figure 50. (See Figure 51 for Octane SDK settings.)
- The antenna beams 1 – 4 (aka Sector 1) are enabled regardless so no need to disable them.
- A maximum of 24 beams can be disabled.

Figure 50: DisabledAntennaList – Select Antenna View



Choose what antennas to disable; note this same dialog is used in Inventory mode where instead you select the antennas to enable.
Default: No beams are disabled

Figure 51: Tag Age Interval – Location Mode Parameters

```
List<ushort> disabledAntennas = new List<ushort> { 10, 15 };
s.SpatialConfig.Location.DisabledAntennaList = disabledAntennas;
```

Default: Empty List

HeightCm

Specify (in centimeters) the height of the xArray to the average height of tags likely to be in its field of view. This value is NOT the height of the xArray above the floor. (Figure 52)

- If no average height can be easily estimated, choose a height corresponding to the largest group of tags at a similar height.

Figure 52: Height – Location Mode Parameters

```
s.SpatialConfig.Placement.HeightCm = 457;
```

Configurable in ItemTest User Interface
Default: HeightCm = 400

Facility X and Y

Specify the X, Y Cartesian coordinates of an xArray within a facility. (Figure 53)

TIP: A blueprint is helpful here when using multiple xArrays. By entering the facility X, Y coordinates, the xArray algorithm will provide location reports where the location is relative to the X, Y facility coordinates of the reporting xArray.

Figure 53Facility X and Y – Location Mode Parameters

```
s.SpatialConfig.Placement.FacilityXLocationCm = 0;
s.SpatialConfig.Placement.FacilityYLocationCm = 0;
```

Configurable in ItemTest User Interface
Default: FacilityXLocationCm = 0; FacilityYLocationCm = 0

Orientation

Specify the anti-clockwise angle of rotation relative to a common North-South orientation the user has chosen. (See Figure 54 for settings in Octane SDK. Figure 55 shows how +90° appears in Location Mode Orientation.)

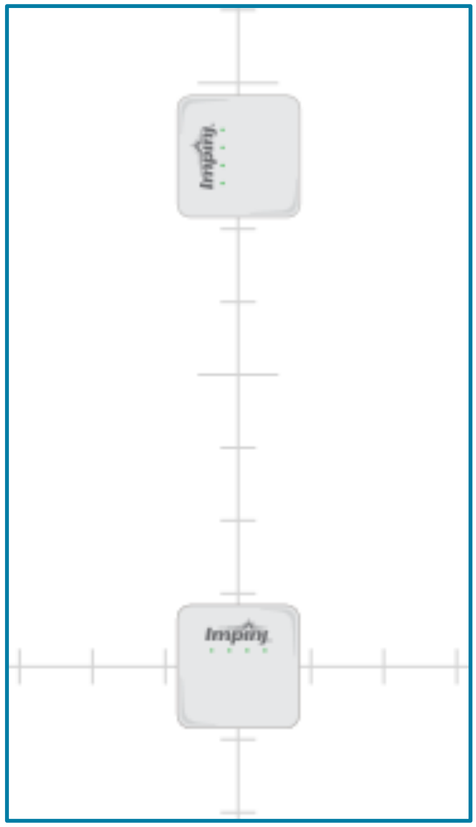
Figure 54: Orientation – Location Mode Parameters

```
s.SpatialConfig.Placement.OrientationDegrees = 0;
```

Configurable in ItemTest User Interface
Default: OrientationDegrees = 0

NOTE: In ItemTest the xArray icon rotates based on orientation. The user is viewing the xArrays in the Location UI from above even though the logo and lights are facing the user.

Figure 55: Orientation in ItemTest Location Mode



The orientation of the top xArray is +90 (degrees)

Enable Reports

Turn reporting functionality on or off. Reported data can be handled by your application. (Figure 56)

- Entry Report– a report generated when a tag is first seen or a when a previously seen tag is re-read after having aged out of the internal database.
- Update Report– a report generated every *n* seconds based on the user's entered value.
- Exit Report– a report generated when a read tag read by the xArray has not been seen for TagAgeInterval seconds.

Figure 56: Enable Reports – Location Mode Parameters

```
s. SpatialConfig.Location.EntryReportEnabled = true;
s. SpatialConfig.Location.UpdateReportEnabled = true;
s. SpatialConfig.Location.ExitReportEnabled = true;
```

Configurable in ItemTest User Interface
Default: Entry = true; Exit = true; Diagnostic = false

Tag Reports

Location mode provides tag reports to be consumed by an application, allowing that application to monitor the (x, y) location of each tag in the field of view. Only the EPC is supported. The xArray in the Location

Mode does not read or report the TID. Moreover, the xArray in Location Mode cannot read a tag's user memory. (Figure 57)

- If TID or user memory is required, you must switch the xArray to Inventory Mode and enable FastID or explicit TID/user memory read operations.
-

Figure 57: Tag Reports – Location Mode Octane SDK .NET code example

```
static void OnLocationReported(ImpinjReader reader, LocationReport report)
{
   Console.WriteLine("Type = {0}", report.ReportType);
   Console.WriteLine("EPC = {0}", report.Epc);
   Console.WriteLine("X = {0} cm", report.LocationXCm);
   Console.WriteLine("Y = {0} cm", report.LocationYCm);
   Console.WriteLine("Timestamp = {0} ({1})",
report.Timestamp,report.Timestamp.LocalDateTime);
   Console.WriteLine("Read count = {0}", report.ConfidenceFactors.ReadCount);
}
```

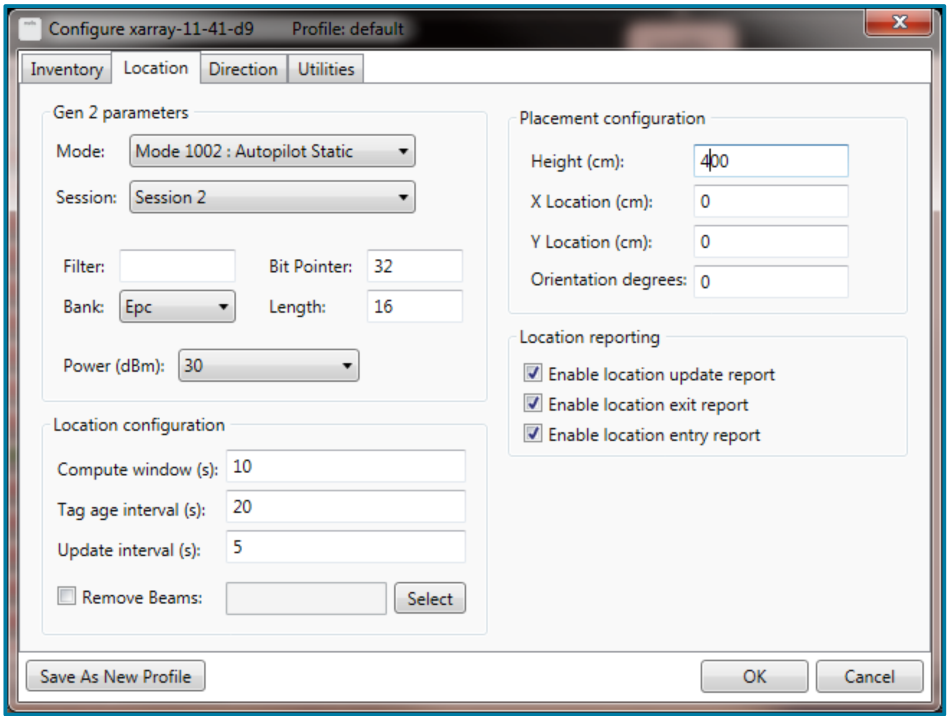
Deployment Testing

Perform the following Location tests to validate the accuracy of your deployment

Basic Location Mode Test

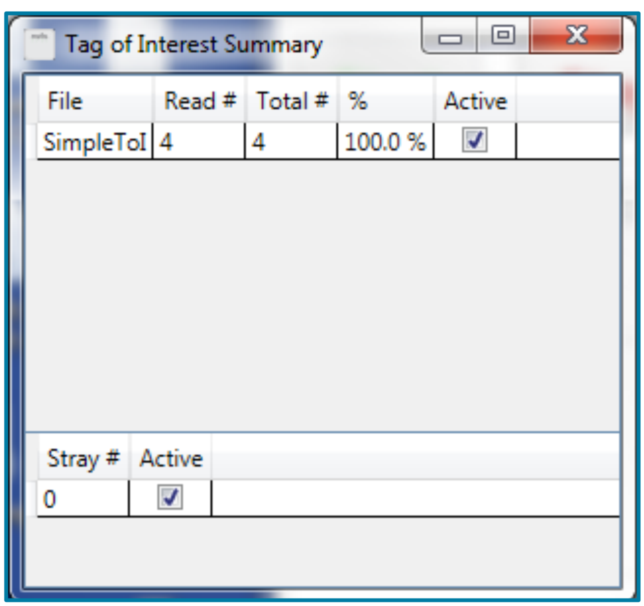
1. In ItemTest, on the Location tab, set the following parameters:
    - Mode = **AutoPilot Static (1002)**
    - Under Location reporting, check all three boxes to enable all location reports
    - Under Location configuration, review the settings. Tag age interval should be at least double the Compute window. Note the value of Compute window to prepare for the next step.
    - Mode, Location Reporting and Location Configuration parameters are correctly set in Figure 58.

Figure 58: ItemTest Basic Location Mode Test Parameters



2. Run ItemTest for a time interval at least double the value you used for Compute window.
3. To display the "Tag of Interest Summary", shown in Figure 59, click *Application Settings* on the main ItemTest screen, then click the Location tab, then select Enable Tag of Interest Summary.

Figure 59: Tag of Interest Dialog Display



Dialog shows percent of tags read from the ToI file. 4 out of 4 is 100%

4. Verify that for each tag reported, X/Y location corresponds to the actual tag position (+/- 1.5m). See Figure 60:

Figure 60: ItemTest Location Mode in Real Time

ItemTest Location Mode showing real-time computed tag locations based on last calculated location

TIP: Use color names in your TOI file to make visual identification easer in ItemTest. See ItemTest help for more information. It's important either to focus on ~ 10 to 15 tags or to assign tags in the same general area the same color. Doing this will help you easily identify location errors.

As you run your test, ask these questions:

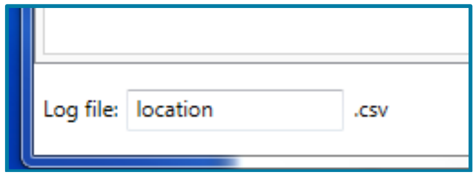
- Do objects appear where you expect them (i.e. within +/- 1.5m of reality)?
- How much tag jitter do you see? Recall that metal walls or shelving in the area may promote jittering.
- Are all of the tags in your TOI being read?
- Do you notice tag location error increases at distances farther out?

Location Log File Test

You can compare ItemTest log files with known tag information to validate read accuracy. For example, you might compare the known X and Y coordinates for each tag with the logged values. Also, as mentioned in Accuracy Rules-of-Thumb, you will want to verify that your tag read counts are adequate to insure location accuracy.

- To capture log files (which are saved by default to the <ItemTest installation folder>/log subdirectory), simply enter a name in the textbox appearing in ItemTest's bottom left corner, such as in Figure 61.

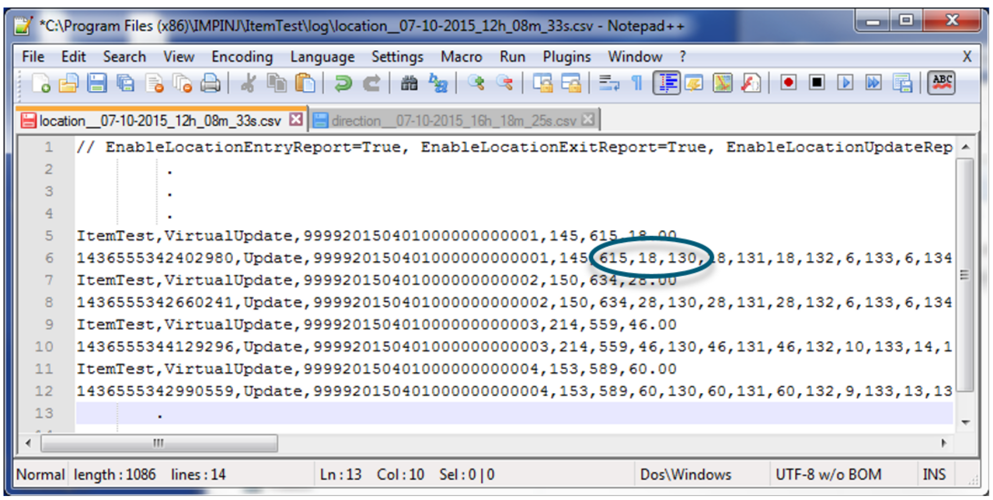
Figure 61: Location Log File, .csv



Example Location log file with 4 tag reports

- See circled region in Figure 62; Tag EPC=99992015040100000000001 was seen at X=145, Y=615 with ReadCount=18 times:

Figure 62: Location Log File



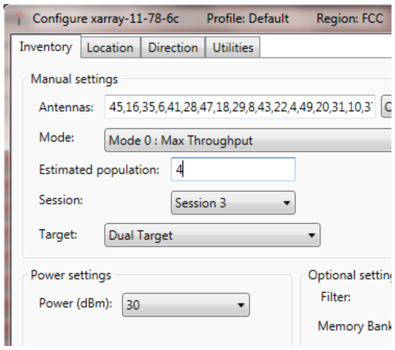
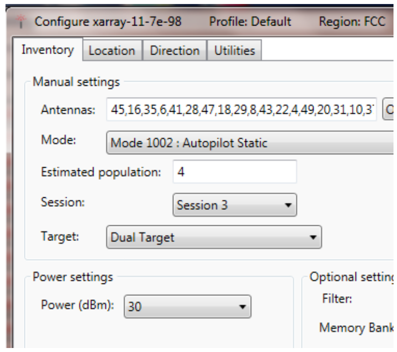
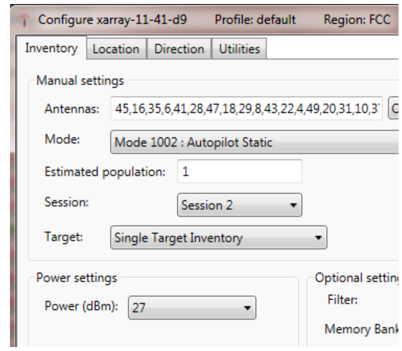
Location Mode Overdeployment Test

For Location Mode, the overlap – the percent of tags read by more than one xArray in a deployment – tells you how much weighted averaging between xArrays is applicable, and helps you see how busy your tags are having to reply to multiple xArrays (Location Mode uses the Dual Target Search mode – a Reader/Tag communication-intensive Search Mode). ItemTest does not measure overlap, but you can calculate the percent of tags each xArray has read from performing the following:

Steps to calculate the percent of tags each xArray has read:

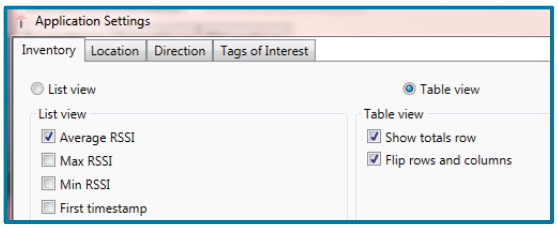
1. Create a TOI File that includes the set of tags to test. See ItemTest help for more information on creating TOI file. TIP: A handheld reader can helpful for capturing your tag EPC's.
2. Import and enable the TOI file into ItemTest.
3. Now instead of using Location Mode, simulate Location mode with Inventory Mode in Dual Target Search mode. For the best performance, alternate Session between xArrays. Table 9 shows an example of three xArray configuration dialogs showing the settings.

Table 9Three xArray Configuration Settings Comparisons

1.LOBBY XARRAY	2. MEAD XARRAY	3. RAINIER XARRAY
		

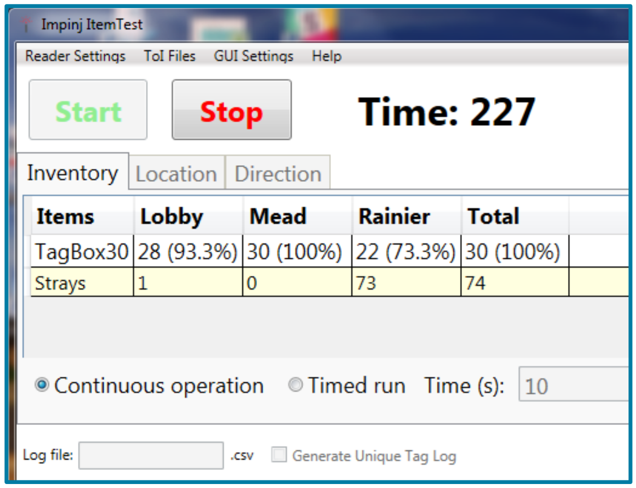
4. Configure Table View in the Application Settings menu, shown in Figure 63:

Figure 63: GUI Setting Dialog with TableView Selected



5. Run Inventory for a few minutes, as in Figure 64. Notice the Lobby (93.3%) and Mead (100%) xArrays have a large amount of overlap. Later when running Location Mode Lobby and Mead xArrays will be reading from the same set of tags and applying weighted averages. While the weighted averaging is favorable for calculating tag X and Y coordinates, xArrays reading from the same tag population will steal reads from each other reducing read counts. A similar behavior is also explained in Operating Handhelds Readers with xArray.

Figure 64: Tag Read percents by xArray



Interleaving Inventory and Location Modes

Users such as apparel retailers may deploy xArrays to ensure highly accurate inventories AND to know the locations of tagged merchandise. In Location Mode, the reader uses Dual Target mode to ensure, ideally, hundreds of reads for every tag in short periods. Conversely, to ensure the greatest number of unique tags read within the shortest periods of time, a user will likely choose Inventory Mode and use Single Target and either Session 2 or Session 3.

To meet both goals, users will want to schedule duty cycling (interleaving) between Inventory and Location Modes based on:

- The time it takes to do a full tag inventory in Inventory Mode
- The time it takes, in the Location Mode, to read 90% of those tags seen in Inventory Mode

On average, time-bound runs (e.g. 10 minutes with all xArrays operating) read between 3% and 10% more unique reads in Inventory Mode than in the Location Mode. (Session 2 or 3 with Single Target and with the majority of antennas enabled.) Given a long enough amount of time, the Location Mode should ultimately arrive at the same or close to the number of unique tags as Inventory Mode. However, because the time required to reach this goal is indeterminate, we recommend a continuous cycle of interleaving modes for a majority of use cases.

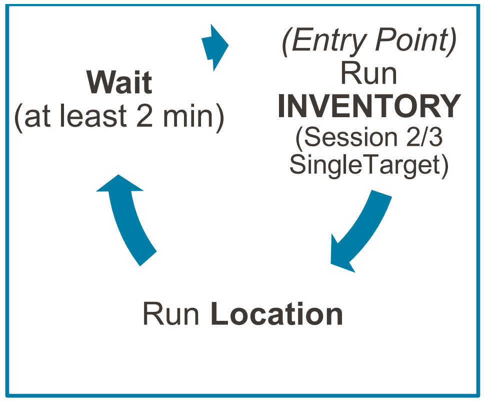
Basic Process for Interleaving between Inventory and Location Modes

Differences in tag responses to different modes can cause issues if xArrays are run in different modes back-to-back. For accurate results, a delay is required after running in Location Mode.

1. Run xArrays in Inventory Mode (Single Target, Session 2 or 3) for the appropriate period.
2. Stop the xArrays.
3. Immediately start the xArrays in Location Mode and run for the appropriate period.
4. Stop the xArrays.
5. After operating in Location Mode, you MUST wait for tags to decay before switching again to Inventory Mode. This is because the tags' inventoried flag state (A or B) are not known. See Figure 65 for a simple example of cycling the tags.
    - If your deployment is using only Monza tags, wait at least 2 minutes after stopping your xArrays and starting the next modes.

- The XArrayLocationWam example in the Octane SDK[O] provides sample code.
- With a mixed tag population (i.e. tags with multiple silicon suppliers), you will need to carefully consider the best way to accomplish your goals. Decay times can vary considerably between manufacturers and between chip models within the same manufacturer.

Figure 65: A Simple Cycle for Enabling Location, Maximum Reading



Key Parameters When Interleaving Between Inventory and Location Modes on xArray

Key parameters to consider when toggling between Inventory and Location Modes are listed in Table 10:

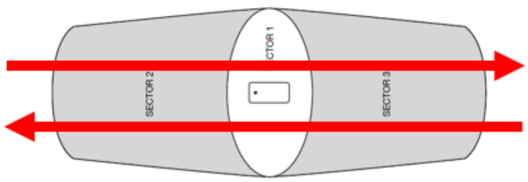
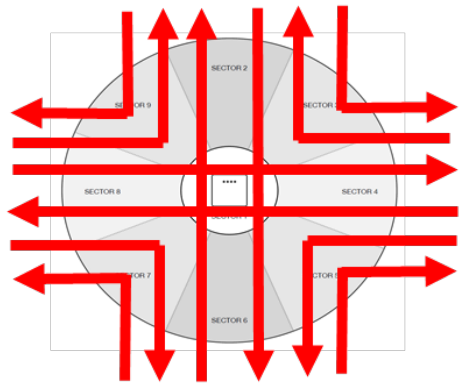
Table 10: Key Parameters for Inventory and Location Modes

PARAMETER	INVENTORY MODE	LOCATION MODE
Reader Mode	1002 Autopilot Static is recommended	1002 Autopilot Static is recommended
Session	Use either All xArrays in Session 2 or All xArrays in Session 3	Set xArrays such that adjacent xArrays alternate between Sessions 2 and 3 when configuring multiple xArrays
Search Mode	Single Target	N/A
Compute Window	N/A	Appropriate for the use case

DIRECTION MODE: CONFIGURATION AND TESTING

When a gateway (xArray/xSpan) operates in the Tag Direction Mode, it reports the direction of moving tags. Currently, the gateway can reliably track the direction of a maximum of 20 tags in high sensitivity mode and 50 tags in high performance mode. Any more, and the algorithm exhibits instability. The gateway tracks direction of movement by observing and recording tags moving between virtual sectors and through transition zones. Users frequently operate gateways in Tag Direction mode to track tags moving through intersecting hallways, open areas, corridors, and doorways. Table 11 shows the different movement tracking trajectories between both xSpan and xArray.

Table 11: Gateway Direction Tracking, xSpan and xArray

XSPAN	XARRAY
 xSpan tracks movement between 2 sectors	 xArray tracks movement between 2 to 4 sectors

Users may need to know precisely when tags are transitioning through hallway intersections or through doorways. In many healthcare settings, for instance, users may desire information regarding the direction of travel of valuable medical devices, or of a person wearing a tagged badge. Knowing “who”, “where” and “what” are key pieces of information that can be used to optimize safety and efficiency in any organization.

In Tag Direction Mode, tags enter and exit virtual sectors, generating events that let downstream software determine a tag’s direction of travel.

The key constraints when operating in the Tag Direction Mode are:

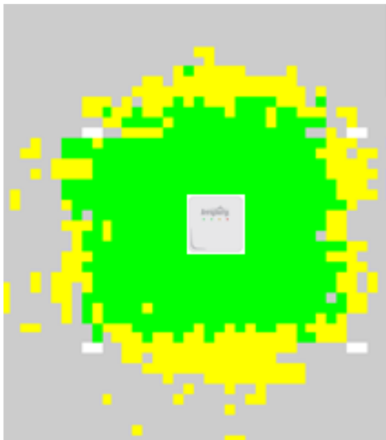
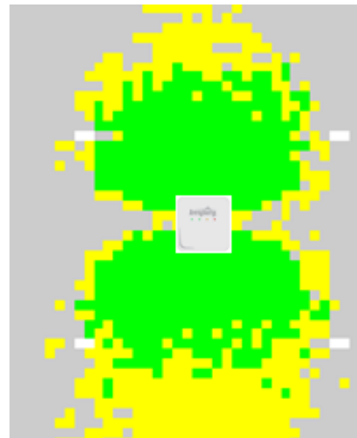
- Sectors set for observation cannot be adjacent.
- The maximum number of tags in the gateways’ field of view must be <= 20 tags in high sensitivity mode and <= 50 tags in high performance mode.

Concepts to know

- **Field of view width and overlap:** You can set the width of the coverage of the gateway’s beams. Using Tag Direction Mode, the gateway will produce large areas of coverage outside the specified sectors using both the narrow and wide fields of view, as shown in the diagrams below. However, the wide field of view produces minimal overlap between the sectors, key to effective direction tracking. In circumstances where there is a high potential for multipath and you are tracking movement across only two sectors, a narrow field of view may help reduce errors. When side mounting the xSpan, both Narrow and wide will provide tag direction. Start with narrow since it provides the more compact read zone and works well.

Table 12 shows the coverage area differences between using a Narrow and Wide field of view. Both diagrams show Tag Direction Mode operating in Sectors 2 and 6 on xArray or Sectors 2 and 3 on xSpan. Note the large areas of coverage outside the desired sectors for both the narrow and wide fields of view in the diagrams below. Also note that the wide field of view will have some space between the sectors so there is minimal overlap between the sectors.

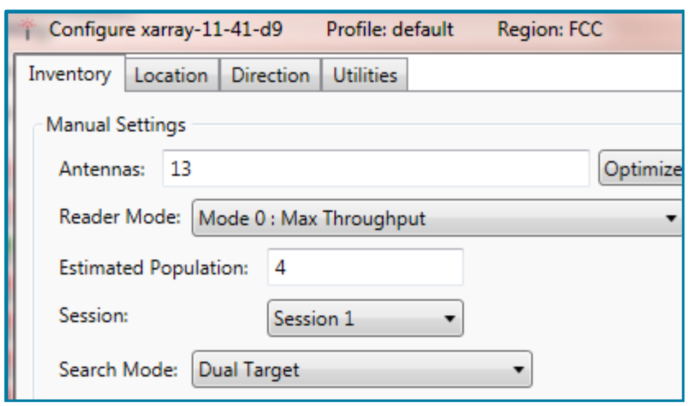
Table 12: Field of View Map

NARROW FIELD OF VIEW	WIDE FIELD OF VIEW
	
xArray: Ring 2 - beams 13 and 17 xSpan: Beams 4 and 5	xArray: Ring 4 - beams 29 and 33 xSpan: Beams 8 and 9

To map the coverage area in your location to the sectors that you selected in direction mode, perform the following steps:

1. Configure your xArray in ItemTest as shown in Figure 66: (For xSpan use antenna 4)

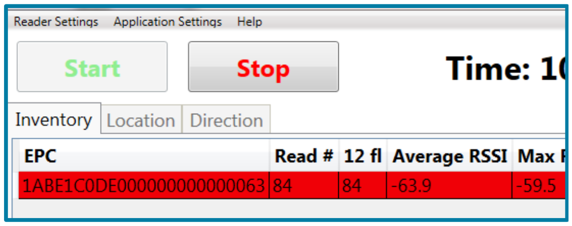
Figure 66: Configuring Antenna 13



xArray uses it for direction mode for Sector 2 / Narrow field of View

2. Run Inventory mode
3. Move the tagged item around the field of view to see where you can and not see the tag. In Figure 67 the tag became red when the tag was moved outside of Antenna 13's coverage.

Figure 67: Tag Moved outside Direct Narrow Field of view



- See the xArrayDirection example in the Octane SDK (C# and Java to learn how to program Direction Mode on both xArray and xSpan).
- Battery-assisted (BAP) tags may give inaccurate locations when operating xArray or xSpan in the Tag Direction Mode. This is because the tag does not rely on the xArray or xSpan for its power source and is likely to be visible to opposing beams. Therefore, we do not recommend using BAP tags when in the Tag Direction Mode.

Configuration

The following configuration parameters control Tag Direction Mode performance. For each we provide a table with key settings along with advice on how to choose appropriate values.

Easy Settings Recommendation

Table 13 shows the easy settings for tag direction.

Table 13: Tag Direction Easy Settings

SETTINGS	VALUES
Sectors	xArray: Enable 2 and 6 xSpan: Enable 2 and 3
Mode	High Performance
Power	30 dBm
Field of View	Narrow with 2 sectors, otherwise wide
Age Interval	4 seconds
Update Interval	2
Send Updates	False
Tag Population	20

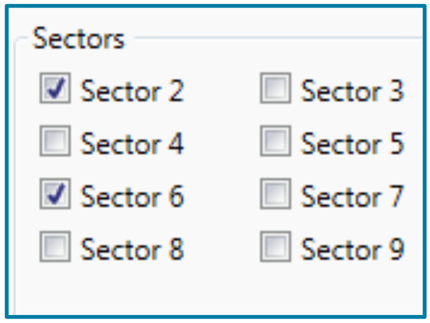
EnableSector

xArray: Enable/Disable Sectors 2 through 9. See Figure 68. (Sector 1 is reserved and is not selectable.)

xSpan: Always Enable Sectors 2 and 3.

- Default: None
- On xArray, selected sectors must be non-adjacent. Example: 2 and 3 is invalid but 2 and 4 is valid. Note that this entails a practical limit of four selected sectors (all even, or all odd). The Octane SDK code shown in Figure 69 shows an exception being thrown when Sectors 2 and 3 are selected for xArray.

Figure 68: xArray – Enable Sectors in ItemTest



Enabled Sectors are checkmarked. Sector 1 is not shown because it is reserved.

Figure 69: EnableSector – Tag Direction Mode Parameters

```
settings.SpatialConfig.Mode = SpatialMode.Direction;
DirectionConfig directionConfig = settings.SpatialConfig.Direction;
List<ushort> enabledSectorIDs = new List<ushort> {2, 3};
directionConfig.EnabledSectorIDs = enabledSectorIDs;
. . .
reader.ApplySettings(settings);
```

Mode

Use Mode to favor either performance (throughput) or sensitivity (signal reception). See Figure 70.

- Select High Performance if the area is relatively free of interference from other RF emitters. This will cause the gateway to use MaxThroughput.
- Select High Sensitivity if there are other readers nearby. This will operate the reader in one of the DRM modes using Miller encoding.

TIP: Since High Performance has a much faster link speed, it is preferred over High Sensitivity unless RF interference is an issue. High Sensitivity Mode is limited 20 tags in the field of view; Performance mode is limited 50 tags in the field of view

Figure 70: Mode – Tag Direction Mode Parameters

```
directionConfig.setMode(DirectionMode.HighPerformance);
```

Select High Performance or High Sensitivity
Default: HighPerformance

TxPowerinDbm

Specify the transmission power for ALL antennas used in the selected sectors Figure 71. While using Tag Direction Mode you cannot choose different powers for each sector/antenna.

- Range: 10 – 30 dBm
- Default: 30 dBm

TIP: Consider reducing transmission power if you are experiencing failures in Tag Direction Mode that are attributable to multipath.

TIP: With thorough testing and observation, you can adjust the power level to constrain the read-zone. One caveat of this is that as tag silicon sensitivities improve, you will need to re-evaluate zone coverage.

Figure 71: TxPowerInDbm – Tag Direction Mode Parameters

```
// Set TxPower to 28.5 dBm
directionConfig.setTxPowerinDbm(28.5);
```

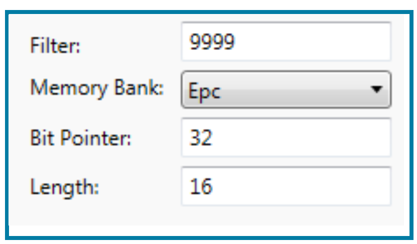
Default: 30 dBm

Gen2 Filter

Use a Gen2 filter to prevent unwanted tags in the field of view from responding. Figure 72 shows the Gen2 filters in the ItemTest UI. (For Octane SDK settings, see Figure 73.)

- You can use filtering to help limit the tag population to the Tag Direction Mode maximum of 20 tags.

Figure 72: Gen2 Filter in ItemTest



Tag Filter shown at the top level

Figure 73: Tag Age Interval – Location Mode Parameters

```
s.Filters.TagFilter1.TagMask = "9999";
s.Filters.TagFilter1.MemoryBank = MemoryBank.Epc;
s.Filters.TagFilter1.BitPointer = BitPointers.Epc;
s.Filters.TagFilter1.BitCount = 16;
s.Filters.Mode = TagFilterMode.OnlyFilter1;
```

Configurable in ItemTest User Interface; up to 2 filters
Default: No Filter

FieldOfView

This controls the depth and breadth of the field of view. Behind the scenes, this helps the gateway auto-select the appropriate sectors and antennas. Try changing this parameter if you are experiencing read problems due to multipath.

Figure 74 and Figure 75Figure 74 illustrate the problem and solution of overlapping.

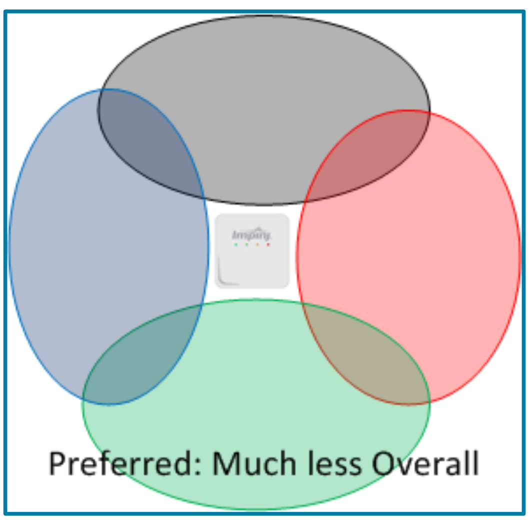
- Choices: Narrow | Wide | ReaderSelected

TIP: Use *Wide* when more than two sectors are selected; when there are just two sectors you can use *Narrow* or *Wide* (often Narrow is preferred).

Figure 74: Narrow Field of View Overlap

Narrow FieldOfView produces excess overlap in Tag Direction Mode when 4 sectors are used. (xArray only)

Figure 75: Wide Field of View Overlap



Wide FieldOfView has acceptable overlap (xArray only)

FieldOfView has three enumerations: Narrow, Wide, and ReaderSelected. (See Figure 76 for an example of FieldOfView in Octane SDK.)

- Wide: uses (outer) ring 4 on xArray
- Narrow: uses (inner) ring 2 on xArray
- ReaderSelected: Narrow if 2 sectors are enabled, otherwise 'Wide'

Figure 76: Field of View – Tag Direction Mode Parameters

```
// Select between Narrow, Wide or Reader Selected
directionConfig.setFieldOfView(DirectionFieldOfView.WIDE);
```

Default: ReaderSelected

Entry, Exit, and Update ReportEnabled

Use this parameter to select reports of interest. (Figure 77)

- Choices: Entry & Update & Exit
- Default: None
- Entry reports are generated when a tag is first read.
- Update reports are sent every "update interval" seconds, indicating that a tag is still visible to the reader typically of limited use in Tag Direction Mode.
- Exit reports are sent when a previously read (on this gateway) tag has not been read for greater than TagAgeIntervalSeconds.

Figure 77: Tag Age Interval – Location Mode Parameters

```
// Choose reports of interest - usually Entry and Exit
directionConfig.setEntryReportEnabled(true);
directionConfig.setExitReportEnabled(true);
directionConfig.setUpdateReportEnabled(false);
```

Default: setEntryReportEnabled is true; setExitReportEnabled is true; setUpdateReportEnabled is true

UpdateIntervalSeconds

This is the amount of time between gateway reports of the tag's current computed sector. (Figure 78)

- Range: [1-65535]
- Two (2) seconds is a commonly chosen value.

Figure 78: UpdateIntervalSeconds – Tag Direction Mode Parameters

```
// 2 seconds
directionConfig.setUpdateIntervalSeconds((short) 2);
```

Enter integer value; You will need to set the value to the range of [1-65535]. 2 is a reasonable choice.
Default: 0

TagAgeIntervalSeconds

This parameter specifies the time interval in seconds for the gateway to generate a tag exit report after the tag has not been read (i.e left the field of view). (Figure 79)

- Range: [0-65535]
- When setting TagAgeIntervalSeconds=0, exit reports will be generated within 0.5 seconds of a tag leaving the field of view.
- Increasing TagAgeIntervalSeconds will:
  - Increase the amount of time before notification of when a tag leaves the field of view.
  - Reduce the chance of receiving intra transitional enter/exit events due to tags moving slow enough between sectors that the `TagAgeIntervalSeconds` threshold is met. Also, with a WIDE field of view there is a greater chance of intra transitional enter/exit events because the gap between the sectors under the gateway. See Tag Direction Mode Concepts to know.

By understanding the tradeoffs, you can decide between filtering out transitional enter/exit reports, setting a high `TagAgeIntervalSeconds` threshold, or both.

Figure 79: TagAgeIntervalSeconds – Tag Direction Mode Parameters

```
// 4 seconds
directionConfig.setTagAgeIntervalSeconds((short) 4);
```

Enter integer value
Default: 5

Data: Tag Reports

Direction mode tag reports are consumed by an application, allowing that application to monitor the direction of travel of each tag that pass by the gateway's field of view.

There are 3 types of direction reports: entry, update, and exit. Table 14.

Table 14: Direction Report Types

	ENTRY	UPDATE	EXIT
Reports Occur	when a tag is first seen	while a tag is in the field of view	after a tag leaves the field of view
Report Fields	The firstSeenSector sector identifies the sector of entry	lastReadSector is the sector that the tag is in	Contains both firstSeenSector and lastReadSector sectors, that together identify the direction of movement of a tag.
Comments		Used by applications that not only need to track direction, but what sector a tag is in at a point in time.	Often, exit events are all the application needs to consume.

Figure 80: Tag Reports – Direction Mode Octane SDK .NET code sample

```
static void OnDirectionReported(ImpinjReader reader, DirectionReport report)
{
    Console.WriteLine("Direction report");
    Console.WriteLine("Type = {0}", report.ReportType);
    Console.WriteLine("EPC = {0}", report.Epc);
    Console.WriteLine("Last Read Sector = {0}", report.LastReadSector);
    Console.WriteLine("Last Read Timestamp = {0}", report.LastReadTimestamp);
    Console.WriteLine("First Seen Sector = {0}", report.FirstSeenSector);
    Console.WriteLine("First Seen Timestamp = {0}", report.FirstSeenTimestamp);
    Console.WriteLine("Tag Population Status = {0}", report.TagPopulationStatus);
}
```

Tag Reports: TagPopulationStatus

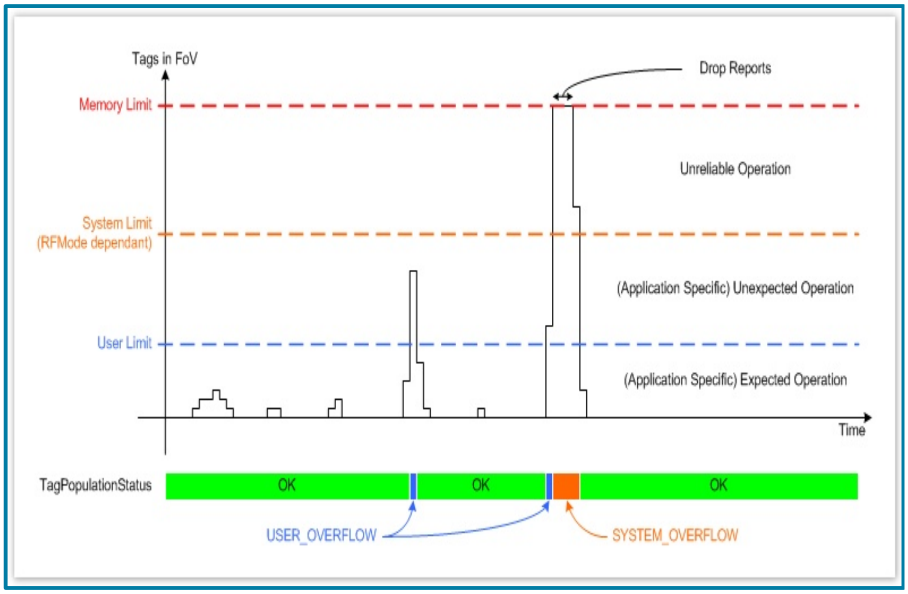
- You will want to track the `TagPopulationStatus` field to:
  - Verify the performance of the tag direction algorithm
  - Set a tag count limit based on your use case
- When it comes to verifying the performance of the tag direction algorithm, watch the *System Limit*, which is reached depending on the RF Mode used. See Table 15: for when the system limit is reached based on the number of tags in the field of view.

Table 15: System Limit Status Values

TAG POPULATION STATUS VALUES	REASON	# OF TAGS IN THE FIELD OF VIEW
OK	No limit reached	<= User Limit
USER_OVERFLOW	User limit reached	> User Limit
SYSTEM_OVERFLOW	System limit reached	> 20 in High Sensitivity Mode > 50 if High Performance

- Your code should always rely on tag population status to monitor the system limit. Keep in mind that the system limit could change in future versions of Octane firmware. In any case, you never want to exceed the system limit, which results in unreliable data.
- In the worst case, as seen in the Figure 81, the memory limit is reached resulting in dropped reports.
- As for User Limits, imagine a situation where you should never see more than five tags in the field of view at one time. Setting the User Limit to 5 will alert your code by setting `TagPopulationStatus == USER_OVERFLOW` whenever more than 5 tags are in the field of view.

Figure 81: Tag Population Status



TagPopulationStatus changes as the number of tags in the field of view increases

EXTERNAL REFERENCES

[A] Support Link: *Impinj ItemTest Software* (https://support.impinj.com/hc/en-us/articles/208589707-Impinj-ItemTest-Software) – ItemTest Software download and resource page.
[B] Support Link: *Impinj ItemSense Software Brochure* (https://support.impinj.com/hc/en-us/articles/205425027-ItemSense-Software-Brochure) – ItemSense brochure and brief for more information about ItemSense software and its capabilities.
[C] Third Party Link: *LLRP Tool Kit* (http://www.llrp.org/) – LLRP (Low Level Reader Protocol) Tool Kit features an open source library for programming readers for low-level radio frequency
[D] Support Link: *Latest Firmware, Utilities, and Development Libraries for Impinj Readers and Gateways* (https://support.impinj.com/hc/en-us/articles/204383817-Latest-firmware-utilities-and-development-libraries-for-Impinj-readers-and-gateways) – LTK and SDK releases and related documentation for readers.
[E] Support Link: *RFID Link Budget Whitepaper* (https://support.impinj.com/hc/en-us/articles/212284327-RFID-Link-Budget-Whitepaper) for more information regarding RF gains and losses and read distance.
[F] Support Link: *Monza Learning Center - App Notes and Tips* (https://support.impinj.com/hc/en-us/sections/200454528-Learning-Center-App-Notes-and-Tips) – Review the Learning Center for information about patented Monza-based technologies such as Autotune, Endura and Integra.
[G] Support Link: *RFID Link Budget Whitepaper* (https://support.impinj.com/hc/en-us/articles/212284327-RFID-Link-Budget-Whitepaper) – Review for a discussion of the physics behind overall RFID link and how to determine where link failure can occur from hardware specifications.
[H] Support Link: *Third Party xArray Mounting Options* (https://support.impinj.com/hc/en-us/articles/211473518-Third-Party-xArray-Mounting-Options) – Review for more information on xArray deployment scenarios (for authorized partners only).
[I] Support Link: *Third Party xSpan Mounting Options* (https://support.impinj.com/hc/en-us/articles/206246864-Third-Party-xSpan-Mounting-Options) – Review for more information on xSpan deployment scenarios (for authorized partners only).
[J] Support Link: *xArray Documentation* (https://support.impinj.com/hc/en-us/articles/203627878-xArray-Documentation) – Link to the xArray Partner and Documentation page
[K] Video Link: *xArray Gateway - Mounting* (https://support.impinj.com/hc/en-us/articles/204938757-xArray-Gateway-Mounting-VIDEO) – Link to an xArray gateway installation instructional video
[L] See G.
[M] Support Link: *Understanding EPC Gen2 Search Modes and Sessions* (https://support.impinj.com/hc/en-us/articles/202756158-Understanding-EPC-Gen2-Search-Modes-and-Sessions) – Link to learn more about Reader Search Modes and Sessions, including TagFocus technology.
[N] Support Link: *FastTrack Training Request* (http://hubs.impinj.com/fasttrack-training-request) – Link to request an Impinj Technical Training to learn about the basics of RAIN RFID
[O] Support Link: *Octane SDK* (https://support.impinj.com/hc/en-us/articles/204381437-Octane-SDK) – Link to the Octane SDK download and resources page

GLOSSARY

Beam: Gateways produce dual linear polarized beams. The term beam is often used as a shorthand.

Boresight: The axis of maximum gain (maximum radiated power) of a directional antenna.

Dielectric: The ability of a substance to store electrical energy in an electric field. Examples include light dielectric (cardboard and foam), medium dielectric (plastics and shirts), heavy dielectric (jeans, books, glass), and very heavy dielectric (liquids and people).

ETSI: European Telecommunications Standards Institute, ETSI A European regulatory body that governs electronic emission of signals by radios. An ETSI gateway occupies the frequency band of 865-868 MHz.

FCC: Federal Communications Commission, FCC A US regulatory body that governs electronic emission of signals by radios. An FCC gateway occupies the frequency band of 902-928 MHz.

Jitter: inconsistent tag X and Y coordinates in Location mode due to multipath.

Multipath: phenomenon that results in RF signals reaching the receiving antenna on the gateway by two or more paths as a result of bouncing off reflective surfaces such as metal or glass.

Proof of Concept: a test deployment with performance tuning per the recommendations found in this guide. Most import, the proof of concept should provide a clear decision on moving forward with a wide deployment.

Read Margin: The amount of power still available to a tag above its lowest level required to operate.

Read Range: how far away the tag can be read from the gateway's antenna.

Side Lobe: A radiation pattern formed on a perpendicular angle from the main antenna beam when the main beam is steered outward from its boresight.

Tag: Synonymous with "Label". Composed of RFID tag silicon, a tag antenna, and typically a printed label with an adhesive backing or a handing tag.

Tagged Item: (or Endpoint): An item given a digital identity by having a tag applied to it. When a Tagged item is read, consider its Tagged item performance.

Tagged item performance: The effective performance of a tag applied to an item, typically measured by the minimum power needed for a tag to operate and the power backscatter back to the reader. This is commonly referred to as the effective read range of a tag applied to an item in the band of interest. Note that the tag performance is completely frequency dependent and will vary with the materials (i.e., dielectrics) it is applied to and the surrounding environment.

xPortal: An Impinj gateway that incorporates Speedway reader and a Dual-Linear Phased Array (DLPA) antenna.

APPENDIX 1: GATEWAY SECTOR AND BEAM DIAGRAMS

Figure 82: xArray 52 beam, 9 sectors

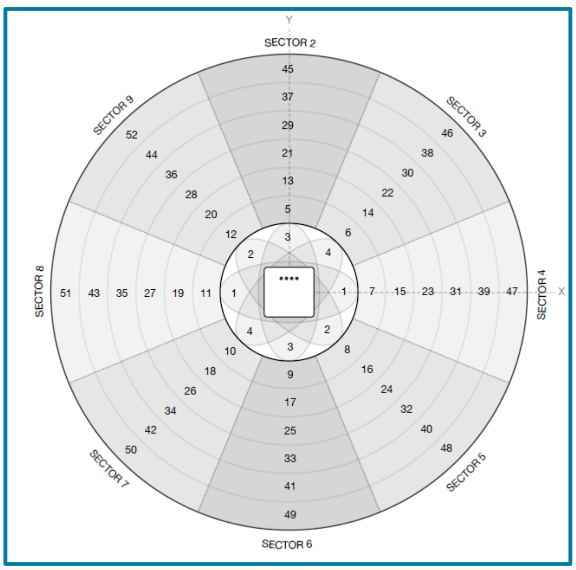
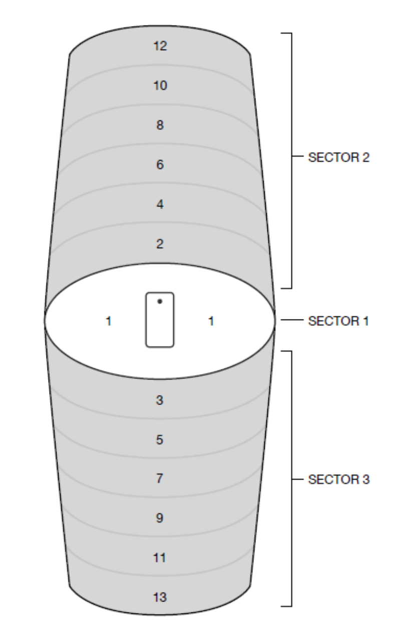


Figure 83: xSpan 13 Beams and 3 Sectors



APPENDIX 2: OPERATING HANDHELDS READERS WITH A GATEWAY

Handhelds can work in the same area as a gateway (xArray/xSpan), but performance will be affected somewhat when reading tags underneath the gateway (in the gateway's field of view). Because the gateway beams are always moving, the handheld is still able to find the tags, but tag read performance (both the tag read rate and the number of unique tag EPCs read) will drop because each tag has both a gateway and a handheld to communicate with them. We will quantify the impact on the handhelds' performance and proscribe best practices concerning the TSL 1128 and ATID AB700 handhelds sold by Impinj.

Performance Expectations and Best Practices

The following recommendations are made to maximize the unique tag reads by the handhelds. The handheld user does not necessarily know the operation of the gateway so these recommendations take this into consideration.

- Session recommendations:
  - Use Session 1 on the handheld
  - Use Session 2 and 3 on the gateway – (Typically done anyway)
- The handheld user should notice slightly slower read rates and will need to allow extra time to scan. As usual, pay attention to when the handheld indicates that it has read all of the tags in the direction it is pointed.
- Keep in mind the closer the tag is to the gateway, the stronger the RF transmission, and hence, the more impact from the gateway(s).
- Table 16 maps the state of the xArray from the status lights:

Table 16: Gateway Status Lights

XARRAY	RF
LLRP light (next to power light) is green	Likely transmitting RF
LLRP light (next to power light) is Blinking	Not transmitting RF
Status light is Blue	Likely transmitting RF
Status light is Yellow	Not transmitting RF

- Table 17 *estimates* the percentage decrease in the number of unique tag reads when the xArray is running inventory in Reader mode: AutoPilot Static (1002). Please take these numbers as Estimates**.**

Table 17: Handheld Unique Tag Read Decrease with xArray

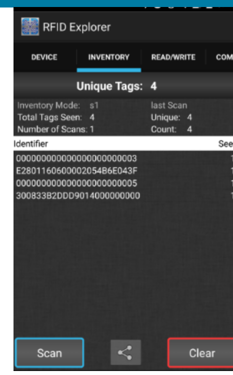
BRAND/MODEL	INVENTORY	LOCATION
TSL 1128	*1% (FCC), 0% (ETSI)*	*1%, 2% (ETSI)*
ATID AB700	*1% (FCC), 0% (ETSI)*	*1%, (FCC and ETSI)*

Decrease in the number of unique tags read for handhelds when xArray is on

- Disclaimer: These numbers will increase as more RF is introduced. Possible reasons include:
  - A deployment with multiple xArrays in close proximity.
  - A smaller room with reflective walls.
  - The xArray's Reader mode is MaxThroughput (0) which transmits the most RF.
- In a stress test with extreme conditions (tags directly under the xArray, Inventory Mode using only sector 1, dual Target search mode, and MaxThroughput) all to maximize the RF intensity, the decrease in unique tags read was about 11% for ATID AB700 and 7% for the TSL 1128 handheld.
- If using an ETSI handheld (Europe) set the handheld to "hop" between all four channels to optimize your scan under any gateway. A single gateway can use all the four channels, or multiple gateways can alternate between different frequencies. See ETSI Region and other Four Frequency Channel Regions. In any case, it is best to configure the handheld to use all four frequencies.

Tips for Handheld

Table 18: Tips for Handhelds

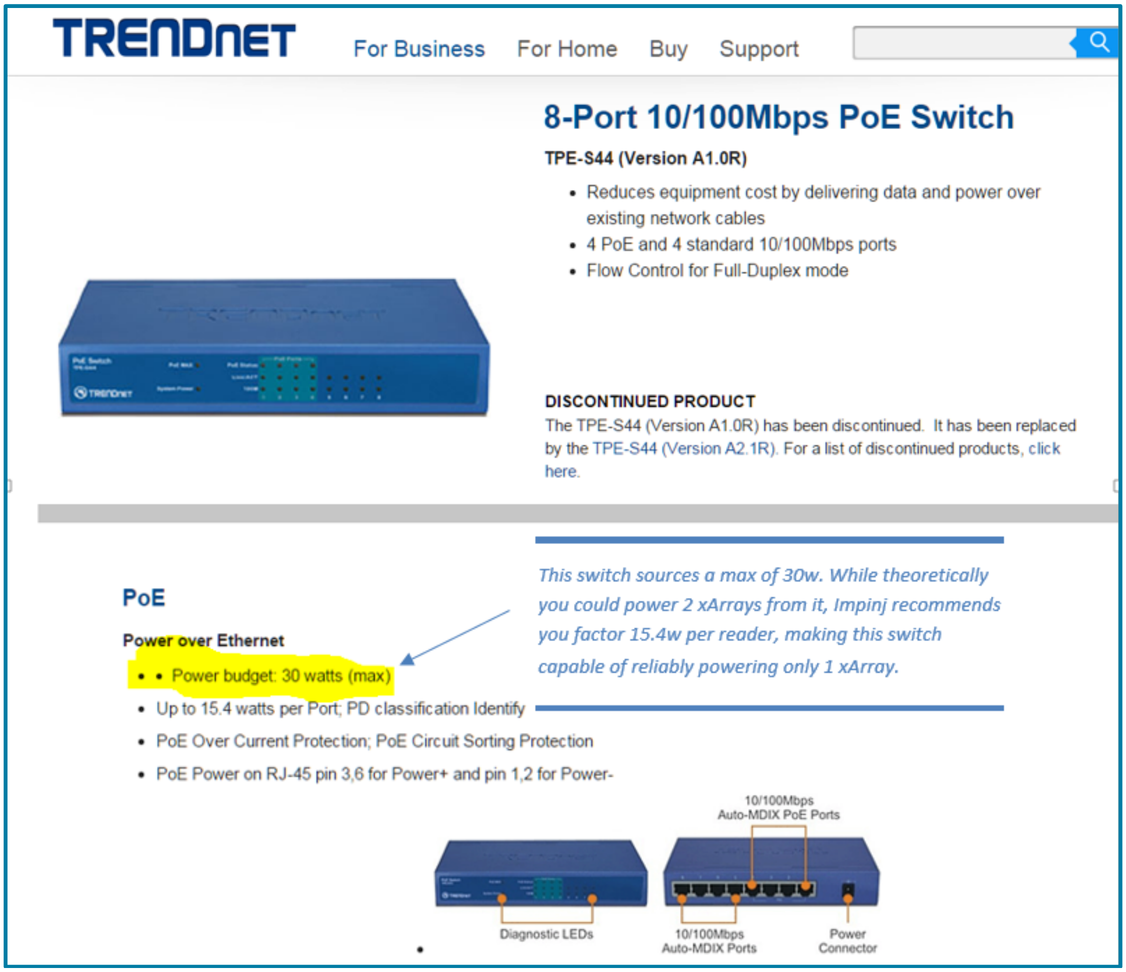
MODEL	TIP	ILLUSTRATION
TSL 1128	The TSL "Explorer" application is unable to read tags when the gateway was running Inventory Mode in Session 1 in search mode= Single target with either TagFocus on or off. The reason was that gateway Inventory run sets all of the tags to be in the B state. The TSL ignores when it runs its tag inventory. Programming the gateway to inventory using either Session 2 or 3 with the TSL running in session 1 will prevent this issue.	 TSL – RFID Explorer in Session 1
ATID AB700	Set the handheld to Session 1 and Target AB (dual target) while the gateway is running in Session 2 or 3.	Installed application DEMO_UHF can specify session 1 and Target AB

APPENDIX 3: HARDWARE CONSIDERATIONS

Anticipate your machine and network resource needs by considering the following:

Electrical power: The gateway is an 802.3af class 3 device that typically consumes 13.8 watts and draws a maximum of 15.4 watts. Running an underpowered gateway will result in tag read failures. Consider the following diagram depicting a PoE router that might seem sufficient to power two gateways, but only reliably powers one, as illustrated in Figure 84.

Figure 84: PoE Router Power Considerations



APPENDIX 4: SOFTWARE CONSIDERATIONS

Once you've completed your evaluation of gateways in the candidate environment, you need to consider how you will control them and collect the data they produce. While architecting a software system is beyond the scope of this document, we mention below some important concepts you may consider in your software design, and we provide guidance for settings parameters for use with the various Modes (Inventory, Location, and Tag Direction).

SDK Considerations

- Octane SDK .NET and Java versions provide extensive, easy to use class libraries for gateway. Also, included are sample applications for Inventory, Location, and Direction Modes. These examples provide a good starting point for developing your own application.
- The Java LTK supports reader-initiated connections back to a pre-configured server. If your architecture requires readers to initiate connections back to a home server, then you will want to use the Java LTK.
- While LTK and Octane SDK are both supported, writing applications in Octane SDK is quicker and simpler.

Scaling

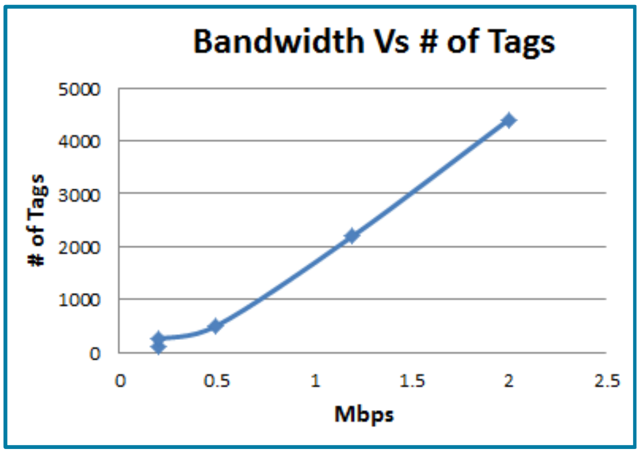
UpdateInterval (Location Mode)

On xArray, Location Mode requires a value of UpdateInterval that is sufficient without overloading the network infrastructure using Octane SDK. This is not necessary in smaller deployments and typically starts to be noticeable with ten or more deployed xArrays. CPU load may require some testing to determine, but guidance is provided in this document for setting this parameter.

Bandwidth usage in Location Mode

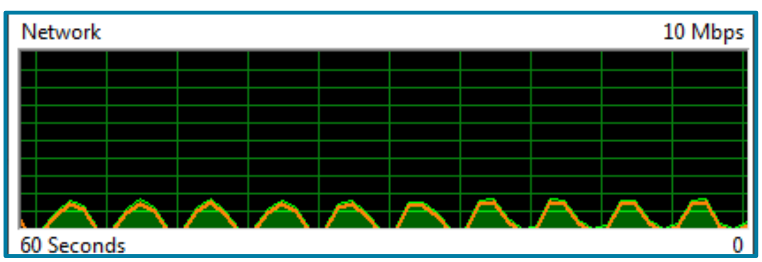
- In Location Mode, the number of tags read derives the bandwidth required as shown in Figure 85.

Figure 85: Bandwith vs. Tags



- The network traffic shown in Figure 86 pulses each time an update report is sent from the xArray to the host.

Figure 86: Bandwidth Usage Report



Bandwidth usage from xArray reading 4400 tags. Almost 2 Mbps are sent in each update report.

What to consider when assessing bandwidth requirements:

- The Frequency of each update is defined by the update interval so the update interval can be adjusted to increase or decrease bandwidth usage.
- Bandwidth usage will be related to your CPU usage since typically the bulk of the processing for location application is in response to a tag report.

- The above graph shows just one xArray. Adding additional xArrays running Location mode can scale the bandwidth requirements incrementally since often update intervals are in synch between xArrays. In other words, adding a second xArray reading that reads 2000 tags in Location Mode will scale the bandwidth requirement from 1.1 Mbps to 2.2 Mbps.

ETSI Region and other Four Frequency Channel Regions

Europe's ETSI region provides four frequency channels you can use for gateway transmission (865.7, 866.3, 866.9, and 867.5). For best performance, we recommend you alternate frequencies between adjacent gateways as you would with sessions alternating between adjacent gateways. Other regions with only four channels can alternate between adjacent gateways in the same way.

By default, in ETSI the gateway will only transmit on 866.9 MHZ, so if you don't specify your frequencies your Gateways will all transmit on the same frequency! Be sure to specify your frequencies especially when your gateways are close enough to have their field of view overlap.

For example, Table 19 shows how you could use the Octane SDK to set up adjacent gateways using the ETSI frequencies.

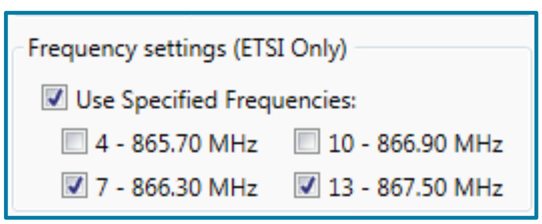
Table 19: Adjacent Gateways

GATEWAY	ADJACENT GATEWAY

`// Code to setup transmit frequency`
`List<double> freqList = new List<double>();`
`freqList.Add(865.7);`
`freqList.Add(866.9);`
`settings.TxFrequenciesInMhz = freqList;`
	`// Code to setup transmit frequency`
	`List<double> freqList = new List<double>();`
	`freqList.Add(866.3);`
	`freqList.Add(867.5);`
	`settings.TxFrequenciesInMhz = freqList;`

NOTE: The gateway supports setting transmit frequencies in Inventory and Location Modes (Not Direction) Setting frequencies is required for the gateway to minimize the interference when there are adjacent gateways. In the ItemTest example in Figure 87, the gateway will switch between the two selected frequencies.

Figure 87: ItemTest Settings to Switch Between Two Specified Frequencies



NOTE: If and only if running inventory with single target search mode, configuring each gateway to use all four channels is good option. Because each gateway will hop between all four channels with some frequency collisions, each gateway can take full advantage of the narrow spectrum when communicating with tags. Tag read performance is frequency dependent so the more frequencies that the gateway can use the better the chance of reading a tag. This configuration is not recommended for inventories with dual target search mode or location mode.

Troubleshooting

If you encounter an issue this document doesn't help you solve, please see the FAQ we maintain on our website for xArray and xSpan. For issues the FAQ doesn't address, please consult with the gateway certified partner who sold you the product. If you're a certified partner and need technical support, send email to support@impinj.com.

NOTICES