

Mac Admins India



macOS updates via Declarative Device Management
Tech Deep Dive

Shyam Sundar | 09-Mar-24



Agenda

What is DDM?

macOS updates via
DDM

Troubleshooting

DDM EA



Declarative Device Management

- It is a faster and more efficient way of managing macOS devices.
- DDM is built on top of the MDM protocol and provides a more efficient and secure way to manage devices.
- Its Autonomous and Proactive
- Avoids common performance and scaling issues seen with MDM

Declarative data model

Declarations

Status channel

Extensibility

Declarations

- Declarations are the payloads that define policy and desired state on a device.
- Declarations make up the policy an organization wants to define for a device.
- They are serialized as JSON objects and have required properties that allow the policy to synchronize with the management server.
- Declarations come in four flavours:



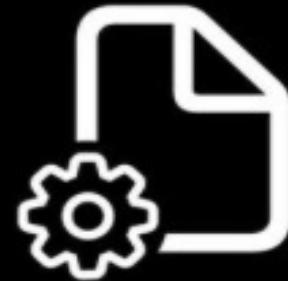
Declaration types



Configurations



Assets



Activations



Management

Configuration

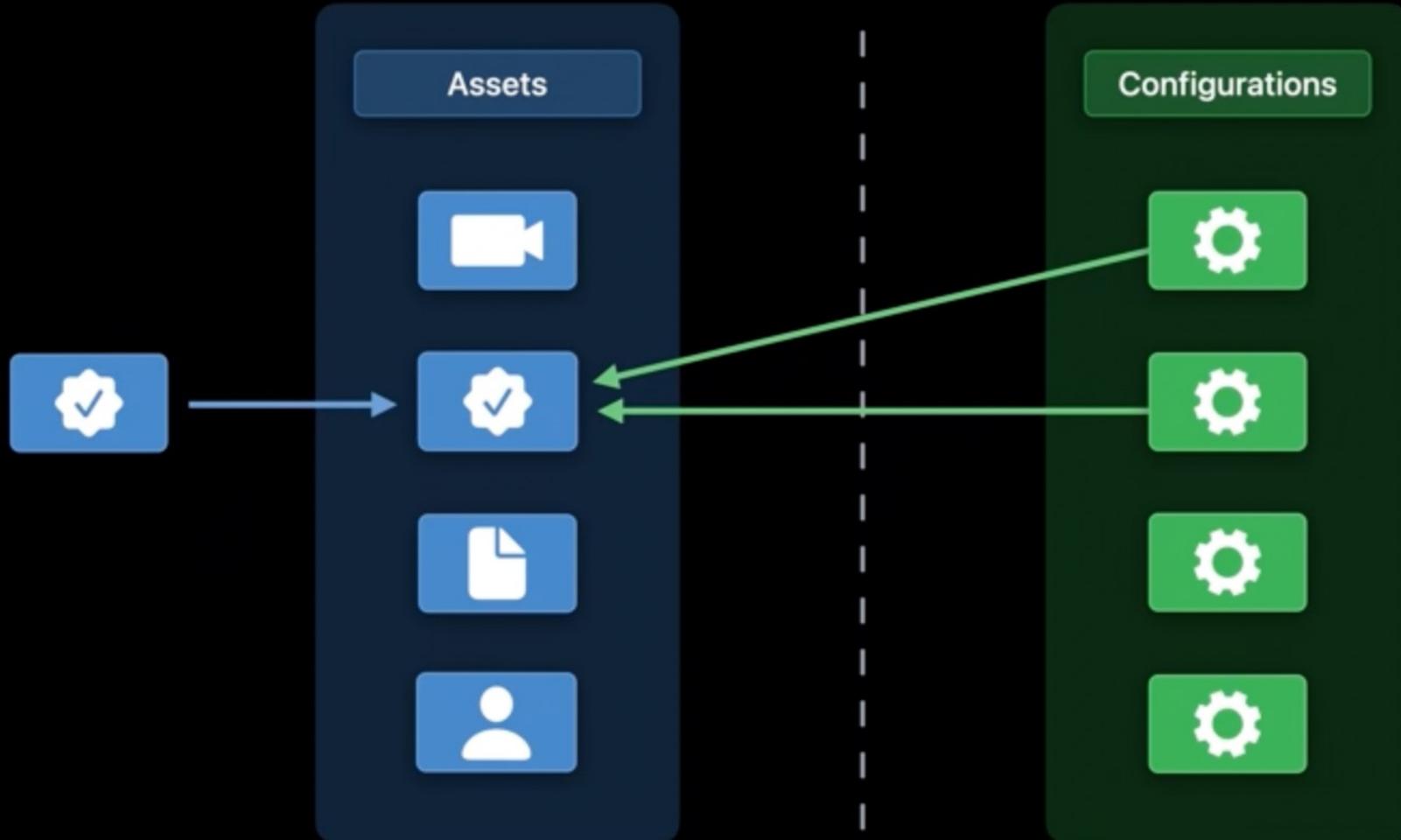
Configuration represents policies being applied to the device.

```
{  
  "Type": "com.apple.configuration.passcode.settings",  
  "Identifier": "4AE28C9F-7082-4521-B435-2550E8B4D57A",  
  "Server Token": "EAB5142C-7E86-4C8E-B576-B7F94D0F6CEA",  
  "Payload": {  
    "RequirePasscode": "true",  
    "RequireComplexPasscode": "true",  
    "MinimumLength": 6,  
    "MaximumFailedAttempts": 10,  
    "MaximumGracePeriodInMinutes": 10,  
    "MaximumInactivityInMinutes": 10,  
    "PasscodeReuseLimit": 0  
  }  
}
```

Assets

- Data needed by a configuration
- Assets have a one-to-many relationship with configurations

```
{  
    "Type": "com.apple.asset.useridentity",  
    "Identifier": "0DEE79A7-3244-42A0-A4CB-46333AC0F63A",  
    "Server Token": "469D704C-A817-4B71-A1B5-B221FDB43E97",  
    "Payload": {  
        "FullName": "First Last",  
        "EmailAddress": "user@example.com"  
    }  
}
```

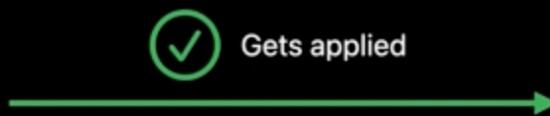
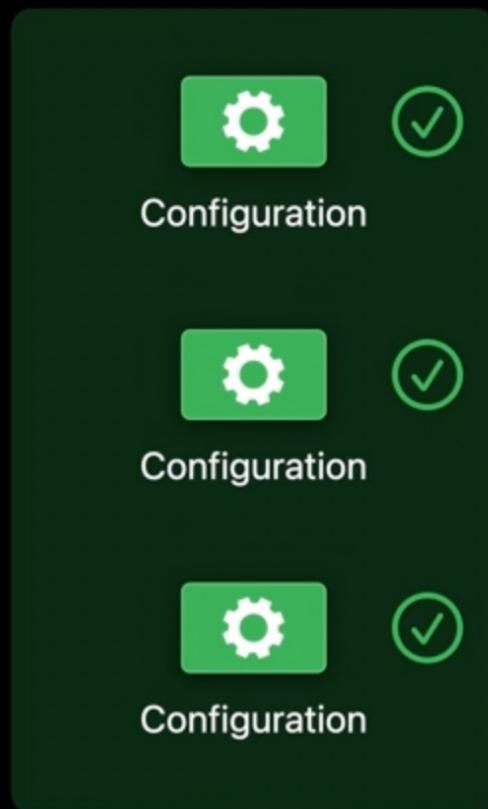


Activations

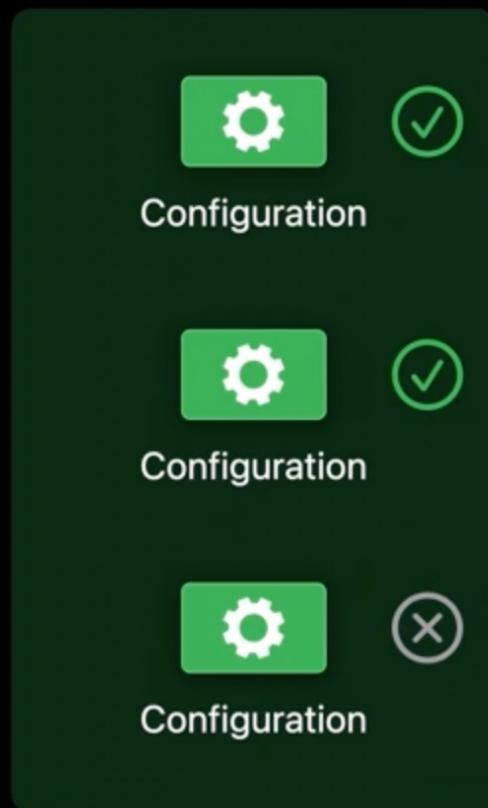
Activations are a set of configurations that are applied atomically to the device and can include predicates

```
{  
    "Type": "com.apple.configuration.passcode.settings",  
    "Identifier": "4AE28C9F-7082-4521-B435-2550E8B4D57A",  
    "Server Token": "EAB5142C-7E86-4C8E-B576-B7F94D0F6CEA",  
    "Payload": {  
        "RequirePasscode": "true",  
        "RequireComplexPasscode": "true",  
        "MinimumLength": 6,  
        "MaximumFailedAttempts": 10,  
        "MaximumGracePeriodInMinutes": 10,  
        "MaximumInactivityInMinutes": 10,  
        "PasscodeReuseLimit": 0  
    }  
}
```

Activations

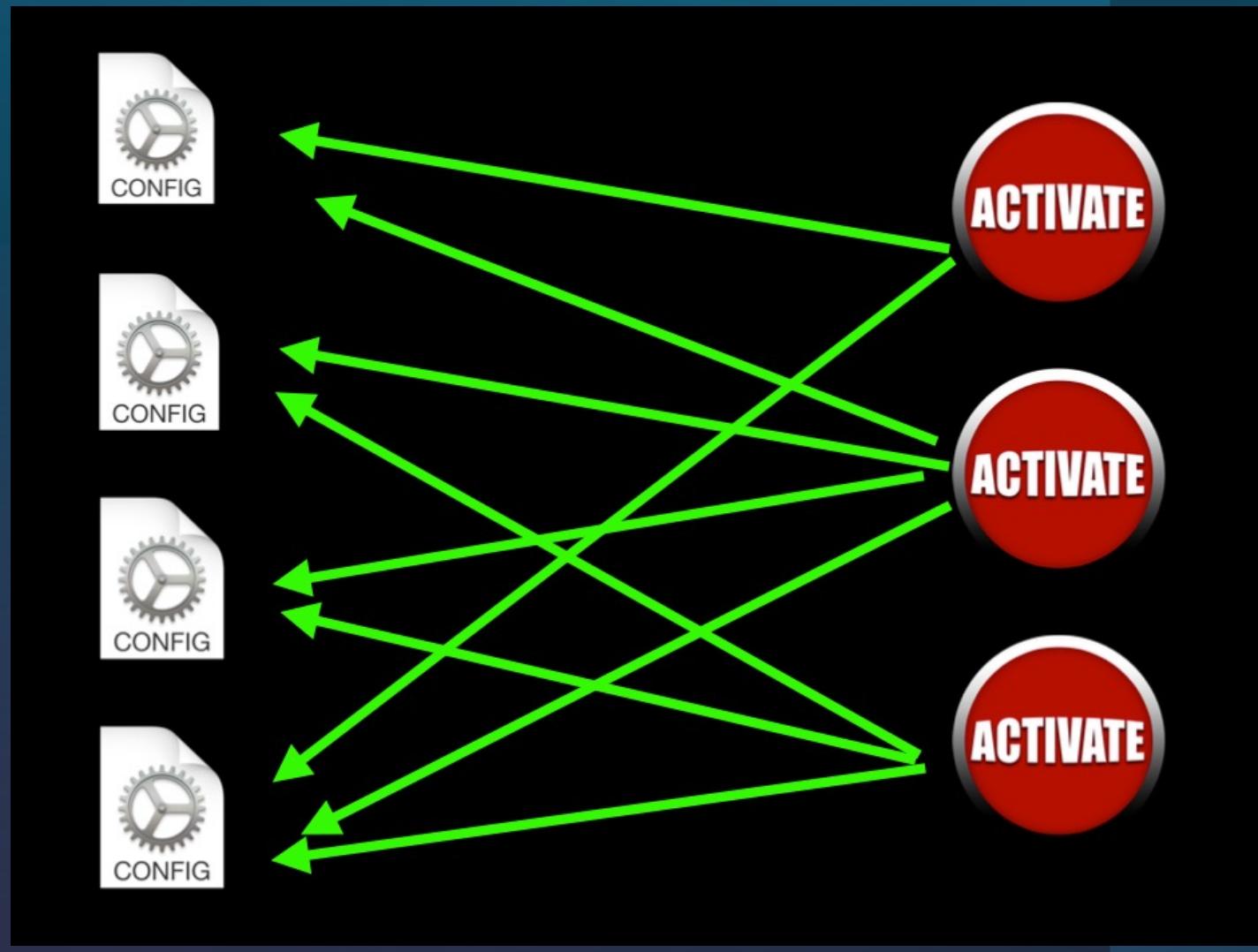


Activations



✖ Not applied





Predicates

```
{  
    "Type": "com.apple.activation.simple",  
    "Identifier": "2AD45A8B-B656-4E51-8406-D97549CDE134",  
    "Server Token": "7BCE079A-73D6-4A3E-8F5B-9B6021F43140",  
    "Payload": {  
        "StandardConfigurations": [  
            "F3FB5A3F-9835-4A9B-9ED0-D2C813E6668E",  
            "EBC2380F-4F8A-4E75-B9AC-F6300AFBD7FE"  
        ]  
    },  
    "Predicate": "(device.model.family == 'iPad')"  
}
```

Activation predicates

Predicate determines activation state

Processed if predicate evaluates to true

For example, `device.model.family == 'iPad'`

Or `device.operating-system.version > '14'`

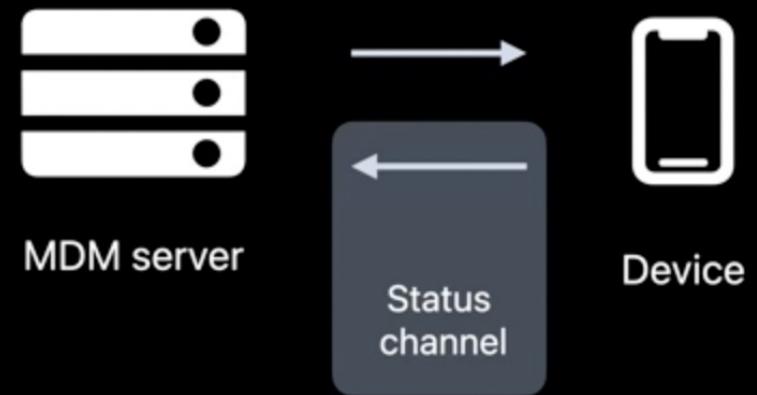
Predicates are automatically re-evaluated

Management

- Represent properties of the overall management state
- Organization information
- MDM Server capabilities
- Conveys static information to the device

Status Channel

- Clients report to server
- Server subscribes to status items
- Status reports are incremental
- Status items are key-paths
 - device.operating-system.family
 - device.operating-system.version
 - device.model.family
 - device.model.identifier



Example: Status subscription configuration

```
{  
    "Type": "com.apple.configuration.management.status-subscriptions",  
    "Identifier": "FA6BACF9-5F66-4261-9E0F-E3A056D9BA12",  
    "ServerToken": "490EF9C9-99B0-4035-ABB9-03876D05894A",  
    "Payload": {  
        "StatusItems": [  
            { "Name": "device.operating-system.version" },  
            { "Name": "device.operating-system.family" },  
            { "Name": "device.model.family" }  
        ]  
    }  
}
```

Example: Initial status report

```
{  
    "StatusItems": {  
        "device": {  
            "model": {  
                "family": "iPhone"  
            },  
            "operating-system": {  
                "version": "14.5",  
                "family": "iOS"  
            }  
        },  
        "Errors": []  
    }  
}
```

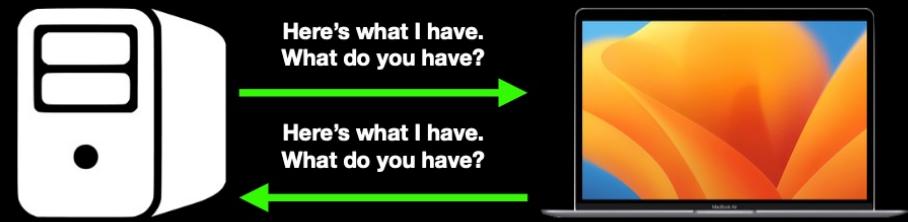
Example: Status report for changed status item

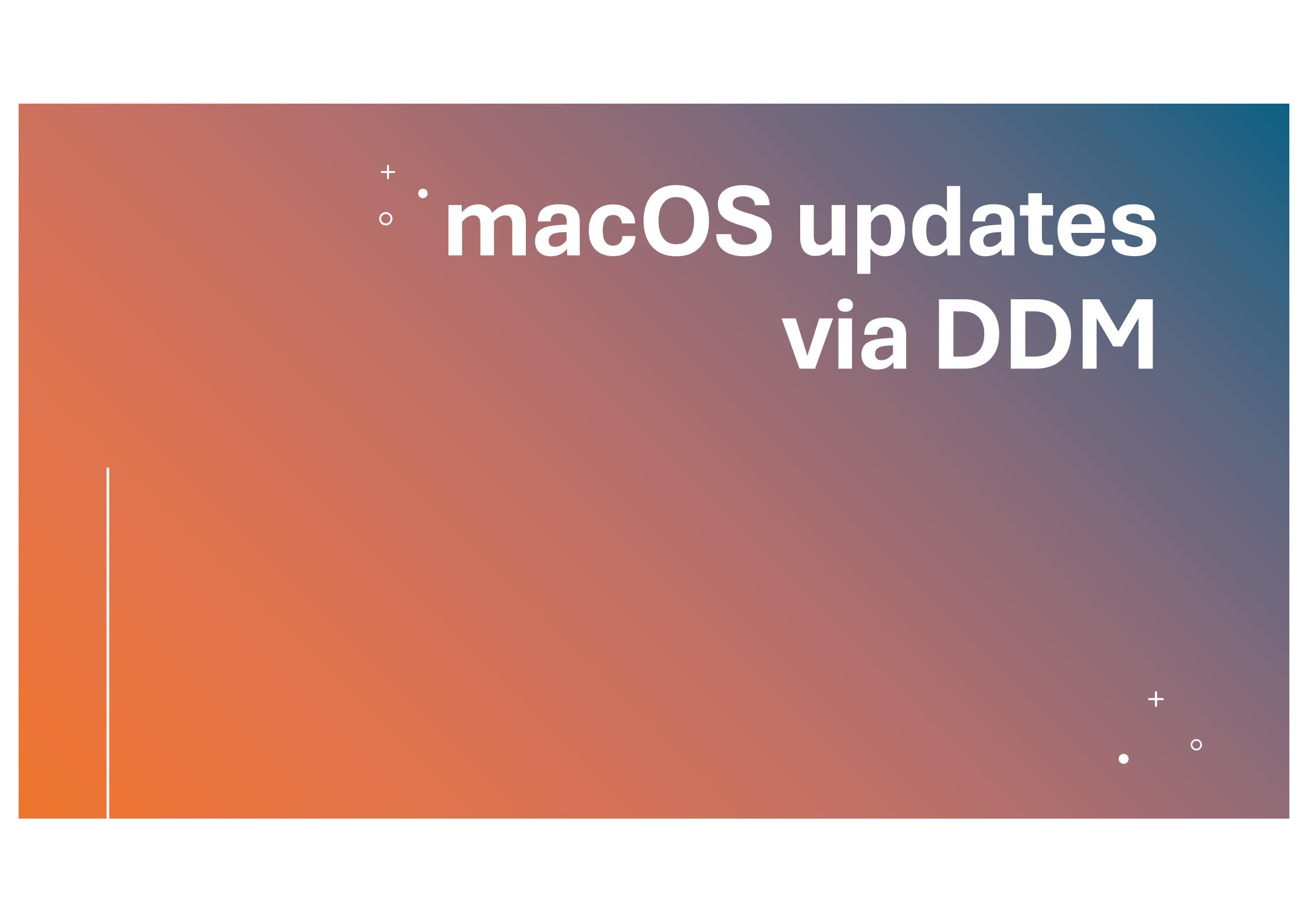
```
{  
  "StatusItems": {  
    "device": {  
      "operating-system": {  
        "version": "15.0"  
      }  
    }  
  },  
  "Errors": []  
}
```

Extensibility

- Compatibility over time
- Device and server advertise what they support
- Supported features
- Supported payloads

Extensibility





macOS updates via DDM

MDM



DDM



Here's the plan.
You got this?

Thanks, I got this.
Check in with you later.



[Computers](#)[Inventory](#)[Search Inventory](#)
[Search Volume Content](#)
[Licensed Software](#)[Content Management](#)
[Policies](#)
[Configuration Profiles](#)
[Software Updates](#)[Restricted Software](#)[Mac Apps](#)[Patch Management](#)[eBooks](#)[Groups](#)[Smart Computer Groups](#)[Static Computer Groups](#)

Computers

Software Updates

 Use new experience **BETA**[Computer Group](#) [Mobile Device Group](#)[←](#) **1** [→](#) 1 - 4 of 4

GROUP NAME



TYPE

<input type="checkbox"/> All Managed Clients	Smart Group
<input type="checkbox"/> All Managed Servers	Smart Group
<input type="checkbox"/> asdfasdf	Smart Group
<input type="checkbox"/> FileVault Encrypted	Smart Group

Dashboard

Computers >

Devices >

Users >

Settings

Computers : Software Updates

← Update 1 group

 Automatically download OS update on the selected device Download and install
Automatically download and install OS update on the selected device Download and schedule to install
Automatically download and schedule to install OS update on the selected device

Force install date is only supported for computers with macOS 14 or later and mobile devices with iOS 17 or iPadOS 17 or later. Computers and mobile devices not meeting these criteria will not receive an update.

Force Install Date

Required

 Setting a date and time prior to the present date and time will immediately enforce the software update formula. Download, install, and allow deferral
Automatically download OS update on the selected device and allow user to defer installation Download, install, and restart
Automatically update OS and force restart on the selected device

Target Version

Choose version to update all selected devices to

 Latest version based on device eligibility Latest major version Latest minor version Specific version

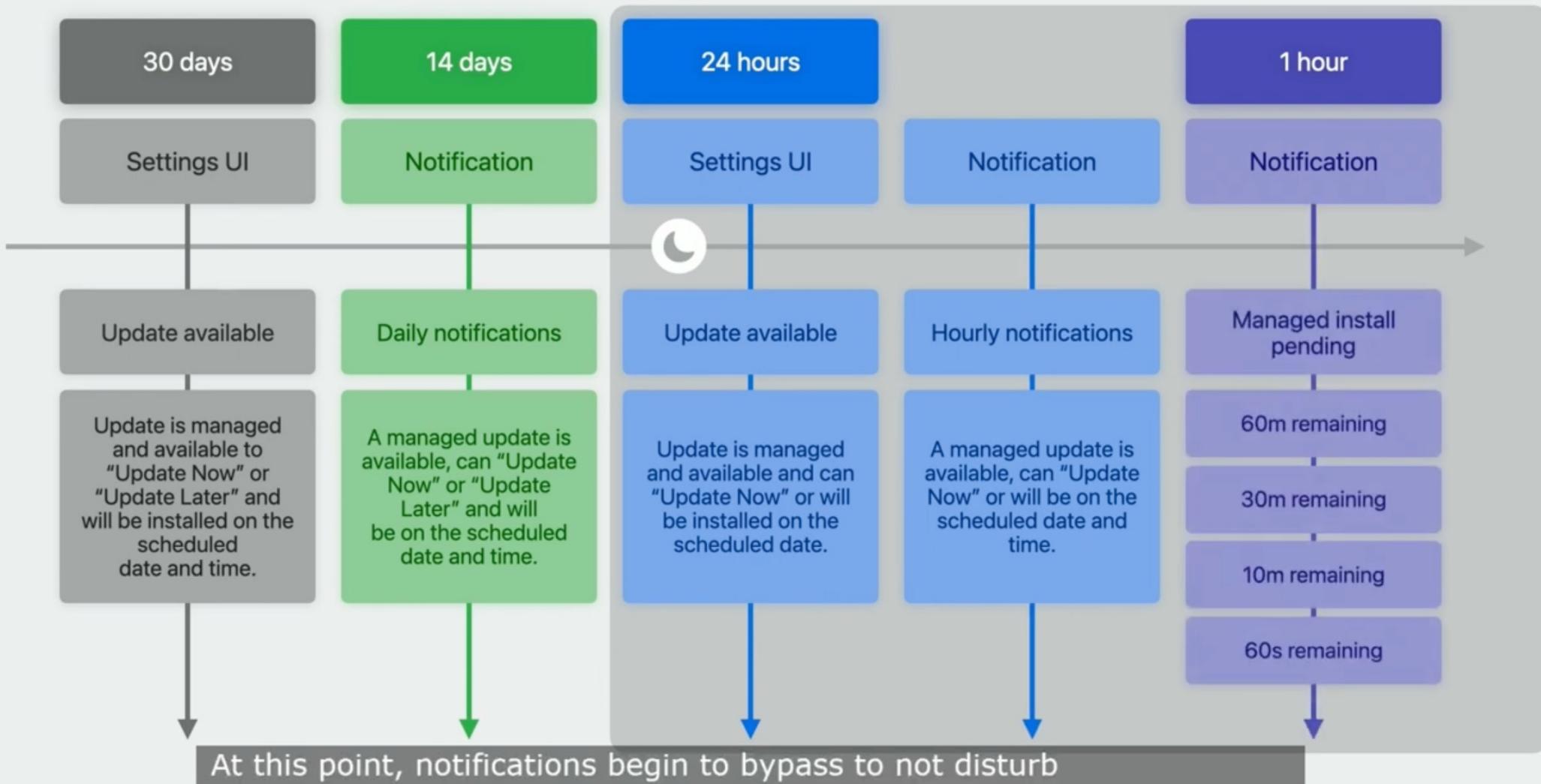
Cancel

Apply

Update available notification



Software update enforce notification experience



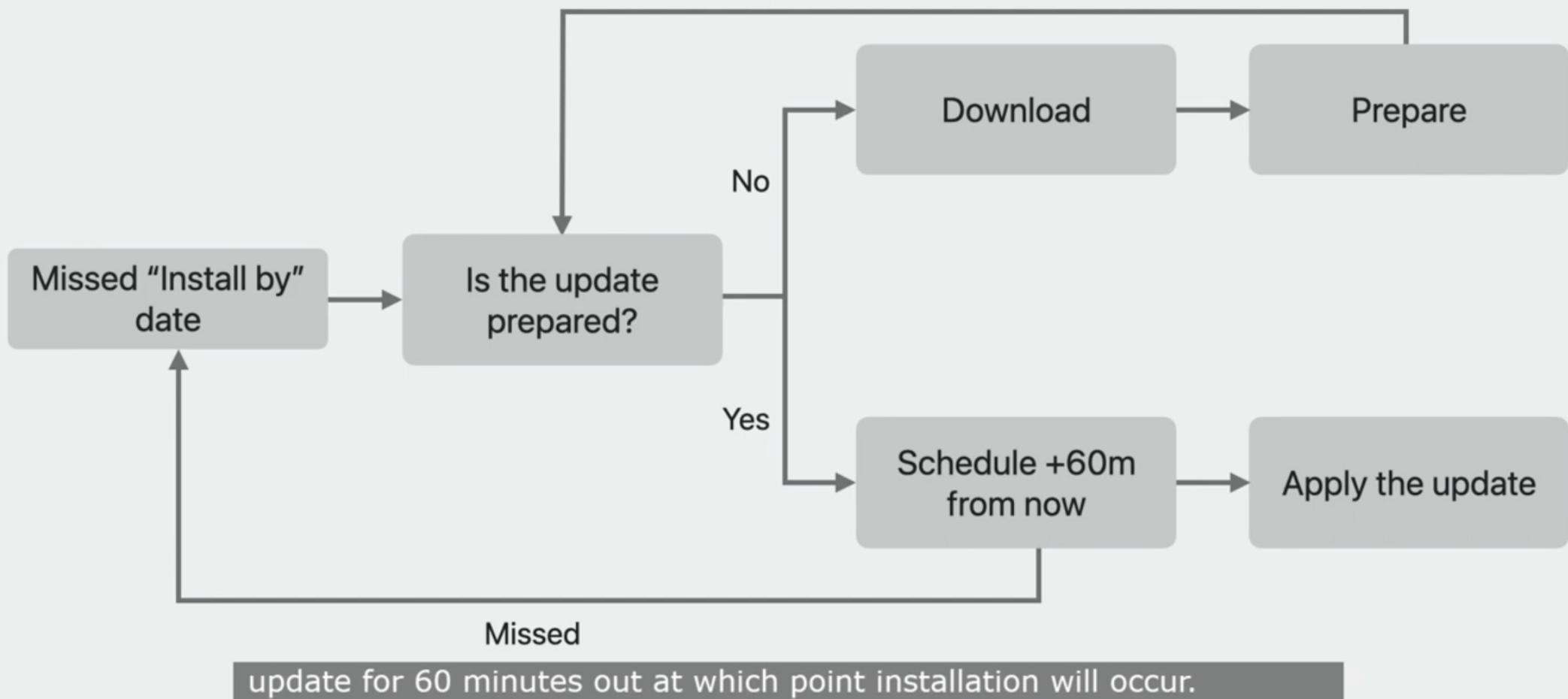
Update past-due notification

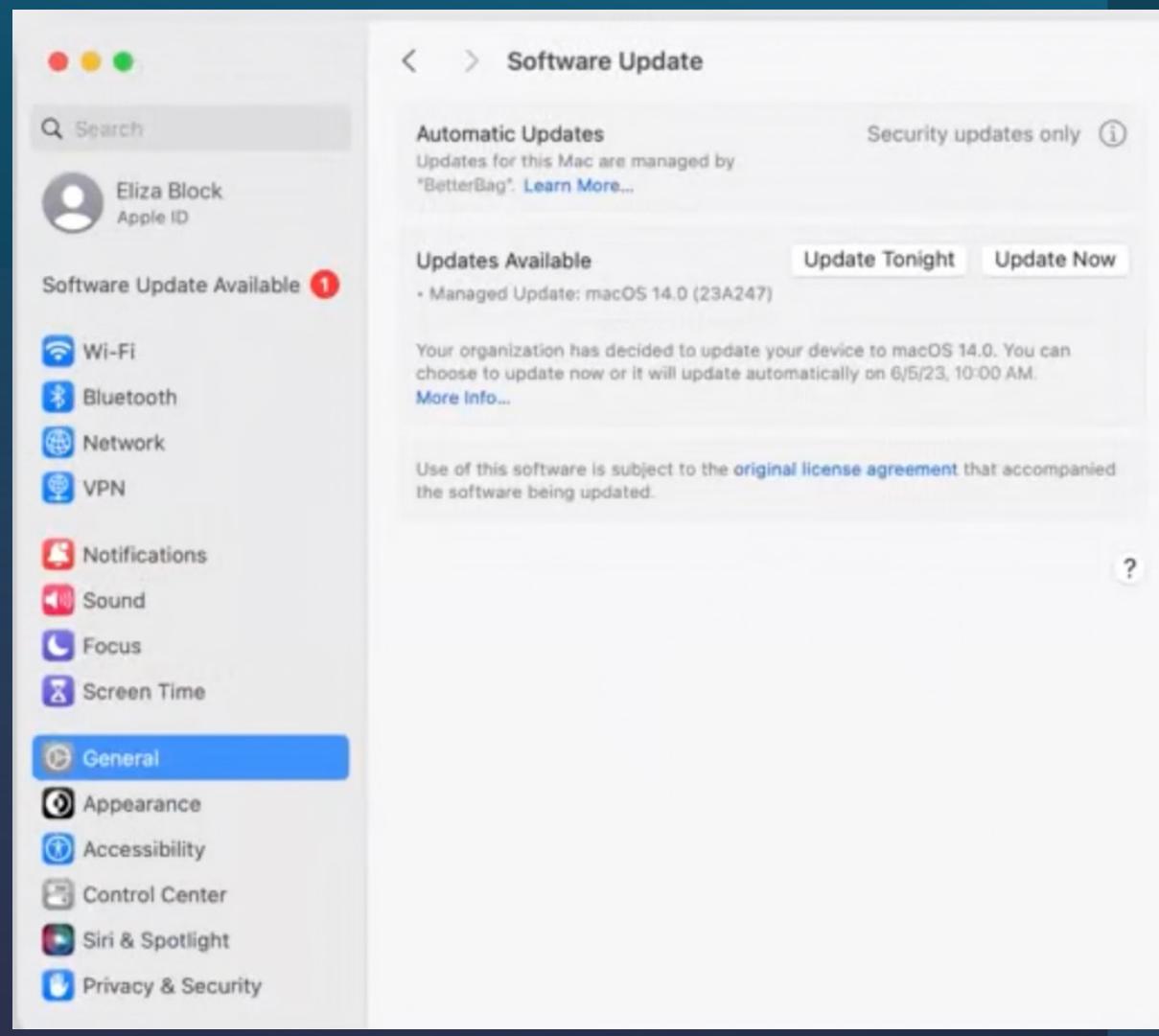


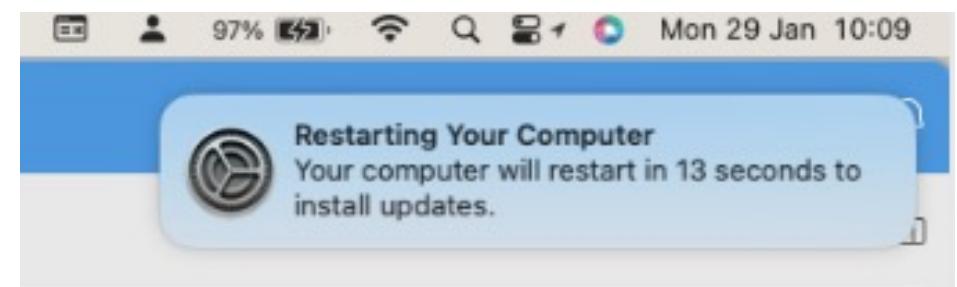
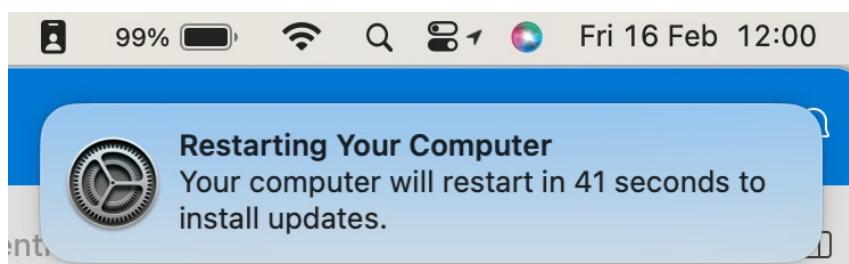
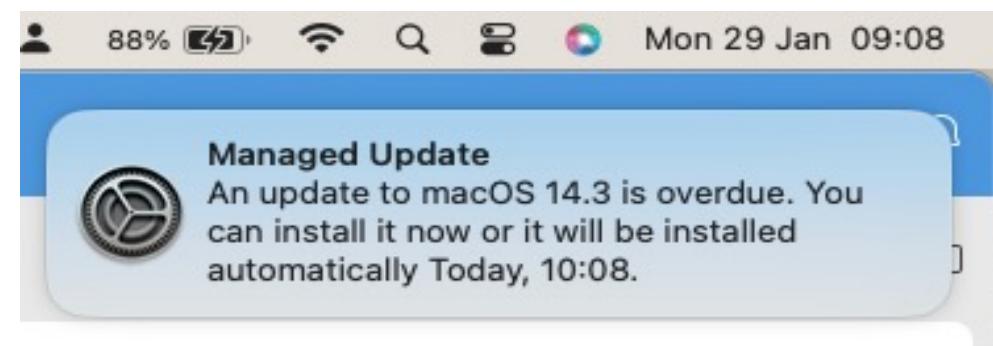
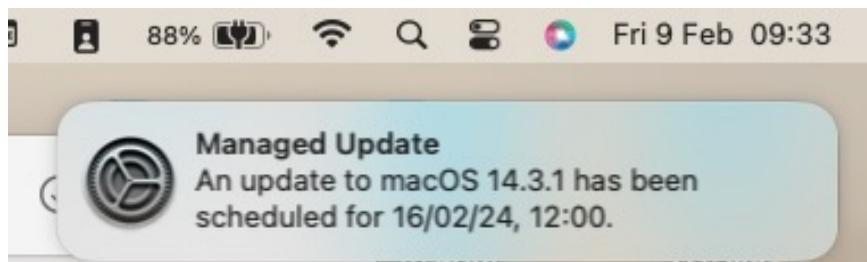
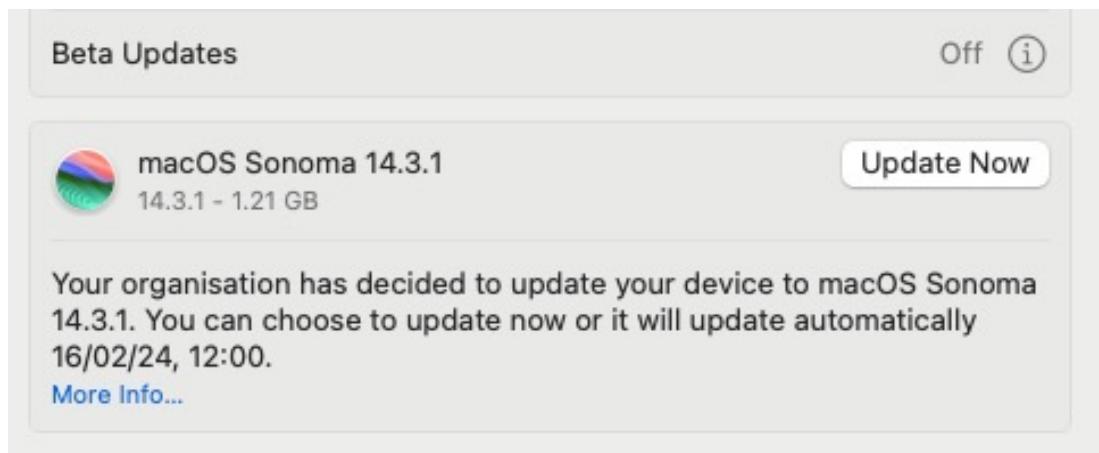
Managed Update

An update to macOS 14.0 is past due. You can install it now or it will be installed automatically within the next hour.

Missed “specific enforcement” deadline









• Troubleshooting



DDM Local Status for macOS

- This .plist is created when a DDM declaration for a software update hits your Mac.
- `/var/db/softwareupdate/SoftwareUpdateDDMStatePersistence.plist`
- The contents change based on whether there's an update available for your Mac or not

Without update

```
{  
    SUCorePersistedStateContentsType = SoftwareUpdateCorePersistedStateFile;  
    SUCorePersistedStateCoreFields = {  
    };  
    SUCorePersistedStateCoreSecureCodedObjectsFields = {  
    };  
    SUCorePersistedStateCoreVersion = "2.1.0";  
    SUCorePersistedStatePolicyFields = {  
        Declarations = {  
        };  
    };  
    SUCorePersistedStatePolicySecureCodedObjectsFields = {  
    };  
    SUCorePersistedStatePolicyVersion = "1.0";  
}
```

With update

```
{  
    SUCorePersistedStateContentsType = SoftwareUpdateCorePersistedStateFile;  
    SUCorePersistedStateCoreFields = {  
    };  
    SUCorePersistedStateCoreSecureCodedObjectsFields = {  
    };  
    SUCorePersistedStateCoreVersion = "2.1.0";  
    SUCorePersistedStatePolicyFields = {  
        Declarations = {  
            "com.apple.RemoteManagement.SoftwareUpdateExtension/ABCDEFG123-1234-ABCD-5678-  
A1B2C3D4E5:abcdeFghiJKLMNOpqrsUVwxyz1234567890.abcdeFghiJKLMNOpqrsUVwxyz1234567890=" {  
                RMStoreDeclarationKey = "com.apple.RemoteManagement.SoftwareUpdateExtension/ABCDEFG123-1234-ABCD-5678-  
A1B2C3D4E5:abcdeFghiJKLMNOpqrsUVwxyz1234567890.abcdeFghiJKLMNOpqrsUVwxyz1234567890=";  
                TargetLocalDateTime = "2023-11-14T00:00:00";  
                TargetOSVersion = "14.1.1";  
            };  
        };  
    };  
    SUCorePersistedStatePolicySecureCodedObjectsFields = {  
    };  
    SUCorePersistedStatePolicyVersion = "1.0";  
}
```

DDM Software Update Notification

- DDM Logs are stored under install.log (/var/log/install.log)
- Search for “Posted DDM notification with success: YES” to get all notification
- Search for “forced=1” to get the details of forced restart time.

Extended Attributes

+
o
•

+

•

o

EA

- defaults read

```
/var/db/softwareupdate/SoftwareUpdateDDMStatePersistence.plist |  
grep TargetLocalDateTime | awk -F '"' '/1/ {print $2}' | sed 's/T/, /'
```

- defaults read

```
/var/db/softwareupdate/SoftwareUpdateDDMStatePersistence.plist |  
grep TargetOSVersion | awk -F '"' '/1/ {print $2}'
```

- grep "SUOSUManagedServiceDaemon: Posted DDM notification
with success: YES" /var/log/install.log | awk '{print \$1, \$2}'

- grep "forced=1" /var/log/install.log | awk '{print \$1, \$2}'

References

- <https://developer.apple.com/videos/play/wwdc2022/10046/>
- <https://it-training.apple.com/tutorials/deployment/dm005>
- <https://developer.apple.com/wwdc21/10131>
- <https://www.youtube.com/watch?v=MvCM82bh6D0>
- <https://www.jamf.com/blog/getting-to-know-declarative-management/>
- <https://developer.apple.com/videos/play/wwdc2023/10041/>
- <https://www.youtube.com/watch?v=XZwzKhS7H08&t=1395s>
- <https://www.sudoade.com/ddm-software-update-logs/>



Thank you