Impinj® IOT Connector FOR   
SAP® Hybris® Commerce  
Installation Guide

Installation GUIDE

**Installation Guide**

Table of Contents

[1 Overview 1](#_Toc450679276)

[1.1 The Impinj Platform 1](#_Toc450679277)

[2 Installation Prerequisites 1](#_Toc450679278)

[3 Installing the Impinj IoT Connector for SAP Hybris Commerce with Data Hub Standalone 2](#_Toc450679279)

[3.1 Preparing the Target Server 2](#_Toc450679280)

[3.2 Building the Impinj IoT Connector 2](#_Toc450679281)

[3.3 Installing Tomcat 3](#_Toc450679282)

[3.4 Deploying Data Hub on Tomcat 4](#_Toc450679283)

[3.4.1 Data Hub WAR File 4](#_Toc450679284)

[3.4.2 Context.xml 4](#_Toc450679285)

[3.4.3 Encryption Key 5](#_Toc450679286)

[3.4.4 Local Properties File 5](#_Toc450679287)

[3.4.5 Server.xml File 6](#_Toc450679288)

[3.4.6 Log Management 7](#_Toc450679289)

[3.5 Final Hybris Configuration 7](#_Toc450679290)

[3.5.1 Configure Data Hub Extensions 7](#_Toc450679291)

[3.5.2 Start Hybris Commerce 7](#_Toc450679292)

[3.5.3 Import Data Hub Configuration Impex 8](#_Toc450679293)

[3.6 Start Data Hub 8](#_Toc450679294)

[4 Impinj IoT Connector Resource Files 8](#_Toc450679295)

[4.1 Data Hub Extension Properties File 8](#_Toc450679296)

[4.2 Master Data File 10](#_Toc450679297)

[5 Additional Information 11](#_Toc450679298)

[5.1 Impinj Resources 11](#_Toc450679299)

[5.2 Hybris Demo Store - b2c\_acc recipe 11](#_Toc450679300)

[5.3 MySQL Initial Setup 11](#_Toc450679301)

[5.4 RAIN RFID Tagging in Retail 12](#_Toc450679302)

[5.5 Public ItemSense Sandbox 12](#_Toc450679303)

[5.6 Terminology and Acronyms 14](#_Toc450679304)

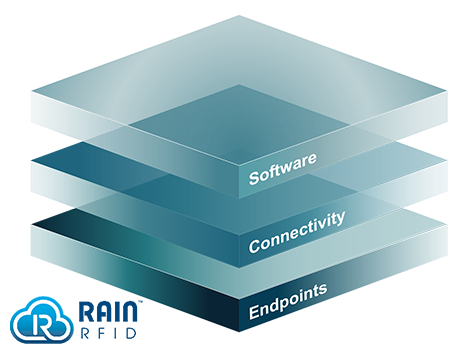
[6 Version & Revision History 15](#_Toc450679305)

[7 Notices 15](#_Toc450679306)

1. Overview

The Impinj IoT Connector for SAP Hybris Commerce is a Data Hub integration of Hybris Commerce and the Impinj Platform. Each time the Data Hub job is run, ItemSense™ software is queried for the real-time availability of RAIN RFID tagged products in the store (warehouse). This information is then processed by Data Hub and published to Hybris Commerce.

* 1. The Impinj Platform



The Impinj Platform enables companies to easily integrate ItemIntelligenceSM into enterprise applications such as Hybris Commerce. The Impinj Platform consists of endpoints, connectivity and software layers. RAIN tagged items or products are the endpoints. RAIN readers are the connectivity layer and ItemSense is the software layer. The ItemSense software aggregates and transforms raw RAIN data providing REST APIs to provide data to enterprise systems, in this case, Data Hub. ItemSense queries report on items found in the physical environment managed by the RAIN readers that are configured and managed in ItemSense. For this connector, that location is configured as a warehouse in Hybris Commerce, and will generally be a store providing real time available inventory.

Figure The Impinj Platform

1. Installation Prerequisites

To deploy the Impinj IoT Connector for SAP Hybris Commerce, you must have the following environment installed:

* Hybris Commerce (or see [Hybris Demo Store (b2c\_acc recipe)](#_hybris_Demo_Store_1))
* A Data Hub server meeting the [pre-installation requirements](https://wiki.hybris.com/display/release5/System+Requirements+-+Release+5) (if demonstration or development, the Hybris Commerce server can be used)
* A servlet container, typically Apache tomcat7 (see [Data Hub Prerequisites](https://wiki.hybris.com/display/release5/Getting+Started+with+the+Data+Hub#GettingStartedwiththeDataHub-DataHubPrerequisites) for other suggestions)
* A Database Server (see [Data Hub Prerequisites](https://wiki.hybris.com/display/release5/Getting+Started+with+the+Data+Hub#GettingStartedwiththeDataHub-DataHubPrerequisites) for supported Database Servers)
* A running ItemSense instance, RAIN readers, and SGTIN-96 encoded RAIN tagged items (Impinj Platform). Details on a [public ItemSense sandbox](#_Public_ItemSense_Sandbox) are included later in this document. To purchase ItemSense, contact your Solution Implementer or visit [Impinj for details on where to buy Impinj products](http://impinj.com/where-to-buy). See the RAIN RFID Tagging in Retail later in this document for additional details on best practices and global standards.
* A master data file defining all product that might be available in store or warehouse being the Impinj Platform monitors and manages. A sample master data file based on the sample UK Apparel store is include. Additional details can be found in the [Master Data File](#_Master_Data_File) section of this document.

1. Installing the Impinj IoT Connector for SAP Hybris Commerce with Data Hub Standalone

The following instructions detail deploying this extension on Data Hub in a standalone configuration. These instructions detail deploying Data Hub and the database on a single server. For production deployment, please follow multi-tier system recommendations in the [Hybris Commerce System Requirements](https://wiki.hybris.com/display/release5/System+Requirements+-+Release+5).

* 1. Preparing the Target Server

The following third party tools should be installed on the target server:

* Oracle JRE/JDK version 8.0 (supported for Hybris 5.5.1 +)
* Maven 3+
* git (optional to clone this extension from the [Impinj github site](https://github.com/impinj))

Identify a location to deploy Data Hub. That location will be referred to as <DATA-HUB-INSTALL-HOME> in the balance of this document.

* 1. Building the Impinj IoT Connector

Change to the <DATA-HUB-INSTALL-HOME> directory. Either unzip the zip file containing this extension or clone the Github repository.

|  |
| --- |
| To clone the github repository |
| cd <DATA-HUB-INSTALL-HOME>  git clone <https://github.com/impinj/itemsense-hybris-datahub-java.git> |

Ensure Maven (version 3 or greater) is installed and in shell path.

At this point, the Data Hub SDK is required to compile the extension. If Hybris Commerce has not yet been downloaded and installed, see Hybris demo store section later in this document.

The Data Hub SDK is included in the Hybris download and must be locally available for the extension build.

The file name will be of the form datahub-extension-sdk-5.x.y.z-RCn.jar were “datahub-extension-sdk” is the file, “5.x.y.z-RCn” is the version and “jar” is the packaging. *The version of your SDK file may vary depending on the minor version of SAP Hybris Commerce.* The file name and version for *your* Hybris installation need to be “installed” in the local Maven repository and accurately reflected in the pom.xml file for the build to succeed.

For example, for datahub-extension-5.7.0.2-RC1.jar:

|  |
| --- |
| Install the Data Hub Extension Jar in Maven |
| On a UNIX system:  cd <HYBRIS-INSTALL-HOME>/hybris/bin/ext-integration/datahub/sdk  mvn install:install-file -Dfile=datahub-extension-sdk-5.7.0.2-RC1.jar  -DgroupId=com.hybris.datahub -DartifactId=datahub-extension-sdk  -Dversion=5.7.0.2-RC1 -Dpackaging=jar  On a Windows system:  cd <HYBRIS-INSTALL-HOME>\hybris\bin\ext-integration\datahub\sdk  The maven command is the same on both operating systems:  mvn install:install-file -Dfile=datahub-extension-sdk-5.7.0.2-RC1.jar  -DgroupId=com.hybris.datahub -DartifactId=datahub-extension-sdk  -Dversion=5.7.0.2-RC1 -Dpackaging=jar |

Navigate to the itemsense-hybris-datahub-java 🡪datahub 🡪impinj directory under <DATA-HUB-INSTALL-HOME>, which is referred to as the <IMPINJ-EXTENSION-HOME> going forward, and validate (or update) the Impinj connector pom.xml file dependency version matches *your* SDK. In our example you will see:

|  |
| --- |
| Pom.xml |
| <dependencyManagement>  <dependencies>  <dependency>  <groupId>com.hybris.datahub</groupId>  <artifactId>datahub-extension-sdk</artifactId>  <version>5.7.0.2-RC1</version>  <scope>provided</scope>  </dependency>  . . . |

There are two (2) resource files, a properties file (config.properties) and the master data file used by this extension, and are found in the source tree. It is recommended you make any changes to these files in the source directory rather than the target directory so the changes are not lost upon a build.

The properties file contains the full path to the master data file and will need to be updated appropriately for your deployment. In general, these files are ready to use with the public ItemSense sandbox and the UK apparel store front installed with the b2c\_acc recipe. A deeper discussion of each of these files can be found in the [Impinj IoT Connector Resource files](#_Impinj_Connector_Resource) section of this document.

You are now ready to build the extension. Change directories to the <IMPINJ-EXTENSION-HOME>.

|  |
| --- |
| Build the Extension |
| mvn clean install |

You should now have the extension jar files built into target directories under for impinj-raw, impinj-canonical and impinj-target.

* 1. Installing Tomcat

The balance of these instructions assume Tomcat 7 will be the application server, which is shipped with Hybris Commerce. If you wish to use a different supported application server, see [Third-Party Application Servers](https://wiki.hybris.com/display/release5/Third-Party+Application+Servers) on the Hybris wiki.

Connect your browser to <https://tomcat.apache.org/download-70.cgi> to download the latest version of Tomcat 7 for your system. Install Tomcat in the <DATA-HUB-INSTALL-HOME> directory. The Tomcat root directory will be referred to as <TOMCAT-INSTALL-HOME> going forward.

Hybris recommends creating a startup file for tomcat. Samples of startup files as well as instructions for using SSL HTTPS for communicating to and from Data Hub can be found here: <https://wiki.hybris.com/display/release5/Downloading+the+Data+Hub+and+Configuring+It+for+Standalone+Use+with+Tomcat>.

* 1. Deploying Data Hub on Tomcat
     1. Data Hub WAR File

The Data Hub WAR file must be deployed to Tomcat. These instructions use a context.xml file to identify the file, but it can also be copied to the <TOMCAT-INSTALL-HOME> 🡪 WEB-INF 🡪 lib directory and renamed to datahub-webapp.war.

The WAR file is found 1 directory over from the SDK file used previously and has specific version numbers as the SDK jar did. *Take caution as you configure Tomcat to correctly specify the version you are deploying.*  The WAR file can be found here:

|  |
| --- |
| Find the Data hub war file |
| On a UNIX system:  cd <HYBRIS-INSTALL-HOME>/hybris/bin/ext-integration/datahub/web-app  On a Windows system:  cd <HYBRIS-INSTALL-HOME>\hybris\bin\ext-integration\datahub\web-app |

* + 1. Context.xml

The context.xml file specifies the WAR file and required jars. In our case we will create datahub-webapp.xml for the datahub-webapp. Create the following directory structure: <TOMCAT-INSTALL-HOME>/conf/<enginename>/<hostname> where:

* enginename (Catalina) specifies the highest level container to receive requests and
* hostname defines a virtual host (localhost) under the engine which in turn can support may Contexts or webapps

For in-depth Tomcat 7 information , see [Apache Tomcat 7 Documentation Index](http://tomcat.apache.org/tomcat-7.0-doc). For additional Data Hub information, see [Downloading the Data Hub and Configuring It for Standalone Use with Tomcat](https://wiki.hybris.com/display/release5/Downloading+the+Data+Hub+and+Configuring+It+for+Standalone+Use+with+Tomcat).

Create datahub-webapp.xml file in the <TOMCAT-INSTALL-HOME>/conf/Catalina/localhost directory with the following data, being sure to specify correct paths, the correct version of the WAR files, and correct JDBC driver jar for the database server you will use.

|  |
| --- |
| datahub-webapp.xml |
| <Context antiJARLocking="true"           docBase="<HYBRIS-INSTALL-HOME>/hybris/bin/ext-integration/datahub/web-app/datahub-webapp-5.7.0.2-RC1.war"           reloadable="true">      <Loader className="org.apache.catalina.loader.VirtualWebappLoader"              virtualClasspath=                  "<TOMCAT-INSTALL-HOME>/lib/;                   <PATH-TO-JDBC-DRIVER>/mysql-connector-java-5.1.18.jar;                   <IMPINJ-EXTENSION-HOME>/impinj-raw/target/\*.jar;                   <IMPINJ-EXTENSION-HOME>/impinj-canonical/target/\*.jar;                   <IMPINJ-EXTENSION-HOME>/impinj-target/target/\*.jar; |

* + 1. Encryption Key

Copy an encryption key into the Tomcat lib directory. The key used for developer trails can be downloaded and used for this exercise: [hybris 5 Developer Training Trail - Data Hub Environment Setup - Step 1](https://wiki.hybris.com/display/tr52/hybris+5+Developer+Training+Trail+-+Data+Hub+Environment+Setup+-+Step+1)

|  |
| --- |
| Encryption Key File |
| On a UNIX system:  cp <ENCRYPTION-KEY-FILE> <TOMCAT-INSTALL-HOME>/lib/.  On a Windows system:  COPY <ENCRYPTION-KEY-FILE> <TOMCAT-INSTALL-HOME>\lib\. |

* + 1. Local Properties File

The local properties file provides key information for the extension to integrate with the Hybris Commerce system as well as key information on the data hub extension. Note that targetsystem.hybriscore url below assumes Hybris Commerce is deployed to the same server with default targetsystem values. If that is not the case in your environment, update all targetsystem values including the datahub.extension.exportURL appropriately.

This is also the place the database server is specified. The example below is for MySQL. For additional details or a different database server, see [Configuring the Data Hub for Use with a Relational Database](https://wiki.hybris.com/display/release5/Configuring+the+Data+Hub+for+Use+with+a+Relational+Database).

Create a local.properties files under <TOMCAT-INSTALL-HOME>/lib folder and add the following data:

|  |
| --- |
| Local.Properties |
| ##############################  ###  DATA-HUB INTEGRATION  ###  ##############################  #  #common settings  #the targetsystem properties for main Hybris Commerce system  #  targetsystem.hybriscore.url=http://localhost:9001/datahubadapter  targetsystem.hybriscore.username=admin  targetsystem.hybriscore.password=nimda    sapcoreconfiguration.autocompose.pools=GLOBAL  sapcoreconfiguration.autopublish.targetsystemsbypools=GLOBAL.HybrisCore  sapcoreconfiguration.autopublish.sleeptime:5000  sapcoreconfiguration.autopublish.initialsleeptime:5000    ##################  ###  DATA-HUB  ###  ##################  #  # URL and credentials for data hub target datahub.extension.exportURL=http://localhost:9001/datahubadapter  datahub.extension.userName=admin  datahub.extension.password=nimda  # name of the Data Hub encryption key  # (default name of key from hybris trails)  datahub.encryption.key.path=encryption-key.txt  #data hub data url and port (match server.xml)  datahub.server.url=http://localhost:9080/datahub-webapp/v1    sapidocoutboundadapter.usedynamicidocnumberoffset=true    ###############  ###  MYSQL  ###  ###############  #dataSource.driverClass=com.mysql.jdbc.Driver    dataSource.className=com.mysql.jdbc.jdbc2.optional.MysqlDataSource  # database host = localhost  # database Name = integration  dataSource.jdbcUrl=jdbc:mysql://localhost/integration?useConfigs=maxPerformance  #  # administrative user on localhost integration db  dataSource.username=usr\_datahub  dataSource.password=usr\_datahub    #media storage  #media.dataSource.className=com.mysql.jdbc.jdbc2.optional.MysqlDataSource  #media.dataSource.jdbcUrl=jdbc:mysql://localhost/integration?useConfigs=maxPerformance  #media.dataSource.username=usr\_datahub  #media.dataSource.password=usr\_datahub |

* + 1. Server.xml File

Modify server.xml from {TOMCAT\_DIR}/conf directory with the following values so it does not collide with Hybris Commerce. Specifically, 8009 is the default AJP port for Hybris and 8080 will be claimed by the default tomcat applications applications for the tomcat bundled with Hybris.

|  |
| --- |
| Server.xml |
| <!-- A "Connector" represents an endpoint by which requests are received  and responses are returned. Documentation at:  Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)  Java AJP Connector: /docs/config/ajp.html  APR (HTTP/AJP) Connector: /docs/apr.html  Define a non-SSL HTTP/1.1 Connector on port 9080  -->  <Connector port="9080" protocol="HTTP/1.1"  connectionTimeout="20000"  redirectPort="8443" />    <!-- Define an AJP 1.3 Connector on port 9009 -->  <Connector port="9009" protocol="AJP/1.3" redirectPort="8443" /> |

* + 1. Log Management

To enable log management, place logback.xml in <TOMCAT-INSTALL-HOME>/webapps/datahub-webapp/WEB-INF/classes/. A sample file can be found in github repo at datahub/tomcat/webapps/datahub-webapp/WEB-INF/classes/logback.xml or on the Hybris wiki in the [Data Hub Solution book](https://wiki.hybris.com/display/release5/Data+Hub+Solution+Book).

* 1. Final Hybris Configuration
     1. Configure Data Hub Extensions

Hybris Commerce configuration must be updated to enable Data Hub integration. To do so, validate or update <HYBRIS-INSTALL-HOME>/hybris/config/localextensions.xml with the datahubadapter and datahubbackoffice extensions:

|  |
| --- |
| Localextentions.xml |
| <extension name="datahubadapter" />  <extension name="datahubbackoffice" /> |

You will need to run “ant clean all” from the command line to update your Hybris Commerce per the Hybris wiki. See [Locating and Configuring the Data Hub Adapter](https://wiki.hybris.com/display/release5/Locating+and+Configuring+the+Data+Hub+Adapter) for additional information. If deploying the demo store see the Hybris Demo Store - b2c\_acc recipesection below to proactively configure these extensions

* + 1. Start Hybris Commerce

The following start method is specified by the Installer instructions:

|  |
| --- |
| Start Demo Store Hybris Commerce |
| cd <HYBRIS-INSTALL-HOME>/installer  ./install.sh -r b2c\_acc start |

* + 1. Import Data Hub Configuration Impex

Finally, access to Hybris admin console (<http://localhost:9001/login.jsp>). Login (default credentials are admin/nimda or use credentials in your implementation). Navigate to Console 🡪 ImpEx Import and import the following text:

|  |
| --- |
| Import Data hub configuration Impex |
| INSERT\_UPDATE BackofficeRole;UID[unique=true];locname[lang=en];description;groups(uid);backOfficeLoginDisabled  ;datahubadmingroup;Datahub Admin Group;Datahub Admin Group;;false;    INSERT\_UPDATE UserGroup;UID[unique=true];groups(uid)[mode=append]  ;admingroup;datahubadmingroup;    INSERT\_UPDATE DataHubInstanceModel;instanceName[unique = true];instanceLocation  ;impinj-datahub;http://localhost:9080/datahub-webapp/v1;    REMOVE DataHubInstanceModel;instanceName[unique = true];  ;localhost; |

Be sure to modify the URL and port according to your data hub server deployment.

* 1. Start Data Hub

Start Data Hub using the Tomcat server start up file reference above. If you did not create one, see [Installation of the Data Hub on a Tomcat Server](https://wiki.hybris.com/display/release5/Downloading+the+Data+Hub+and+Configuring+It+for+Standalone+Use+with+Tomcat#DownloadingtheDataHubandConfiguringItforStandaloneUsewithTomcat-InstallationoftheDataHubonaTomcatServer) for operating system specific details, including System Variable to set and samples of how start Tomcat.

1. Impinj IoT Connector Resource Files

There are two resource files to be modified with the Data Hub extension. Example file are included with the extension and are found in <IMPINJ-EXTENSION-HOME>/impinj-canonical/src/main/resources.

* 1. Data Hub Extension Properties File

This file provides implementation specific information to execute scheduled Data Hub jobs. The properties file must be name config.properties and must be in the extension class path. Below is a sample config.properties file defining 1 job to run every 15 seconds connecting to the public ItemSense sandbox.

|  |
| --- |
| extension config.properties |
| ### Number of jobs to create ###  jobs=1  ### Jobs properties ###  ### Jobs must be in order starting from 1, and should have jobs.x. as prefix, followed by the key value ###  ## every 15 seconds ##  jobs.1.cronexpression=0/15 \* \* \* \* ?  # ItemSense base URL  jobs.1.endpointurl=http://sandbox.solutions.impinj.net/itemsense  # ItemSense username  jobs.1.username=HybrisExtension  # ItemSense password  jobs.1.password=impinj2016  # hybris Commerce warehouse being monitored and managed by ItemSense  #(this one is from the UK apparel store)  jobs.1.warehouse=ap\_warehouse\_e  jobs.1.hybrismasterdatapath=/HybrisFashionWebStoreMasterData.txt  # Filter for items reported by itemsense in the last 10 minutes (600 seconds)  jobs.1.itemlookbackwindowinseconds=600 |

The table below defines the name value pairs that make up the config.properties file.

|  |  |  |
| --- | --- | --- |
| Name | Value | Description |
| jobs | 1 | Number of Data Hub jobs configured in this file. Note: this value appears only 1 time in the file |
|  |  |  |
| **Each job will have the following name value pairs** | | |
| jobs.1.cronexpression | 0/15 \* \* \* \* ? | Schedule for this data hub job. See apache quartz for format details (<http://www.quartz-scheduler.org/documentation/quartz-1.x/tutorials/crontrigger>) |
| jobs.1.username | ItemSense username | User configured in ItemSense. Must have DataReader role assigned and be able to inspect for a running job. See [ItemSense user role definitions](http://developer.impinj.com/itemsense/docs/api/) on the [Impinj Developer Portal](http://developer.impinj.com/) for additional information. |
| jobs.1.password | ItemSense user password | Password configured for ItemSense User |
| jobs.1.warehouse | ap\_warehouse\_e | Warehouse name in Hybris Commerce to update with ItemSense data received in this Data Hub job |
| jobs.1.endpointurl | ItemSense URL | ItemSense URL |
| jobs.1.hybrismasterdatapath | File-path-to-master-data-file | Path and name of the Location of master data file that is used to map the EAN values decoded from the EPCs values ItemSense returns and map them to product code in Hybris Commerce. Data Hub web app must have read permission. The values specified must match Hybris Commerce |
| jobs.1.epcprefix | Null or a pattern to match | String identifying the first few character of the EPC to match to filter out tags found by ItemSense |
| jobs.1.itemlookbackwindowinseconds | Number of seconds (N) | Any Item not updated by ItemSense with the last N seconds will be filtered out and not added to the available inventory |

* 1. Master Data File

The master data file is a basic file that containing product codes and EANs that may be found in the warehouse monitored and managed by ItemSense. The file is specified in the config.properties by job so it is possible to have a different master data file per job or the file can be shared between jobs. This would come into play for example if a retailer had a single Hybris implementation supporting multiple brands.

The purpose of the master data file is twofold. It is a tool to map the EAN values decoded from EPCs read by ItemSense to product codes which is how the available inventory data is loaded in to Hybris. It also provides the possible inventory that might be in the location so products with quantity of 0 can be updated/set to 0.

The format is the product code followed by a pipe (|) followed by a valid EAN code. The EAN value must be valid values. It is worth noting many of the EAN values in the example apparel\_UK store are not valid. Additional information on valid EANs can be found in the [RAIN RFID Tagging in Retail](#_RAIN_RFID_Tagging) section of this document.

1. Additional Information
   1. Impinj Resources

* [www.impinj.com](http://www.impinj.com)
* [developer.impinj.com](http://developer.impinj.com/) – ItemSense documentation including the API reference, code samples and information on installing, configuring and using ItemSense
* [github.com/impinj](http://www.github.com/impinj) – Visit Impinj on github for this extension and other tools in support of the Impinj Platform.
  1. Hybris Demo Store - b2c\_acc recipe

Hybris Commerce provides recipes to facilitate setting up development or demonstration environments. The Impinj IoT Connector for SAP Hybris Commerce has been tested with the b2c\_acc recipe.

On the target machine, select the directory for the Hybris installation. For reference, this directory will be referred to as the <HYBRIS-INSTALL-HOME>. For windows, select a directory close to the system root directory with no spaces in it, such as c:\hybris on windows.

* Extract the Hybris Commerce installation zip into the <HYBRIS-INSTALL-HOME>.
* Go to the <HYBRIS-INSTALL-HOME> directory and navigate to the installer directory.
* Consider enabling Data Hub as part of the b2c\_acc recipe to proactively address the [Final Hybris Configuration](#_Final_Hybris_Configuration_1). To do this, add extName ‘datahubadapter’ and extName ‘datahubbackoffice’ to the list of extensions in the build.gradle file in the b2c\_acc recipe directory.
* Per the wiki instructions for [Sample Scenarios Installation](https://wiki.hybris.com/display/release5/Sample+Scenarios+Installation#SampleScenariosInstallation-SystemRequirements), Hybris Commerce can be installed, initialized and started the b2c\_acc recipe using the following commands:

|  |
| --- |
| Hybris Demo Store |
| **On Windows:**  install.bat –r b2c\_acc  install.bat –r b2c\_acc initialize  install.bat –r b2c\_acc start  **On Linux or Mac:**  ./install.sh –r b2c\_acc  ./install.sh –r b2c\_acc initialize  ./install.sh –r b2c\_acc start |

* To configure Hybris Commerce to use a supported database server, see: [Provide Custom Database Configuration](https://wiki.hybris.com/display/release5/Installation#Installation-4.(Optional)ProvideCustomDatabaseConfiguration) on the Hybris wiki. MySQL is referenced in the Data Hub deployment instructions above.
  1. MySQL Initial Setup

These instructions are UNIX specific, but can easily translate to Windows. Install MySQL for your operating system.

For MySQL, you must validate the [mysqld] section of /etc/mysql/my.cnf contains the following lines. Restart mysql if my.cnf configuration is updated.

|  |
| --- |
| /etc/mysql/my.cnf |
| [mysqld]  character-set-server = utf8  innodb\_flush\_log\_at\_trx\_commit=0 |

|  |
| --- |
| Mysql Database and accounts |
| Add the database, user and password for data hub:  mysql -u root –p  create database integration;  create user 'usr\_datahub'@'localhost' identified by 'usr\_datahub';  grant all privileges on integration.\* to 'usr\_datahub'@'localhost';  To confirm:  select schema\_name from information\_schema.schemata;  select user, host from mysql.user; |

The user and database configured Data Hub Tomcat local properties file must match that which is created above. Also, this process can be used if MySQL is deployed for Hybris Commerce, with a different database name. It’s recommended you specify different administrative users as well to simplify trouble shooting.

* 1. RAIN RFID Tagging in Retail

RAIN RFID tagged products are typically encoded following the Serialized Global Trade Item Number (SGTIN) or Serialized GTIN global standard. The SGTIN builds on the Universal Product Code (UPC) and the European Article Number (EAN) bar codes which provide retailers and brand owners a standard way to uniquely identify their products. While the UPC is used in North America and the EAN, which has 1 more digit than the UPC, is generally used by retailers outside of North America, they are both part of the same standard. SGTIN-96 is a specific mapping of 96 bits of memory encoded into an RFID tag following a specific pattern defined in the [GS1 EPC Tag Data Standard](http://www.gs1.org/epc/tag-data-standard).

This extension decodes EPCs using the SGTIN-96 pattern to arrive at an EAN and serial number. This data is used to count the number of items found and by mapping EAN to product code, update Hybris Commerce with the actual number of items on hand.

It is worth noting if the EAN or UPC encoded into the tag is not valid, it will not decode to same value. This generally manifests with an invalid check digit (the right most digit). Multiple web tools are available to encode/decode/validate EAN and EPC including <https://www.gs1us.org/resources/tools/epc-encoder-decoder> and <http://formvalidation.io/validators/ean/>.

* 1. Public ItemSense Sandbox

A public ItemSense is available for testing the Impinj IoT Connector for SAP Hybris Commerce.

URL: <http://sandbox.solutions.impinj.net/itemsense>

UserName: HybrisExtension

Password: impinj2016

The sandbox contains 18 tags encoded with EANs found in the apparel UK storefront. Occasionally some of the tags are removed from the field of view of the Impinj Platform. The EAN for these tags is correctly represented in the included sample master data file. Notice the tags all have a “3014” prefix. Also notice some of the EANs have been updated from the Hybris Apparel store to be valid.

Available tags are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product Code | Original Hybris EAN | Valid EAN | EPC (SGTIN-96 encoded) | Sequence Number |
| 300047513 | 383203982 | 0000383203982 | 30140005FC13EB8000000001 | 1 |
| 30140005FC13EB8000000002 | 2 |
| 300310300 | 542440755 | 0000542440753 | 30140008782B0AC000000001 | 1 |
| 300022356 | 632871132 | 0000632871139 | 30140009E055124000000001 | 1 |
| 300045375 | 657195251 | 0000657195258 | 3014000A4413114000000001 | 1 |
| 29533 | 836246217 | 0000836246214 | 3014000D10180B4000000001 | 1 |
| 3014000D10180B4000000002 | 2 |
| 45572 | 886280602 | 0000886280602 | 3014000DD81B670000000001 | 1 |
| 300040462 | 893541837 | 0000893541833 | 3014000DF434E9C000000001 | 1 |
| 29531 | 989514105 | 0000989514109 | 3014000F7432348000000001 | 1 |
| 300046592 | 1019425517 | 0001019425518 | 3014000FEC298DC000000001 | 1 |
| 3014000FEC298DC000000002 | 2 |
| 3014000FEC298DC000000003 | 3 |
| 3014000FEC298DC000000004 | 4 |
| 3014000FEC298DC000000005 | 5 |
| 3014000FEC298DC000000006 | 6 |
| 300015407 | 1033044592 | 0001033044597 | 3014001024045AC000000001 | 1 |
| 3014001024045AC000000002 | 2 |

* 1. Terminology and Acronyms

|  |  |  |
| --- | --- | --- |
| Term | Acronym | Description |
| European Article Number | EAN | World-wide 13 digit barcoding standard bar code numbering scheme used internationally to identify consumer products. A superset of UPC, and governed by [GS1](http://gs1.org/). |
| Electronic Product Code | EPC | Syntax for unique identifiers assigned to physical objects, unit loads, locations, or other identifiable entity playing a role in business operations.  GS1's EPC Tag Data Standard (TDS) specifies the data format of the EPC, and provides encodings for numbering schemes – including the GS1 Keys -- within an EPC. See [GS1 EPC/RFID](http://www.gs1.org/epc-rfid) |
| GS1 |  | Global, neutral, non-profit standard organization that manages the GTIN system which identifies companies and their products and services as used in UPC, EAN and other bar codes as well as EPCs for RFID tags. See [gs1.org](http://gs1.org/) |
| Global Trade Item Number | GTIN | Globally unique identification number used to identify products and services in global supply chains. It is also used as an umbrella term for the UCC.EAN data structures. See [www.gtin.info](http://www.gtin.info/) |
| RAdio frequency IdentificatioN | RAIN | Shorthand reference to RAIN RFID |
| RAIN RFID |  | Wireless technology that connects billions of everyday items to the internet.  RAIN RFID is a global alliance promoting the universal adoption of UHF RFID technology (Gen 2 Standard) See [rainrfid.org](http://rainrfid.org/). |
| Radio Frequency IDentification | RFID | Generic term for technologies that use radio waves to identify (read and capture) information on a tag attached to an object |
| Serialized Global Trade Item Number | SGTIN | Serialized GTIN which can be used uniquely represent each instance of a product (GTIN) |
| SGTIN-96 |  | Standard EPC scheme using 96 bits of memory for a retail Point-Of-Sale system. Defined in the [GS1 EPC Tag Data Standard](http://www.gs1.org/epc/tag-data-standard) |
| Universal Product Code | UPC | 12 digit barcoding standard bar code numbering scheme used in North America to identify consumer products governed by [GS1](http://gs1.org/). |

1. Version & Revision History

|  |  |  |
| --- | --- | --- |
| Version number | change description | date |
| 1.0.0 | Document Created/Released | 05/10/2016 |

1. Notices

Copyright © 2016, Impinj, Inc. All rights reserved.

Impinj gives no representation or warranty, express or implied, for the accuracy or reliability of information in this document. Impinj reserves the right to change its products and services and this information at any time without notice.

EXCEPT AS PROVIDED IN IMPINJ’S TERMS AND CONDITIONS OF SALE (OR AS OTHERWISE AGREED IN A VALID WRITTEN INDIVIDUAL AGREEMENT WITH IMPINJ), IMPINJ ASSUMES NO LIABILITY WHATSOEVER AND IMPINJ DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATED TO THE SALE AND/OR USE OF IMPINJ PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY OR INFRINGEMENT.

NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY PATENT, COPYRIGHT, MASKWORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT IS GRANTED BY THIS DOCUMENT.

Impinj assumes no liability for applications assistance or customer product design. Customers should implement adequate design and operating safeguards to minimize risks.

Impinj products are not designed, warranted or authorized for use in any product or application where a malfunction may reasonably be expected to cause personal injury or death, or property or environmental damage (“hazardous uses”), including but not limited to military applications; life-support systems; aircraft control, navigation or communication; air-traffic management; or in the design, construction, operation, or maintenance of a nuclear facility. Customers must indemnify Impinj against any damages arising out of the use of Impinj products in any hazardous uses.

Trademarks

Impinj, Monza, Speedway, xArray are trademarks or registered trademarks of Impinj, Inc. All other product or service names are trademarks of their respective companies. For a complete list of Impinj Trademarks visit: [www.impinj.com/trademarks](http://www.impinj.com/trademarks).

Patents

The products referenced in this document may be covered by one or more U.S. patents. See [www.impinj.com/patents](http://www.impinj.com/patents) for details.