

Microsoft Azure IoT Services for Cloud Solution Provider Program

**Technical White Paper**

Authors: Brian Blanchard, Marcus Crast, Mitesh Patekar, Jason Rook

Technical reviewers: Andre Muraro, Microsoft Corporation

**Published:** May 2016

**Applies to:** Microsoft Azure IoT Services

**Summary:** This paper provides an overview of how Cloud Solution Provider (CSP) program partners can utilize various capabilities of Azure IoT Services to provide high value managed services like monitoring and analyzing the usage, performance, and uptime of customer devices in the field.

Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2015 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, Microsoft Azure, Excel, SharePoint, SQL Server, Windows, and Windows Server, are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

# Contents

[Contents 3](#_Toc451957043)

[THE INTERNET OF THINGS (IoT) OPPORTUNITY FOR MICROSOFT CLOUD SOLUTION PROVIDERS 4](#_Toc451957044)

[INTERNET OF THINGS - *CSP IN-A-BOX* 5](#_Toc451957045)

[Option A: Device Monitoring 5](#_Toc451957046)

[Option B: Device Analytics 6](#_Toc451957047)

[Technical Guides: Mastering IoT 6](#_Toc451957048)

[IoT Infrastructure: Deploying the infrastructure 6](#_Toc451957049)

[Option A: Deploying the Device Monitoring Infrastructure 7](#_Toc451957050)

[Option B: Deploying the Device Analytics Infrastructure 7](#_Toc451957051)

[Connect the Simulator 7](#_Toc451957052)

[Data Visualization 8](#_Toc451957053)

[Data Analytics (Aggregation) 8](#_Toc451957054)

[Connect a Real Sensor 9](#_Toc451957055)

[Additional Reference Materials 10](#_Toc451957056)

[What is Internet of Things 10](#_Toc451957057)

[Texas Instrument CC2560STK Multi-Standard SensorTag 10](#_Toc451957058)

[Linux alternatives to the C# IoT Core Field Gateway in the sample 10](#_Toc451957059)

[Device Management: 10](#_Toc451957060)

[IoT Ready Device Development: 10](#_Toc451957061)

[Device Best Practices: 10](#_Toc451957062)

[Misc. References: 11](#_Toc451957063)

## The Internet of Things (IoT) opportunity for Microsoft Cloud Solution Providers

Microsoft’s Cloud Solution Provider program (CSP) enables partners to create managed services practices which combine Microsoft Cloud Services with partners’ value added solution. The CSP program puts the Microsoft partner in control of the customer relationship as well as their own sales and delivery models. The power of the CSP program to allow partners to build and grow new businesses with Microsoft cloud properties is a very important initiative within Microsoft.

Internet of Things (IoT) is an equally important focus within Microsoft resulting in significant investment in Microsoft Azure’s IoT offerings. Microsoft Azure IoT Services allows customers to connect devices to a centralized cloud platform quickly, easily, and reliably to collect data from remote devices, aggregate/analyze the data, make future looking predictions, and in some cases, control the remote device. 

The opportunities for Cloud Solution Providers to build, deploy, and manage IoT centric solutions for customers are endless. Microsoft partners are using Azure’s IoT capabilities to connect devices including sensors, controls, and systems to the cloud. Thus enabling solutions that receive data at scale from those devices, and manage the authorization and throttling of those devices. Microsoft partners are also using Azure’s IoT capabilities to collect and store this ever increasing data with highly scalable compute power and extremely efficient and flexible storage models. Finally, Microsoft partners are providing real-time analytics, predictive analysis, and extensive visualization of IoT data.

With Microsoft Azure IoT Services and the CSP program Microsoft partners now have the ability to provide solutions such as:

* Managed health care device offerings where the CSP Partner sells a device or sensor, manages that endpoint, collects the data, and then delivers that data all in one bundled package with reoccurring revenue streams.
* Managed data insights for construction where the CSP collects data from industrial equipment, analyzes that data, and then provides customers with decision dashboards, all in a reoccurring revenue model.
* Managed data for any industry where the CSP handles all of the data collection, aggregation, and reporting for customers in a model that can scale up, down, and out with little to no capital investment from the CSP.

While the opportunity for CSP partners is endless it all starts with a solid technical understanding of Microsoft Azure and the IoT Services. This document, the accompanying sample code, and technical content is intended to help the CSP partner to accelerate a launch into new business offerings and models with Microsoft’s Cloud Solution Provider program and Microsoft Azure IoT Suite.

## INTERNET OF THINGS - CSP IN-A-BOX

CSP is a program designed especially for Microsoft’s partners to provide managed services. Thus it is designed using special constraints and requirements. For this reason, it does not have full feature parity with other Azure channels as of May 2016. However, Microsoft has been continuously adding new capabilities and features to CSP to get the program at par with other channels. Amongst the features or services not currently offered in the CSP portal are components like the [Azure IoT Suite](https://www.microsoft.com/en-us/server-cloud/internet-of-things/azure-iot-suite.aspx) or smaller sub-features like the Power BI output of Azure Stream Analytics. The Azure IoT Suite consists of pre-packaged solutions specifically focused on Remote Monitoring and Predictive Maintenance. Azure Stream Analytics and the Power BI adapter are two components within the Suite.

To allow CSP Partners to capitalize on the power of Azure’s IoT services, 10th Magnitude has developed a solution and series of documents that align IoT and CSP. This solution and documentation is referred to as CSP IoT in a Box.

The CSP IoT in a Box solution is designed to enable two offerings (or levels of offering) for CSP Partners. These are demonstrated in the following two unique use cases for integrating IoT into the CSP Partners suite of managed service offerings. The use cases are as follows:

* **Option A: Device Monitoring** – Monitor client device uptime and threshold violations
* **Option B: Device Analysis** – Incorporate monitoring and historical data; Go beyond monitoring to provide customer visibility into device performance, trend analysis, predictive maintenance/interactions, etc…

### Option A: Device Monitoring

The Device Monitoring use case applies to most Managed Service Providers (MSPs) in the market. Managed Service Providers are already monitoring VMs, applications, or mission critical services for their customers. MSPs are crucial to the success of their customers’ technology footprint and provide key visibility into asset stability, supportability, and state. The CSP IoT in a Box Device Monitoring solution allows MSPs to natural expand their services to also include monitoring &/or remediation of devices in the field, the same way they support VMs today.

Examples of Device Monitoring services could include things like the following:

* **Fleet:** Monitor GPS, Weight, and Speed of a fleet of vehicles to help customers see where vehicles in the fleet are located and how they are being leveraged.
* **Oil & Gas:** Monitor oil rigs to show pump status and volume of fluid movement
* **Manufacturing:** Monitor temperature, humidity, and vibration of factory equipment to show machine health threshold violations

The Device Monitoring solution for CSP IoT in a Box enables rapid solution development and an easy to integrate solution. However, its biggest limitation is in the amount/age of data being monitored and stored. If near real time monitoring is the central need for the service offering, this solution may be sufficient. The Device Monitoring solution inherently allows for 3 days of data storage and analysis, which is its biggest short coming.

If historical data, trend analysis, or predictive patterns are key to the success of the customer, then the following solution may be more suitable.

### Option B: Device Analytics

Built on the same platform as the Device Monitoring solution in the prior section, IoT CSP in a Box for Device Analytics adds persistent storage for longer term analytics and lower cost data retention. In this type of solution, Azure SQL Database (DB), DocumentDB, or Hadoop could be leveraged to provide persistent storage. For the IoT CSP in a Box solution, the team choose HDInsight (Hadoop as a Service) to provide this persistent layer of data retention. Depending on the desired level of analysis and system integration requirements, DocumentDB or Azure SQL Database (DB) could be substituted in this solution.

By enabling Device Analytics, the CSP can add historical analysis and predictive/proactive services to the service offering.

Examples of Device Analytics service offerings: (expanding examples in the prior section)

* **Fleet:** Historical data points regarding GPS, Weight, and Speed can help flag deviations (unusual driving patterns), Identify potential remediation (Deviation caused by sleepy/distraught driver), predict outcomes (Drivers with this deviation show up 1 hour later).
* **Oil & Gas:** Detect and notify customers of bursts in oil flow, blockage in lines, or other deviations that could impact barrel yields.
* **Manufacturing:** Capture deviations to gauge machine health impact, align with historical behaviors to eliminate false positives, overlay prior performance/behavior patterns to predict maintenance and avoid outages.

By aligning past performance and near-real time monitoring, CSPs can transition from monitoring current state to providing operational guidance.

## Technical Guides: Mastering IoT

Mastery of IoT for CSP requires knowledge of a four specific technical disciplines, including Cloud hosting, data management, device connectivity (or device development), data visualization.

The following technical guides and sample source code projects are designed to accelerate exposure to each area of IoT.

As a general guide, we suggest that technicians new to IoT follow a simple learning path:

* Deploy the IoT Infrastructure to Azure (Cloud Hosting)
* Connect a simulated device to Azure IoT Hub (Device Connectivity for beginners)
* Develop a dashboard based on the simulated device data (Data Visualization)
* Connect a real device to the Azure IoT Hub (Advanced Connectivity)
* Aggregate data using a Hive Script (Data Management for Device Analytics use case)

The following sections of this document will provide links to Source Code Repositories (Repo) and Technical Guides that will walk readers through each of these suggested learning paths.

### IoT Infrastructure: Deploying the infrastructure

CSP IoT in Box accommodates two use cases Device Monitoring and Device Analytics. If you are unsure which to choose please see Options A &B above and choose based upon your business requirements.

### Option A: Deploying the Device Monitoring Infrastructure

To accelerate the creation of a Device Monitoring solution, 10th Magnitude has created a series of ARM templates and PowerShell Scripts that rapidly deploy the Device Monitoring solution.

The device monitoring solution can be found in the following repo: <https://bitbucket.org/10thmagnitude/cspiot/src/>

For guidance on executing the ARM templates to create your CSP Azure IoT Infrastructure, we have created a companion guide to deploying the infrastructure: <https://bitbucket.org/10thmagnitude/cspiot>

**NOTE:** Upon deployment, the ARM templates will return 3 variables that must be captured for the following learning paths to be successful.

Host Name: iothubHostName:

Access Key Name: iotHubSharedAccessKeyName:

Access Key: iotHubSharedAccessKey:

### Option B: Deploying the Device Analytics Infrastructure

To accelerate the creation of a Device Analytics solution, 10th Magnitude has created a separate series of ARM templates and PowerShell Scripts that rapidly deploy the Device Analytics solution.

**IF YOU HAVE ALREADY DEPLOYED THE DEVICE MONITORING INFRASTRUCTURE, SKIP THIS EXERCISE**

The device analytics solution can be found in the following repo: <https://bitbucket.org/10thmagnitude/cspiot/src/>

For guidance on executing the ARM templates to create your CSP Azure IoT Infrastructure, we have created a companion guide to deploying the infrastructure: <https://bitbucket.org/10thmagnitude/cspiot>

**NOTE:** Upon deployment, the ARM templates will return 3 variables that must be captured for the following learning paths to be successful.

Host Name: iothubHostName:

Access Key Name: iotHubSharedAccessKeyName:

Access Key: iotHubSharedAccessKey:

## Connect the Simulator

Device connectivity and management can be the most complex component of any IoT implementation. In some scenarios, the device is intelligent enough to publish data to the IoT Hub via Https, MQTT, or AMQP.

However, most devices are not that intelligent. In the majority of scenarios, you may need to add some level of code to the device OR create a field gateway to connect the device to Azure IoT. The scope of this document doesn’t allow for sufficient depth to discuss all of the options available to stream line this process. For additional guidance and approaches to rapidly connecting a device, we recommend CSP Partners work with capable Microsoft Partners in the IoT business to discuss solutions. As reference [10th Magnitude](http://www.10thmagnitude.com) has a proprietary field gateway that could aid in rapidly connecting a device.

For the sake of this technical guide, we are assuming that the devices being monitored can run basic C# application code to connect to the Azure IoT Hub. As a next step on this learning path, we would advise familiarizing yourself with the way in which devices connect to the Azure IoT Hub. To aid in this learning, 10th Magnitude has created the following sample code and technical guide.

Simulated Device Repo: <https://bitbucket.org/10thmagnitude/iotdevicesimulator/src>

Readme.md Technical Guide: <https://bitbucket.org/10thmagnitude/iotdevicesimulator>

**Configuration Note:** To successfully complete the guide outlined in the ReadMe, or to get the solution in the repo to run properly, you will need the Host Name, Access Key Name, and Access Key outputted from the infrastructure deployment in the prior exercise.

**Output Note:** The simulator will randomly generate specific telemetry data as object attributes. The object and its telemetry data will then be converted to json and published to the Azure IoT Hub. It would be wise to document the json schema created in the simulator prior to proceeding to data visualization. It may also be wise to use this simulator to model json payloads prior to integrating a physical device.

## Data Visualization

To this point, the learning path has been focused on connecting synthesized sensors to get artificial telemetry data. If you’ve been following along, you should now have a feel for how the backend of Azure IoT functions. Now let’s look at the part of the solution that is most important to your customer, the actual device.

There are three primary ways to connect a device to Azure IoT:

* **Azure IoT Ready Devices:** The device being monitored can communicate over AMQP, MQTT, or HTTP/1. See the device manufacturer for details regarding the configuration to communicate with Azure IoT.
* **Connected Devices:** The device can communicate over an internet connection, but can’t communicate via an Azure IoT protocol: See the Azure IoT Protocol Gateway for more information <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-protocol-gateway/>
* **Isolated Devices:** The device is not capable of connecting to Azure IoT or a protocol gateway. For these devices, the following technical guide will facilitate the communication with Azure IoT.

Connecting an isolated device to Azure IoT, requires a field gateway of sorts. The field gateway serves as a service aggregation between the device and the cloud based IoT solution. In this example, we will walk through the use of a C# Universal Application that can run on a windows device.

In the real world, 10th Magnitude uses a very similar approach to connect isolated devices. In our standard IoT solutions, we deploy a custom built 10M IoT Field Gateway, which runs on a pre-fabricated, customized raspberry pi 2 device with built in touch-display, human inputs, device input connectors, and wifi connectivity. This gateway collects localized data from isolated devices. It then packages the telemetry data and ships it to Azure IoT over a secure channel.

The following Repo & Readme.md will describe how a simplified field gateway would connect to Azure IoT and how it would transmit data. To ensure a realistic experience, 10th Magnitude built this sample code to interact with a real device. The device used in the source code is a Texas Instrument CC2560STK Multi-Standard SensorTag <https://store.ti.com/cc2650stk.aspx>. This device connects via Bluetooth to any windows device. The source code in the repo can be used to transmit environmental data directly to the cloud. In the real world, 10th Magnitude has used this sensor and a similar field gateway to monitor environmental telemetry data of stationary machines on a factory floor.

Repo: <https://bitbucket.org/10thmagnitude/sensortagiothubuwp/src>

Readme.md: <https://bitbucket.org/10thmagnitude/sensortagiothubuwp>

**Configuration Note:** To successfully complete the guide outlined in the ReadMe, or to get the solution in the repo to run properly, you will need the Host Name, Access Key Name, and Access Key outputted from the infrastructure deployment in prior exercises.

**Configuration Note:** The sample code requires the device to be registered manually prior to running the application on a device. This is not effective at production scale. Sadly, Universal Apps do no currently allow for full use of the Azure IoT SDK; most importantly, it can’t access the namespaces required to register devices. Additional code to register the device via the API would be required to automatically provision devices on first run.

## Additional Reference Materials

What is Internet of Things: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-what-is-azure-iot/>

Texas Instrument CC2560STK Multi-Standard SensorTag:

* Where to buy: <https://store.ti.com/cc2650stk.aspx>
* Reference design: [http://www.ti.com/tool/TIDC-CC2650STK-SENSORTAG?keyMatch=cc2650 sensortag&tisearch=Search-EN-Everything#buy](http://www.ti.com/tool/TIDC-CC2650STK-SENSORTAG?keyMatch=cc2650%20sensortag&tisearch=Search-EN-Everything%23buy)

Linux alternatives to the C# IoT Core Field Gateway in the sample:

* Linux IoT Gateway SDK: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-linux-gateway-sdk-get-started/>
* Linux IoT Gateway SDK: Device to cloud messaging: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-linux-gateway-sdk-simulated-device/>
* Linux IoT Gateway SDK Source: <https://github.com/Azure/azure-iot-gateway-sdk/>

### Device Management:

* Device Management Overview: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-device-management-overview/>
* How to use Device Jobs to update firmware: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-device-management-device-jobs/>
* How to use device twins: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-device-management-device-twin/>
* How to find device twins using queries: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-device-management-device-query/>

### IoT Ready Device Development:

* Azure IoT Development Center: <https://azure.microsoft.com/en-us/develop/iot/>
* Azure IoT Hub Developer Guide: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-devguide/>

### Device Best Practices:

* Designing your solution: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-guidance/>
* Supporting Additional Protocols: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-protocol-gateway/>
* MQTT Protocols: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-mqtt-support/>

### Misc. References:

* Scaling IoT Hub: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-scaling/>
* Comparison of IoT Hubs to Event Hubs: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-compare-event-hubs/>
* Getting Started with IoT Hub for .Net: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-csharp-csharp-getstarted/>