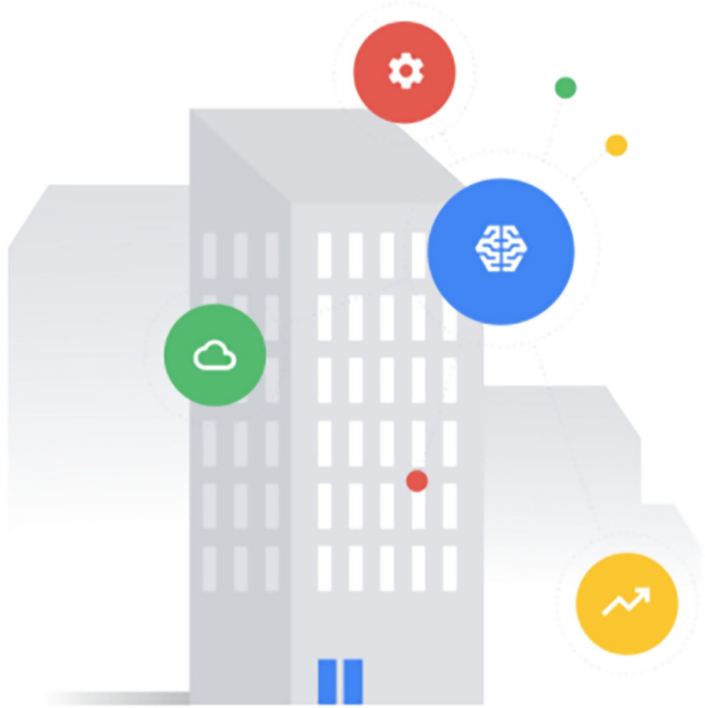
Google

Lesson 1

UDMI overview



Before you get started

This onboarding deck has interactive features and activities that enable a self-guided learning experience. To help you get started, here are two tips for viewing and navigating through the deck.

1 View this deck in presentation mode.

- To enter presentation mode, you can either:
  - Click the **Present** or **Slideshow** button in the top-right corner of this page.
  - Press **Ctrl+F5** (Windows), **Cmd+Enter** (macOS), or **Ctrl+Search+5** (Chrome OS) on your keyboard.
- To exit presentation mode, press the **Esc** key on your keyboard.

2 Navigate by clicking the buttons and links.

- Click the **Back** or **Next** buttons to go backward or forward in the deck. Moving forward, you'll find them in the bottom corners of every slide.
- Click blue text to go to another slide in this deck or open a new page in your browser.
- For the best learning experience, using your keyboard or mouse wheel to navigate is discouraged.

Ready to get started? Let's go!

Lesson 1

UDMI overview

What you'll learn about:

- Technical UDMI principles
- Message transport and encoding
- Types of device messages
- Message subblocks
- UDMI architecture

By the end of this lesson, you'll be able to

- Describe the relationship between UDMI and MQTT.
- Identify the importance of the site model and how each UDMI tool uses it.
- Download the necessary UDMI tools.
- Understand the role of IoT Gateways in the UDMI ecosystem.
- Describe how sequences are used to verify a device's behavior.

Back

Next

Google

UDMI message overview

The UDMI message overview defines some basic conventions and specifications for how messages flow through the system.

The core concept builds on top of a number of different standard technologies and protocols, with a few key additions specific to device management. UDMI itself stands for:

Universal: Applies to a wide range of IoT systems such as HVAC or lighting control.

Device: Oriented around device communications with the cloud.

Management: Primarily for system management and operation, not high-volume data transfer.

Interface: Describes the boundary between the device and cloud, not a specific implementation.

The key points for understanding and assembling a complete system based on UDMI are about a high-level understanding of the message structure, and then the systems and tools that can communicate and validate a UDMI-compliant device.

Back

Next

UDMI message basics

There are 3 types of UDMI messages:

- *State*, which are sent from a device to the cloud
- *Event*, of which there are several kinds (grouped into "sub-blocks") and are sent from a device to the cloud
- *Config*, which are sent from the cloud to a device

The UDMI message schema definition defines how UDMI messages should be structured and represented as a JSON payload (a standard syntax for encoding structured data, rather than just free-form strings).

Underlying data is also organized by "sub-blocks", which provide semantic groupings of data as they apply to various underlying subsystems (e.g., "system" for basic system operation, or "pointset" for data point management).

Back

Note: Lesson 2 will focus on cloud setup.

Next

UDMI message basics (continued)

State messages

Remember, state messages are sent *by* a device.

The device state in the following example was sent from the device to the cloud.

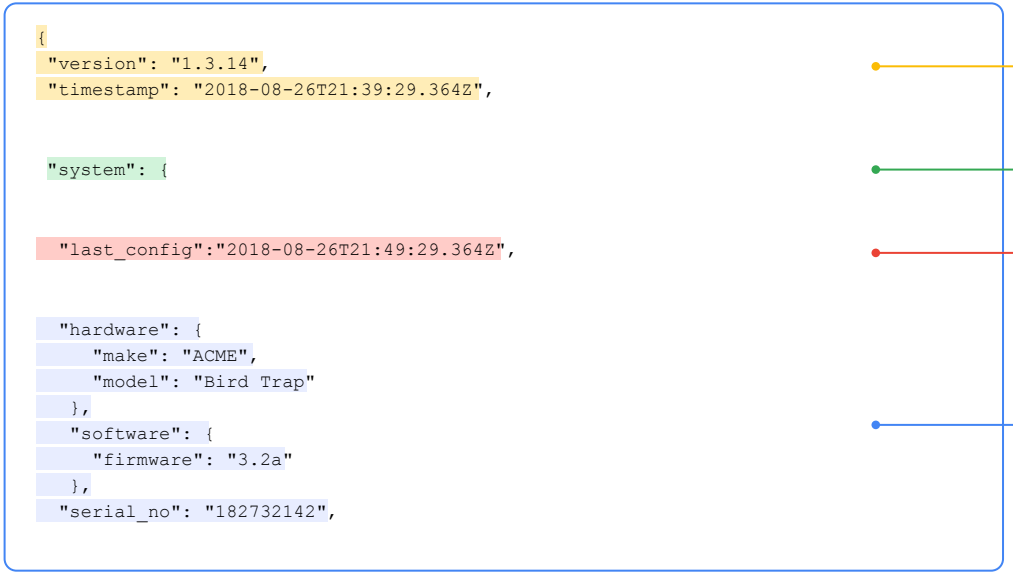
There is one current state per device, which is considered the current device state until a new state message is sent. If a device sends multiple state messages, only the most recently received is considered “valid” and all others may not be processed by the system.

For example, if the state displays an error, it’s assumed that the error still exists until the device sends a new state, which clears out the previous error. It consists of several subsections (e.g., system or pointset) that describe the relevant sub-state components.

Let’s take a look at an example of a device state message in the following slides.

UDMI message basics (continued)

State messages



Standard UDMI header found in all messages

System sub-block containing information about the device

Timestamp of the last config received and accepted by the device

Information about the physical hardware and software (e.g., firmware, operating systems) running on the device and serial number

Back

Next

UDMI message basics (continued)

State messages

```
  "operational": true,
  "status": {
     "message": "Tickity Boo",
     "category": "device.state.com",
     "timestamp": "2018-08-26T21:39:30.364Z",
     "level": 600
    }
  },
"pointset": {
   "status": {  // Status scoped to overall pointset operation
     "message": "Invalid sample time",
     "category": "pointset.config",
     "timestamp": "2018-08-26T21:39:28.364Z",
     "level": 500
   },
   "points": {
     "return_air_temperature_sensor": {
       "status": {  // Status scoped to a specific point in a pointset
         "message": "Point return_air_temperature_sensor unable to read value",
         "category": "pointset.points.telemetry",
         "timestamp": "2018-08-26T21:39:28.364Z",
         "level": 500
       }
     },
     "zone_temperature_setpoint": {
     }
   }
  }
}
```

System operational and status information

System operational and status information

Back

Next

UDMI message basics (continued)

Event messages

Remember, event messages are sent *by* a device, and unlike state and config messages, are scoped to a specific sub-block.

- System events
  System events are primarily used for things like logging, general status, errors, firmware management, specific events (e.g., device startup), and device metrics (e.g., CPU/memory usage).

Let's take a look at an example of an events system message in the next slide.

Back

Next

Google

UDMI message basics (continued)

Event messages

System events


```
{
 "version": "1.3.14",
 "timestamp": "2018-08-26T21:39:29.364Z",
 "logentries": [
   {
     "message": "Configuration received",
     "detail": "Message ID 356633457687432",
     "timestamp": "2018-08-26T21:39:19.364Z",
     "category": "system.config.receive",
     "level": 300
   }
 ]
}
```


The system log entry information

Back

Next

UDMI message basics (continued)

Event messages

Remember, event messages are sent *by* a device.

- Pointset events
  A basic pointset telemetry message contains the point data sent from a device.
  The pointset telemetry message for a device should contain the values for all representative points from that device.

  For example, if the device is a temperature sensor, the pointset event would include the temperature data point.

Let's take a look at an example of an events pointset message in the next slide.

Back

Next

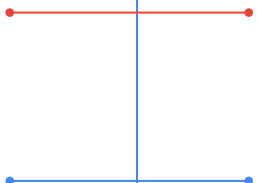
UDMI message basics (continued)

Event messages

Pointset events

```
{
 "version": "1.3.14",
 "timestamp": "2019-01-17T14:02:29.364Z",


 "points": {
   "return air temperature setpoint": {
     "present_value": 21.30108642578125
   },
    "zone temperature setpoint": {
     "present_value": 23
   }
 }
}
```



Standard UDMI header found in all messages

The values of the different points on the device

Back

Next

UDMI message basics (continued)

Config messages

Unlike state and event messages, config messages are sent from the cloud *to* the device.

There is one active config per device, which is considered current until a new config is received.

Config messages describe the desired state of the system. The config block controls the intended behavior of a device. Like state messages, only the most recently received config message is valid, and all other previous ones can be ignored.

Let's take a look at an example of a config message in the following slide.

Back

Next

Google

UDMI message basics (continued)

Config messages

Pointset events

```
{
 "version": "1.3.14",
 "timestamp": "2018-08-26T21:39:29.364Z",

 "system": {
   "metrics_rate_sec": 10,
   "min_loglevel": 400
 },

"pointset": {
   "sample_limit_sec": 2,
   "sample_rate_sec": 500,

"points": {
     "return_air_temperature_sensor": {
     },
     "zone_temperature_setpoint": {
       "set_value": 20
     }
   }
 }
}
```

Configuration timestamp and versioning

System sub-block defining configuration

Pointset subblock defining configuration of the device's points, indicating the expected points, sample rate, cloud-to-device control, etc.

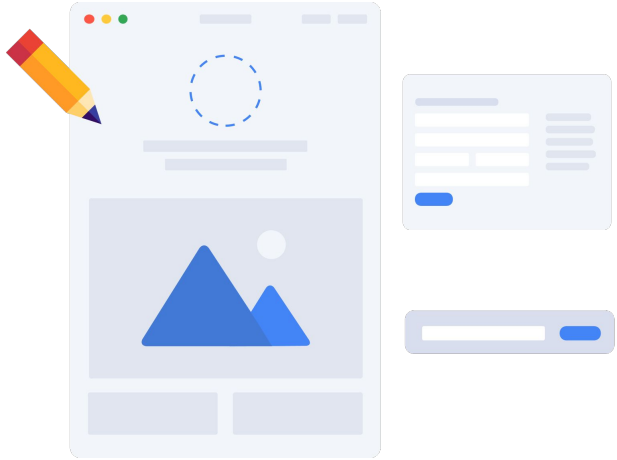
Back

Next

Model basics

UDMI site model

Like a blueprint for a house, it provides the model for what the device should be in the end. The same UDMI site model can be applied to different projects, just like the same blueprint can be used to build multiple copies of a house. A complete site model includes some high-level information about the site overall, along with a specific metadata.json file for each individual device in the site.



Back

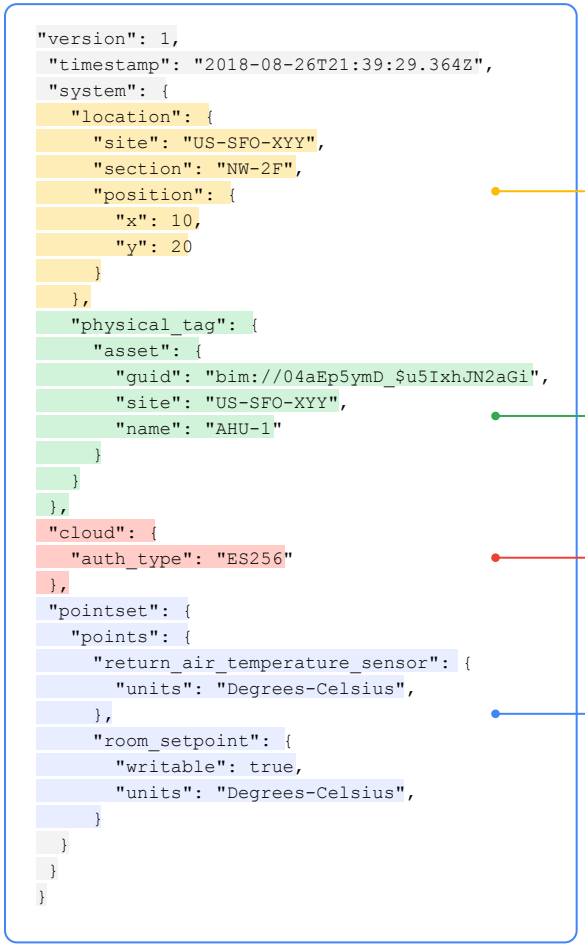
Note: Lesson 3 will focus on UDMI site model and device-to-cloud basics.

Next

Google

Model basics (continued)

Here is an example of the minimum contents of a metadata file.



Back

Next

UDMI tools

Just like a house blueprint, the **site_model** provides information to the various tools that can be used before, during, and after assembling a complete site.

UDMI includes the following tools to support various aspects:

- **keygen** is a script to generate an RSA or ES key for single devices, filling in information that might not be present in a site model.
- **pubber** is a sample implementation of a client-side “device” that implements the UDMI schema, which uses the site_model to determine what information to publish.
- **registrar** is a utility to register and update devices in Cloud IoT Core (GCP) based on the devices listed in the site model.
- **validator** is a utility for validating messages received in a cloud project. The validator is designed to work at-scale and simultaneously validate all messages for a given site.
- **sequencer** is a utility to validate device sequences for a specific device. It's more comprehensive than the validator, but also a bit slower and only works for a single device.

Back

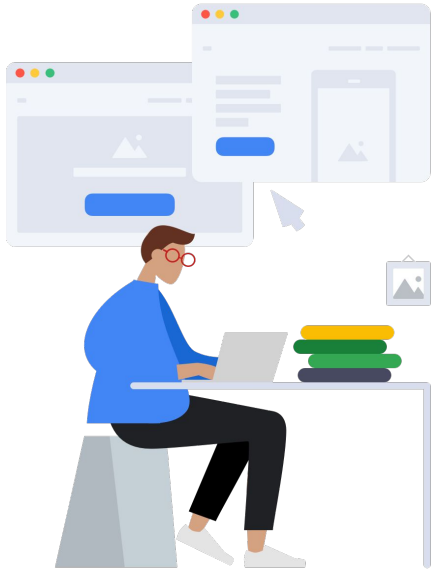
Next

Downloading tools

The UDMI tools must be downloaded. Most of the tool interactions will function with Cloud IoT Core and Pub/Sub.

Before downloading the UDMI tools, you will need the following software installed onto your development system.

- JDK v11
- NPM & Node JS
- Coreutils
- Jq
- Git



Back

Next

Google

Validation

Validation for UDMI is handled by a special tool included in UDMI — the validator.

The validator is designed to run "at scale" for an entire building. It works by observing messages (telemetry, state) being sent from devices to the cloud.

The validator checks:

- If devices are communicating with the cloud
- That messages being sent from devices conform with the UDMI schema
- That messages include all required points, as defined by the site model

Back

Note: Lesson 4 will focus on live programmatic validation.

Next

Downloading tools (continued)

Follow the instructions below to download the UDMI tools.

1. Type the following command into the terminal to download the required software.

```
$ sudo apt-get install default-jdk jq git coreutils
```

2. Next, you will need to install the Google Cloud SDK

   - Google Cloud SDK is required for GCP command-line utilities.
   - For instructions on how to install the Google Cloud SDK, click here.

3. Now you're ready to download the UDMI tools. To download the UDMI repository and tools, clone the git repository in your directory of choice:

```
git clone https://github.com/faucetsdn/udmi.git
```

Back

Next

Gateways

Sometimes a device can't speak to the cloud using MQTT.

Imagine somebody who only speaks French trying to order a coffee in a Thai cafe! The gateway acts as a translator (French-to-Thai) so that the tourist can order the coffee that they want.

The gateway, a directly connected device, proxies the data to the cloud on behalf of the non-MQTT device. From the cloud's perspective, everything is just a "device" and can be interacted with normally. It's the gateway's responsibility to act as the bridge between the two languages.

Gateways themselves are UDMI devices which can directly speak MQTT, just like the Thai native that can always order their own coffee.

The four device types are: reporting, direct, gateway, and proxy.

Back

Note: Lesson 5 will focus on IoT gateways.

Next

Sequences

Sequences, sometimes called event diagrams or event scenarios, depict the objects involved in the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

They define how a particular device should behave and represent a conversation with the device by asking it questions, telling it information, and expecting certain responses.

It's designed like a quiz to see if the device knows what it's talking about. A higher score means that the device is more "fluent" in UDMI and measures the degree to which it conforms to the UDMI specification.

Sequences are invoked on an **individual** device (not the building as a whole), to test specific behaviors.

They are primarily useful by device manufacturers to make sure their device complies with expected standards.

Different subsections define different bits of functionality. For example, a suite of **writeback** tests would probe how a device responds to sequences required to write a data point to a device.

Back

Note: Lesson 6 will focus on sequences.

Next

Lesson 1

Knowledge check

Let’s take a moment to reflect on what you’ve learned so far.

- The next slides will have questions about the concepts that were introduced in this lesson.
- Review each question and select the correct response.

If there are more than two answer options, you won’t be able to move forward until the correct answer is selected.

Back

*Click **Next** when you’re ready to begin.*

Next

Google

Knowledge check 1

This type of UDMI message is primarily used for things like logging, device startup progression, and device metrics (e.g., CPU/memory usage).

Which term is being described?

Select the best answer from the options listed below.

- Pointset events message
- System events message
- Config message
- State message

Back

Next

Google

Knowledge check 1

This type of UDMI message is primarily used for things like logging, device startup progression, and device metrics (e.g., CPU/memory usage).

Which term is being described?

Select the best answer from the options listed below.

- Pointset events message
- System events message
- Config message
- State message

Close... but not quite right! 🤔

Actually, the pointset telemetry message for a device contains the values for all representative points from that device.

Try again

Back

Next

Knowledge check 1

This type of UDMI message is primarily used for things like logging, device startup progression, and device metrics (e.g., CPU/memory usage).

Which term is being described?

Select the best answer from the options listed below.

Pointset events message

System events message

Config message

State message

That's right! 🎉

System event messages are used for logging important system events such as the ones listed in the description.

Back

Next

Knowledge check 1

This type of UDMI message is primarily used for things like logging, device startup progression, and device metrics (e.g., CPU/memory usage).

Which term is being described?

Select the best answer from the options listed below.

Pointset events message

System events message

Config message

State message

Close... but not quite right! 🤔

Actually, config messages describe the desired state of the system.The config block controls a device's intended behavior.

Try again

Back

Next

Google

Knowledge check 1

This type of UDMI message is primarily used for things like logging, device startup progression, and device metrics (e.g., CPU/memory usage).

Which term is being described?

Select the best answer from the options listed below.

Pointset events message

System events message

Config message

State message

Close... but not quite right! 🤔

Actually, state messages consist of several subsections (e.g., system or pointset) that describe the relevant sub-state components.

Try again

Back

Next

Knowledge check 2

This UDMI concept defines how a particular device should behave and represent a conversation with the device by asking it questions, telling it information, and expecting certain responses.

Which term is being described?

Select the best answer from the options listed below.

- Sequences
- Gateways
- Authentication
- Site model

Back

Next

Google

Knowledge check 2

This UDMI concept defines how a particular device should behave and represent a conversation with the device by asking it questions, telling it information, and expecting certain responses.

Which term is being described?

Select the best answer from the options listed below.

- Sequences
- Gateways
- Authentication
- Site model

That's right! 🎉

Sequences are sometimes also called event diagrams or event scenarios. They describe the objects involved in the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

Back

Next

Knowledge check 2

This UDMI concept defines how a particular device should behave and represent a conversation with the device by asking it questions, telling it information, and expecting certain responses.

Which term is being described?

Select the best answer from the options listed below.

Sequences

Gateways

Authentication

Site model

Close... but not quite right! 🤔

Actually, gateways act as a translator in situations where a device can't speak to the cloud using MQTT.

Try again

Back

Next

Google

Knowledge check 2

This UDMI concept defines how a particular device should behave and represent a conversation with the device by asking it questions, telling it information, and expecting certain responses.

Which term is being described?

Select the best answer from the options listed below.

Sequences

Gateways

Authentication

Site model

Close... but not quite right! 🤔

Actually, authentication is a security layer used to protect the system.

Try again

Back

Next

Knowledge check 2

This UDMI concept defines how a particular device should behave and represent a conversation with the device by asking it questions, telling it information, and expecting certain responses.

Which term is being described?

Select the best answer from the options listed below.

Sequences

Gateways

Authentication

Site model

Close... but not quite right! 🤔

Actually, a site model is like a blueprint designed for a UDMI project. It outlines what the project should look like in the end. The same UDMI site_model can be applied to different projects.

Try again

Back

Next

Google

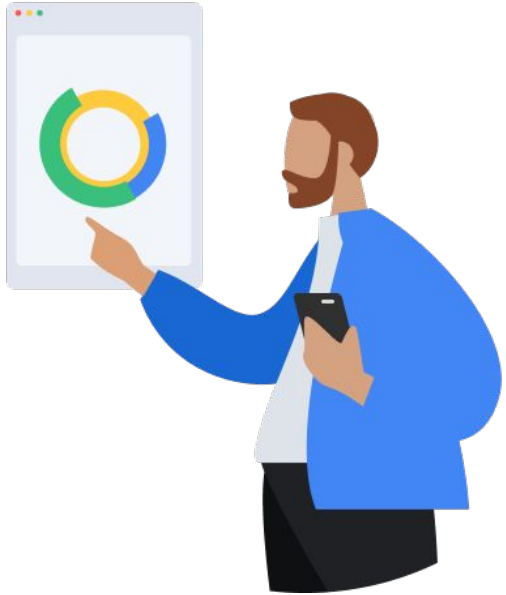
Lesson 1 summary

Let's review what you learned about:

- Technical UDMI principles
- Message transport and encoding
- Types of device messages
- Message subblocks
- UDMI architecture

Now you should be able to:

- Describe the relationship between UDMI and MQTT.
- Identify the importance of the site model and how each UDMI tool uses it.
- Download the necessary UDMI tools.
- Understand the role of IoT Gateways in the UDMI ecosystem.
- Describe how sequences are used to verify a device's behavior.



Back

Next

You completed Lesson 1!

Now's a great time to take a quick break before starting Lesson 2.

Ready for Lesson 2?

Let's go!

Back

*Press the **Esc** key on your keyboard to exit presentation mode.*

Google

Helpful resources

Bookmark these resources for future reference.

- UDMI Project GitHub
  Contains specification for management and operation of IoT systems.
- Git Documentation
  Contains various sources of information about Git contributed by the Git community.