

# Detecting the Display Model using a Custom Field

Use a FileWave Custom Field to store the model of display connected to a Mac or Windows system.

## Adding the Custom Field

1. Download the following Custom Field export - [Display Model Custom Field.customfields](#)
2. Import the downloaded file into "FileWave Admin>Assistants>Custom Fields>Edit Custom Fields>Import".
3. Save changes within Custom Fields dialog.
4. Associate Custom Field with desired Windows devices via "right-click>Edit Custom Field(s) Associations" or check the "Assigned to all devices" checkbox when editing the Custom Field.
  - A Smart Group is very helpful to quickly associate Custom Field

The screenshot shows the 'Custom Fields' dialog in FileWave Admin. On the left, a list of custom fields is shown, with 'Display Model' selected. The right pane shows the 'Field Details' for 'Display Model'. The 'Name' is 'Display Model' and the 'Internal Name' is 'display\_model'. The 'Description' is empty. The 'Provided By' is 'Client Script'. The 'Values' section shows 'Data Type' as 'String', 'Restrict allowed values' is unchecked, and 'Use a default value' is checked. The 'Client Script' section is active, showing a PowerShell script for detecting the display model. The script type is 'PowerShell' and the execution environment is 'Windows'.

Display Name	Internal Name
BitLocker Key	bitlocker_key
Client Config Booster Routing	client_config_booster_routing
Client Config Booster1	client_config_booster1
Client Config Booster2	client_config_booster2
Client Config Booster3	client_config_booster3
Client Config Booster4	client_config_booster4
Client Config Booster5	client_config_booster5
Client Config Debug Level	client_config_debug_level
Client Config Hashed Password	client_config_hashed_password
Client Config Server Address	client_config_server_address
Client Config Tickle Interval	client_config_tickle_interval
<b>Display Model</b>	<b>display_model</b>
Max Battery Capacity	max_battery_capacity
Missing OS Update Count	missing_os_update_count
PowerShell x64	powershell_x64

**Field Details**

**Name**  
Display Model

**Internal Name**  
Using internal name the field can be referenced in other parts of FileWave  
display\_model

**Description**

**Provided By**  
Defines how the field value shall be populated  
Client Script

☒ Assigned to all devices

**Values**

**Data Type**  
String

☐ Restrict allowed values  
☒ Use a default value

**Pending...**

**Client Script**  
This script will be run on the client side on verification. The output of the script will be captured and will serve as the value for the field. The default value will be assigned until the script is executed. If the script fails during client association, the default value will be used.

Script type: PowerShell

Execution Environment: macOS | Windows

```
$MonitorList = Get-WmiObject -Class WmiMonitorID -Namespace "ROOT\WMI"
$MonitorOutput = @()

foreach ($monitor in $MonitorList) {
    $mon = @{}
    $manufacturer = $null
    $name = $null

    foreach($sch in $monitor.ManufacturerName) {
        if($sch -ne '00') {
            $manufacturer += [char]$sch
        }
    }

    foreach($sch in $monitor.UserFriendlyName) {
        if($sch -ne '00') {
            $name += [char]$sch
        }
    }
}
```

- Copy/paste the following PowerShell script into the scripting box under the Windows section:

```
$MonitorList = Get-WmiObject -Class WmiMonitorID -Namespace "ROOT\WMI"
$MonitorOutput = @()

foreach ($monitor in $MonitorList) {
    $mon = @{}
    $manufacturer = $null
    $name = $null

    foreach($sch in $monitor.ManufacturerName) {
        if($sch -ne '00') {
            $manufacturer += [char]$sch
        }
    }

    foreach($sch in $monitor.UserFriendlyName) {
        if($sch -ne '00') {
            $name += [char]$sch
        }
    }
}
```

```
$mon = $manufacturer + " " + $name
$MonitorOutput += $mon
}
$MonitorOutput=$MonitorOutput -join ', '
$MonitorOutput
exit 0
```

- Copy/paste the following Shell script into the scripting box under the macOS section:

```
#!/bin/bash
DisplayModel=`system_profiler SPDisplaysDataType | grep "Resolution:" -B1 | awk -v n=3 'NR%n==1' | sed "s/^[
\t]*/" | sed 's/:/,/g' | tr '\n' ' ' `
echo ${DisplayModel}
exit 0
```

- Save changes within Custom Fields dialog.
- Associate Custom Field with desired devices via "right-click>Edit Custom Field(s) Associations" or check the "Assigned to all devices" checkbox when editing the Custom Field.
  - A Smart Group can be helpful to quickly associate Custom Field.
- Alternatively you can also simply assign the field to all devices.

## Results

Dashboard	Search: Everything Clients Mobile Groups <span>⋮</span> Clear all filters			
	Name	ID	Comment	Display Model
Filesets	▼  Display Model Demo	6744		
Associations	DECKER	243		HWP HP 23xi
Imaging	JOSH-CRYPT	242		GSM LG HDR 4K
iOS Inventory	macOS BigSur Lab 1	344		Apple Virtual,
	Win10-Lab1	335		PRL Parallels Vu

## Related articles

- [Custom Fields](#)

🕒 Revision #2

★ Created 13 July 2023 21:20:50 by Josh Levitsky

✎ Updated 14 July 2023 03:36:11 by Josh Levitsky