Key features of the ThingWorx platform include:

1. \*\*IoT Connectivity:\*\* ThingWorx supports a wide range of protocols and standards to connect with different IoT devices and sensors, such as MQTT, CoAP, OPC-UA, REST, and more.

2. \*\*Data Ingestion and Management:\*\* The platform can handle vast amounts of data generated by IoT devices, store it efficiently, and manage data streams.

3. \*\*Real-time Analytics:\*\* ThingWorx allows users to apply real-time analytics to incoming data, enabling them to gain insights, detect patterns, and make data-driven decisions.

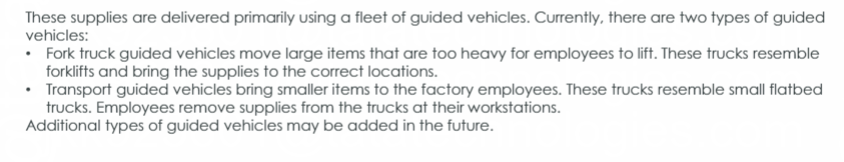
4. \*\*Application Development:\*\* It provides tools for creating custom IoT applications without extensive coding knowledge, using a drag-and-drop interface and reusable components.

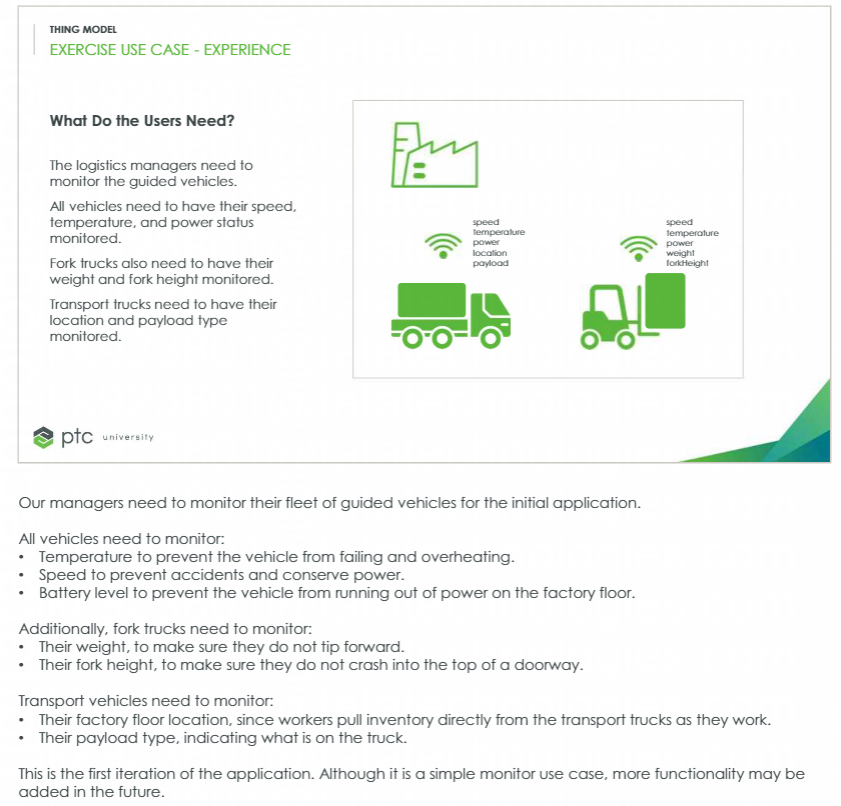
5. \*\*Remote Monitoring and Control:\*\* Users can remotely monitor connected devices and even control them through the platform, making it suitable for industrial automation and maintenance.

6. \*\*Integration Capabilities:\*\* ThingWorx facilitates integration with other enterprise systems, such as ERP, CRM, and other third-party applications, enabling seamless data exchange.

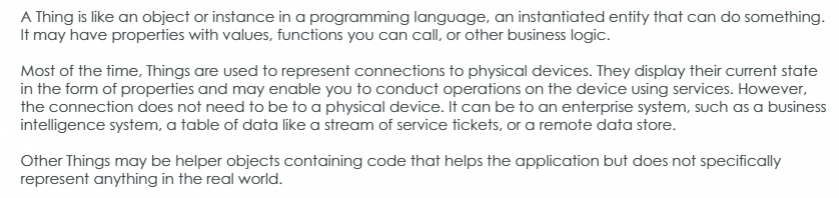
7. \*\*Security and Compliance:\*\* The platform emphasizes security, with features like authentication, access control, and encryption to protect data and devices.

10. \*\*Visualization and Dashboards:\*\* ThingWorx offers customizable dashboards and data visualization tools to display data in a meaningful and easy-to-understand manner.

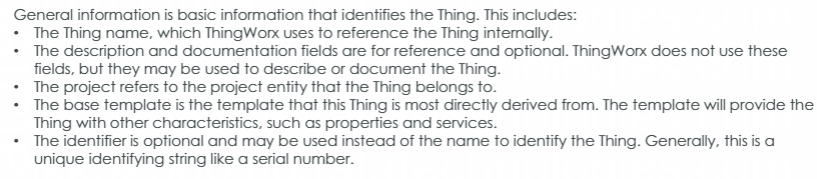
****



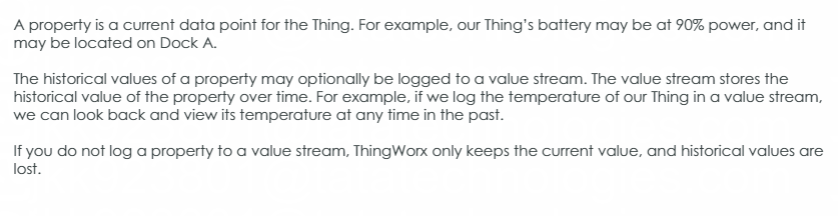
**What is Thing**



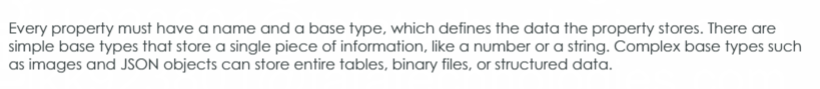
**Genaral Information**



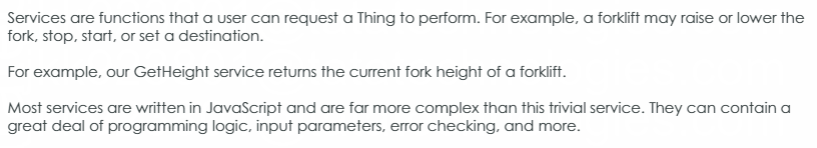
**Property and value stream**



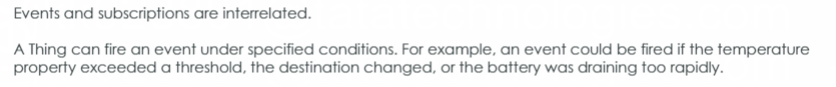
**Property Base Type**

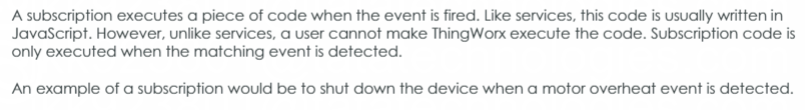


**Services**



**Events and Subscriptions**





BGV: Base Guided Vehicle

Thing Template == M1.BGV.TT

Property: EngineTemp

Speed

Power

Fork Vehicle Thing Template:

M1.FGV.TT >>> Base Thimng Template == M1.BGV.TT

Property: MaxLoad

CurrentLoad

ForkHeight

Transport Vehicle Thing Template == M1.TGV.TT

Property: Location

PayloadType

ThingShape: Battery Powered Thing shape == M1.Battery.TS

Property: PctCharge

IsCharging

**Model Tag:** ThingWorx model tags are a mechanism to label ThingWorx objects and data to assist in grouping, filtering, and locating ThingWorx objects and searching/finding data efficiently.

**Notifications:** As an Administrator, you can use notifications to send e-mail or text messages to ThingWorx users based on an [event](https://support.ptc.com/help/thingworx/platform/r9/en/ThingWorx/Help/Composer/Things/ThingEvents.html#wwID0EA46V).

For that purpose required mail extension.

**Schedulers:** To run the archive, purge, and cleanup services of the Audit Subsystem automatically, configure the corresponding Scheduler Thing to trigger the operation.

**Timer:** Timers are an entity type in ThingWorx that trigger events on a defined interval.

**Mashup:** The Mashup Builder is the tool you use to create your visualization web pages in ThingWorx, and is where individual mashups are defined. A mashup is a ThingWorx web page.

**Master:** A master provides consistent framing of a mashup’s contents (similar to the concept of a master page or template), and is commonly used for items that display throughout the mashup, such as logos, menus, and titles. For example, to display a company logo throughout your mashup, a master can be created to display the logo, so that no matter where the user is in the mashup, they will see the company logo in the same place.

**Gadgets**: Gadgets are reusable contained mashups that have additional metadata to handle the sizing requirements of a dashboard. Gadgets can display historical or current data. Any widget and service can be used in a gadget.

Gadgets allow you to dynamically assign the displayed property values. For example, multiple display gadgets (LED, Gauge, or Chart) can be created, so that you can select the view of the temperature data displayed on your dashboard.

**Data Tables:** A data table is similar to a standard relational database table, but differs greatly in terms of performance. Generally, data tables should be used if you have less than 100,000 rows of data. For larger datasets, a relational database should be used and then connected via a Database Thing template. A Data Shape defines the columns or fields of the data table.

**Streams:** 1. Streams represent time series data.

2. A ThingWorx stream is a list of activities from things or data associated with Things.

**Value Stream:** Value Streams are a mechanism for logging time-series data. Value Streams are similar to Streams in terms of their end results but are much simpler to set up. Creating and populating Value Streams does not require any scripting.

**Persistence Providers:** In ThingWorx, value streams, streams, data tables, blogs, and wikis are data providers. Data providers are databases that store run time data.

Create Project: ThingWorxTraining or TML\_Demo

Thing Shape: VendingMachineShape

Property: 1.CurrentHumidity

2. VMHumiditySetpoint === Persist

3. CurrentTemperature

4. VMTemperatureSetpoint === Persist

5. FieldServiceTech === Base Type: UserName == Persist

6. Location === Persist

Type Link in Chrome: http://localhost:80/Thingworx/ThingShapes/

Thing Template: VendingMachineTemplate == Generic Thing == VendingMachineShape

Thing: VM-TR1 == VendingMachineTemplate

Duplicate the Thing VM-TR2 == VM-TR3

Mashup: VendingMachineMash

Layout == Add Left

Divide it with 6 Halfs 3 x 2

Drag List Widget on first cell

add VendingMachineTemplate === GetImplementingThings

Drag and Drop All Data

DisplayField == name

AutoSelectFirstRow == Check

\*\*\*\*\*\*Add Gadget Widget to the Mashup\*\*\*\*\*

add VendingMachineTemplate === GetImplementingThings

Click on Dynamic Check Box

GetProperties

in GetimplimentingThings == Return Data == drag Name entity to DynamicThings's Entity Name

GetimplimentingThings == Drag the SelectedRowsChanged event to the GetProperties service

drag Gauge widget to 2 cell

drag Current Temp to Gauge

\*\*\*\*\*\* Add LED Widget \*\*\*\*\*\*

Drag LED Widget to % cell

CurrentHumidity from GetProperties

\*\*\*\*\*\* Add Open Street Map \*\*\*\*\*

Add Open Street Map to cell 3

ShowSelectionMarker == check

Drag Location

Dynamic == SetProperties

Add Numeric Entry Widget below LED widget named it Set Temperature === CurrentTemperature and Set Humidity == CurrentHumidity

Drag the OpenStreetMap Location Picker

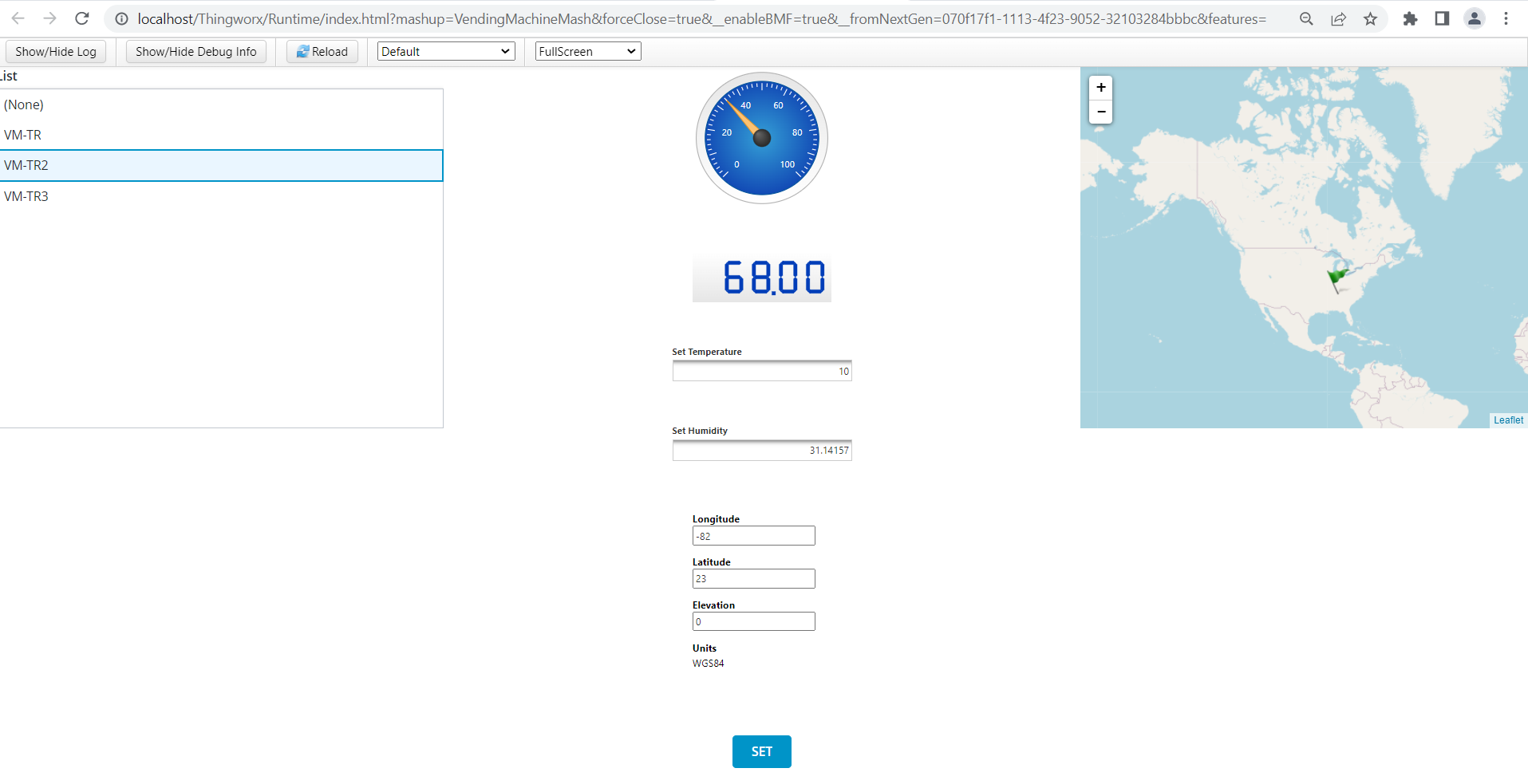
Drag Location from the Properties Panel to Location under the SetProperties service.

Add Button === SET

Click Event === Setproperty

ServiceInvokeCompleted == GetProperties

|  |  |  |  |
| --- | --- | --- | --- |
| Set Temp | 40 | 34 | 10 |
| Set Humidity | 80 | 68 | **31.14157** |
| Longitude | **-105.14** | **–86** | -82 |
| Latitude | **39.95** | 40 | 23 |
|  | **VM-TR2** | VM-TR3 | VM-TR2 |



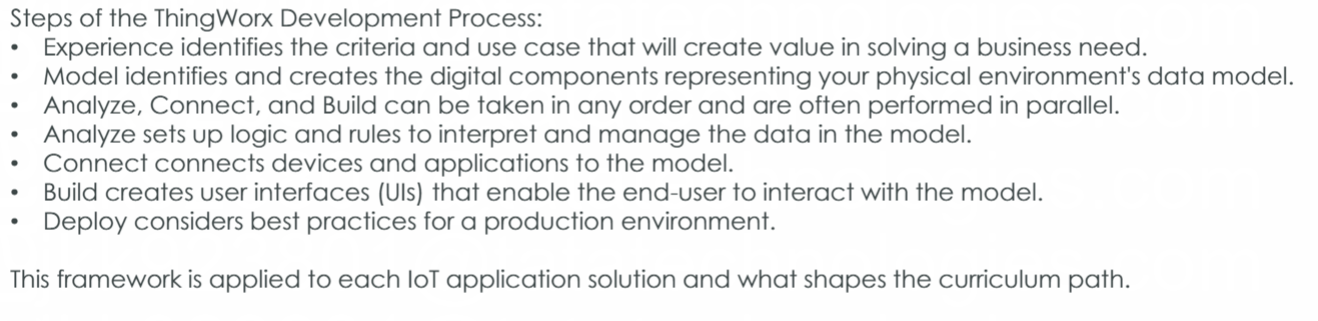
Edit the VendingMachineShape ThingShape.

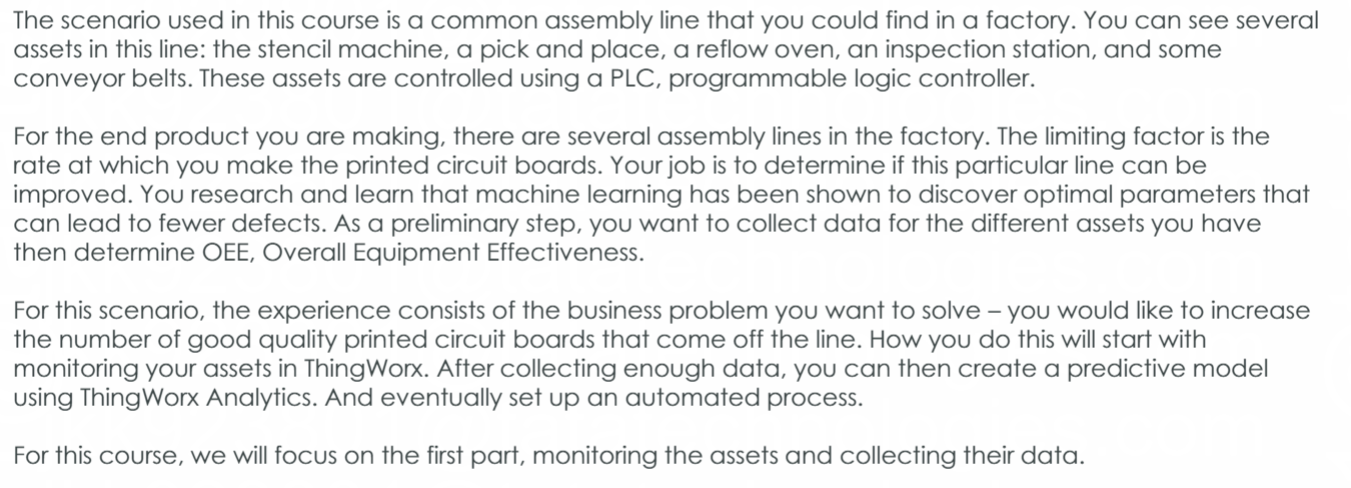
**VMTempMax** === **Default Value = 135 == persist**

**VMTempMax** === **Default Value = -20 == persist**

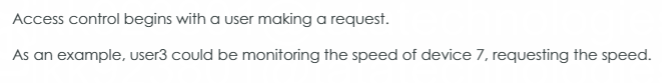
GetProperties service == Add **VMTempMax  & VMTempMax**

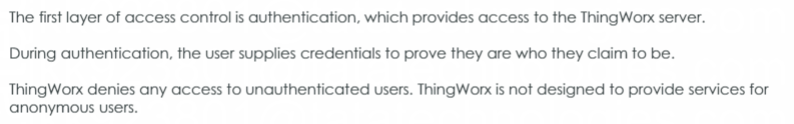
**Thingworx Development Process**

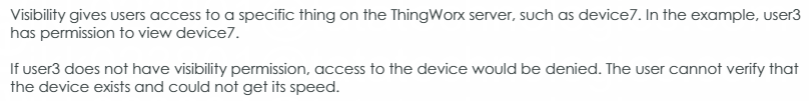


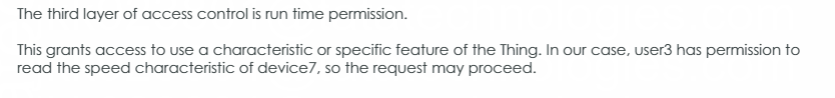


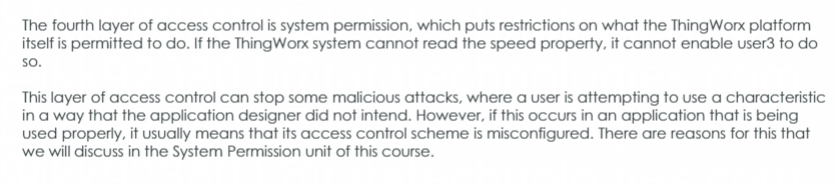
**Authentication**

****

****

****

****

****

**Kepware Server**

ThingWorxKepwareServer

Chaneel2 === Device1 == 123Demo Data == Bind to New Entity == Remote Thing

Thing == EngineTemp Project TML\_Demo

Add Property == DataChange1

Subsription== EngineTempTest

Subsription Info == Enabled

if(eventData.newValue.value <= 20){

me.DataChange1 = "Changes Requiered";

}

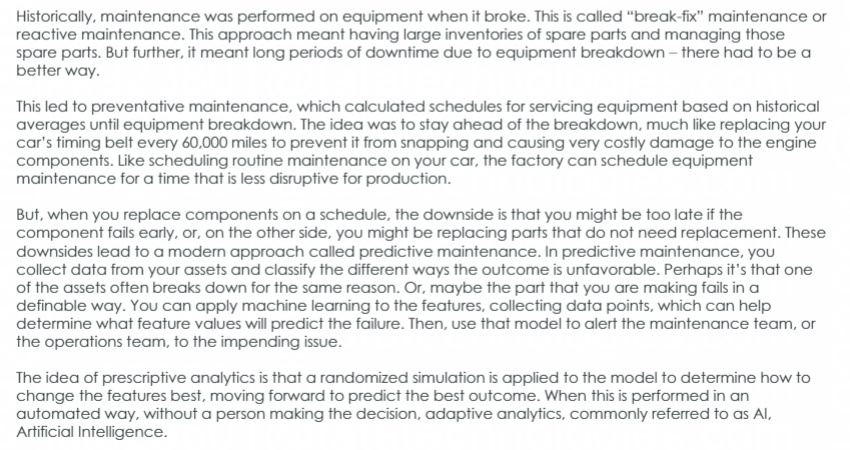
else {

me.DataChange1 = "Good Condition";

}

StateFormating == Demo

**Analytics**



* **Descriptive Analyses**:- used to better understand the data in your dataset.
* **Prediction Model Generation**:- 1.ThingWorx Analytics ingests and analyzes data from connected devices.

2. involves both training the model and validating it.

* **ThingPredictor**:- provides the predictive scoring(to make predictions about future outcomes) capabilities of ThingWorx Analytics.
* **ThingOptimizer**:- provides the prescriptive scoring(to see how certain changes might affect future outcomes.) and optimization capabilities of ThingWorx Analytics.
* **ThingWatcher**:- 1. is a stand-alone tool that can be deployed wherever anomaly detection functionality is required.

2.it observes the data signal from a device, learns what the expected data stream looks like, and then monitors for data points that fall outside of the expected range.

Signals indicate the predictive strength or weakness of specific features on the goal variable. Use **Signals** to explore which features are important to predicting outcomes, and which are not.

Profiles help to find patterns and scenarios within your data and provide performance information based on a selected goal.

**Mashups**

Menu

Media

Style Definition

State Definition

<https://precisionlms.ptc.com/viewer/course/en/31033494/page/31033502>