BrandM3dia
Kiosk Deployment & Operations Manual

Field Execution Guide (Task-Based) + Technical Appendix
Zero-Experience Edition

Version: v2.0-UX   Last updated: 2025-12-23

How to Use This Manual

This manual is designed for field use in high-stress environments. It uses Progressive Disclosure:

- Field Execution Guide (Tasks): short steps, checklists, pass/fail tables, and red-flag stop conditions.
- Technical Appendix: deeper procedures, scripts, and terminal commands (use only when needed).

NOTE: Golden Rule (Digital Twin)

If the kiosk is not registered in asset.bm3group.com, it does not exist.

No registration = no deployment.

CAUTION: Do not skip steps

Follow tasks in order. Skipping steps usually causes failures later (sleep/lock, kiosk exits to desktop, offline site breaks, missing evidence).

Audience: Junior IT technician (no kiosk/Linux experience required).

Goal: If you follow this manual, you can provision, deploy, support, and pack down a kiosk without supervision.

1. Document Control

1.1 Versioning

Version	Date (YYYY-MM-DD)	Author	Change summary
v2.0-UX	2025-12-23	IT Ops	Modular redesign: task-based field guide + technical appendix

1.2 Source of Truth (Links)

Use these links only. Do not create side copies. Replace ADD LINK placeholders per client/event.

- Basecamp Project Root (client/event): ADD LINK
- OneDrive Main Folder (client/event): ADD LINK
- SOP Library Index (company-wide): https://3.basecamp.com/4938325/buckets/20576475/vaults/9409145986
- Kiosk Model Hardware Docs (specs, wiring, drivers): https://3.basecamp.com/4938325/buckets/20576475/card_tables/columns/9116848125
- Scripts Repo (kiosk engine, health scripts): https://3.basecamp.com/4938325/buckets/20576475/vaults/9409146592
- OS Imaging / Baseline Setup SOP: https://3.basecamp.com/4938325/buckets/20576475/vaults/9409240123
- Kiosk Mode SOP (online): https://3.basecamp.com/4938325/buckets/20576475/vaults/9409264331
- Kiosk Mode assets / upload: https://3.basecamp.com/4938325/buckets/20576475/uploads/9409277758
- Pre-Rental / Live Event / After-Sales bundle: https://3.basecamp.com/4938325/buckets/20576475/vaults/9409270125

1.3 Roles & Escalation

- PM (Accountable): client communications, approvals, sign-off.
- IT Lead (Reviewer): validates configuration before go-live.
- Field Tech (You): executes tasks, logs proof, reports issues immediately.

2. Quick Definitions (Glossary for Non-IT Staff)

Use this page when you see unfamiliar words.

Term	Plain meaning	Why it matters
Flycase / Flight Case	Hard shipping case the kiosk travels in.	Damage to the case often means damage inside.
Service Panel / Back Terminal	Locked rear door where PC ports and internal cables are.	Needed to connect keyboard/mouse and inspect internal cables.
Provisioning	Preparing kiosk for deployment (OS + settings + kiosk mode + content).	Makes device stable and consistent.
Baseline Configuration	Standard settings: no sleep/lock, timezone, Chrome, tools.	Prevents interruptions during events.
Kiosk Mode	Locked mode: users can only use approved site/app.	Prevents tampering and keeps focus.
Autostart	Script/app starts automatically after reboot.	Required for recovery after power loss.
Ubuntu 20.04 LTS	Approved Linux OS for kiosks.	Compatibility with scripts/drivers.
Offline Mode / Localhost	Content runs locally without internet (http://localhost).	Used when venue internet is unstable.
WP-CLI	Command line tool for WordPress.	Offline deploy script uses it.
.wpress	WordPress backup file for All-in-One WP Migration.	Offline script restores it.
MeshCentral	Remote monitoring/remote access tool.	Allows remote support (only if enabled).
ZeroTier	Private network (VPN-like) tool.	Used when kiosks must reach internal services.
ETL Certification	Safety/compliance certification for electrical devices.	Required for new units.
DPMS / Screen Blanking	Power-saving that turns screen off.	Must be disabled so kiosk never sleeps.
xrandr	Linux display rotation/resolution command.	Used for portrait displays.
xinput	Linux touch mapping command.	Used when touch is rotated/misaligned.

3. Which Scenario Am I In? (Read This First)

Route yourself in under 60 seconds.

Step 1 - Identify lifecycle stage

- Event is running now -> Scenario F (Live Event Support)
- Event finished / unit returning / warranty case -> Scenario G (Post-Event / After-Sales)
- Preparing unit before deployment -> continue

Step 2 - Identify unit condition (hardware/OS)

- Brand new from supplier/factory -> Scenario A (Brand New Kiosk)
- Previously deployed and boots normally -> Scenario B (From Storage)
- Does not boot / unstable / kiosk mode crashes -> Scenario C (Failed Unit)

Step 3 - Identify content mode

- Stable internet available and approved -> Scenario D (Online Kiosk)
- Internet unreliable / offline required -> Scenario E (Offline Kiosk / Localhost)

Most common combinations

Hardware state / Content mode	Online (URL)	Offline (Localhost)
New	A + D	A + E
From storage	B + D	B + E
Failed (after rebuild)	C + D	C + E

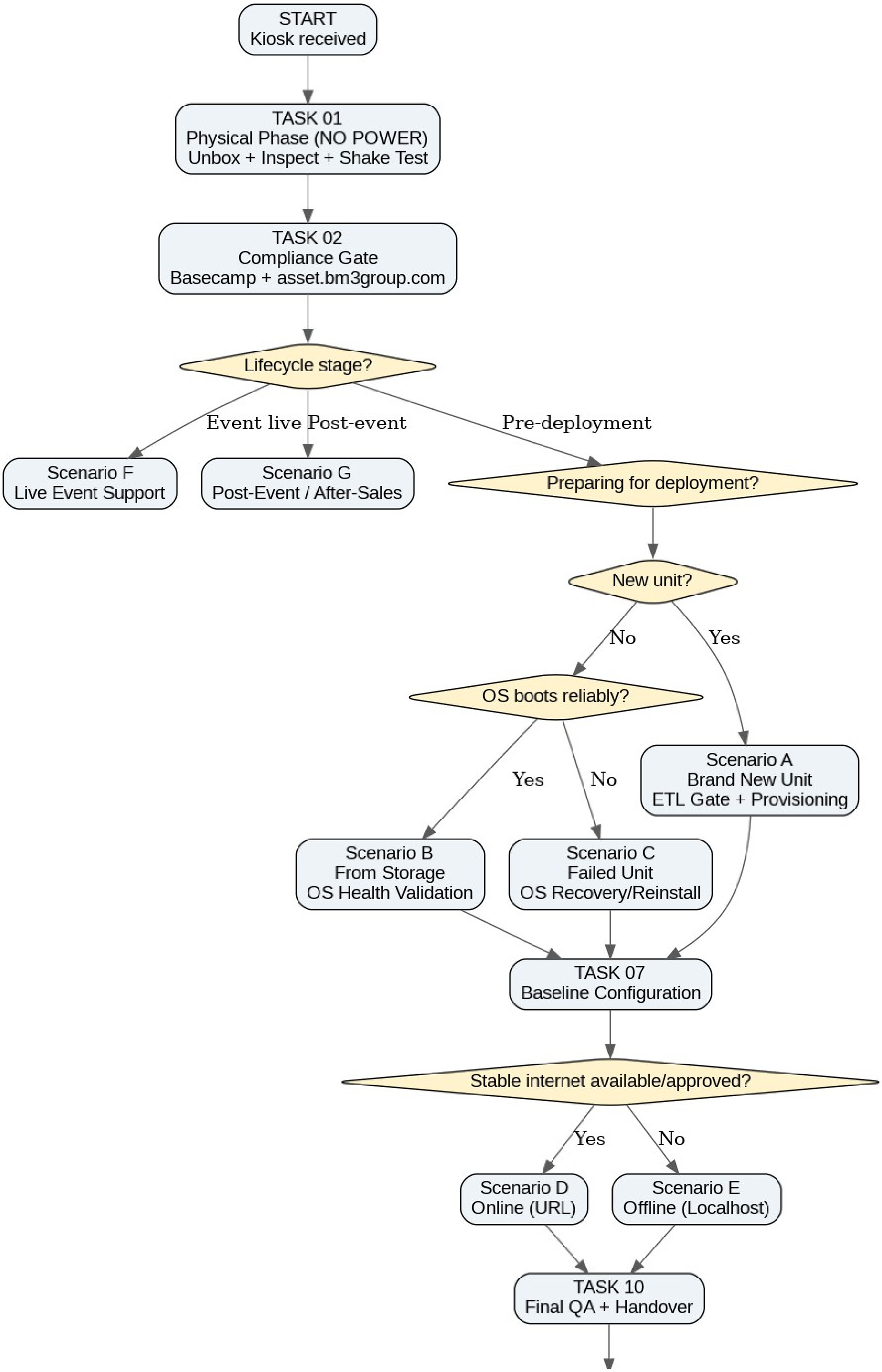
PRO TIP: How to read the scenarios

Scenarios A/B/C describe hardware/OS condition.

Scenarios D/E describe content mode.

Scenarios F/G describe lifecycle stage.

4. Decision Flow Diagram (Scenario Routing)


START
Kiosk received
TASK 01
Physical Phase (NO POWER)
Unbox + Inspect + Shake Test
TASK 02
Compliance Gate
Basecamp + asset.bm3group.com
Lifecycle stage?
Event live
Post-event
Pre-deployment
Scenario F
Live Event Support
Scenario G
Post-Event / After-Sales
Preparing for deployment?
New unit?
No
Yes
OS boots reliably?
Yes
No
Scenario A
Brand New Unit
ETL Gate + Provisioning
Scenario B
From Storage
OS Health Validation
Scenario C
Failed Unit
OS Recovery/Reinstall
TASK 07
Baseline Configuration
Stable internet available/approved?
Yes
No
Scenario D
Online (URL)
Scenario E
Offline (Localhost)
TASK 10
Final QA + Handover

ASCII Flow (works everywhere)

```
START
  |
  v
TASK 01 Structural Integrity (NO POWER) -> TASK 02 Compliance Gate (Basecamp + asset)
  |
  v
Lifecycle?
  - Event is live now -> Scenario F
  - Post-event / warranty -> Scenario G
  - Pre-deployment prep -> continue
  |
  v
Unit condition?
  - New unit -> Scenario A -> ETL Gate -> OS Provisioning -> Baseline -> Online/Offline -> QA/Handover
  - From storage (boots) -> Scenario B -> OS Health -> Baseline -> Online/Offline -> QA/Handover
  - Failed (no boot/unstable) -> Scenario C -> OS Recovery -> Baseline -> Online/Offline -> QA/Handover
```

Mermaid Flow (for systems that support Mermaid)

```
flowchart TD
  S([Start: Kiosk received]) --> P[Physical Phase (NO POWER)]
  P --> CG[Compliance Gate: Basecamp + asset.bm3group.com]
  CG --> L1{Lifecycle stage?}
  L1 -->|Event live| F[Scenario F: Live Event Support]
  L1 -->|Post-event / warranty| G[Scenario G: Post-Event / After-Sales]
  L1 -->|Pre-deployment prep| Q1{New unit?}
  Q1 -->|Yes| A[Scenario A: Brand New Unit]
  Q1 -->|No| Q2{OS boots reliably?}
  Q2 -->|Yes| B[Scenario B: Storage Unit]
  Q2 -->|No| C[Scenario C: Failed Unit]
  A --> ETL[ETL Certification Gate]
  ETL --> PROV[OS Provisioning]
  B --> OSH[OS Health Validation]
  C --> PROV
  OSH --> BASE[Baseline Configuration]
  PROV --> BASE
  BASE --> Q3{Stable internet?}
  Q3 -->|Yes| D[Scenario D: Online]
  Q3 -->|No| E[Scenario E: Offline]
  D --> QA[Final QA + Handover]
  E --> QA
  QA --> END([Go-Live Ready])
```

5. Field Execution Guide (Tasks)

Do tasks in order. Each task has a required output.

If you need to...	Go to...	Output
Confirm tools + keys	TASK 00	Field kit ready
Unbox safely + detect shipping damage	TASK 01	Photos + shake test pass
Log compliance (digital twin)	TASK 02	Basecamp log + asset record
Choose scenario	TASK 03	Scenario code (A/B/C + D/E or F/G)
Power-on safety + first boot	TASK 04	Safe boot confirmed
Validate storage unit OS health	TASK 05	OS Health checklist complete
Provision / reinstall OS	TASK 06	Ubuntu installed + kiosk user
Baseline configuration	TASK 07	No sleep + correct timezone + Chrome/tools
Online mode (URL)	TASK 08	URL autostarts + reconnect validated
Offline mode (Localhost)	TASK 09	Offline works with internet unplugged
Final QA + client handover	TASK 10	QA PDF + proof photos + sign-off
Live event support	TASK 11	2-hour checks + incident logs
Post-event pack-down + return	TASK 12	Photos + red/green tag + asset update

TASK 00 - Field Kit Readiness

Item	Required?	Why
Key 600 (service panel)	Required	Open back terminal and inspect internal cables
USB keyboard + USB mouse	Required	Configure OS, Wi-Fi, Chrome, scripts
2x USB sticks (8GB+)	Required	One installer, one backup/export
Ethernet cable (2m+)	Required	Stable network first choice
Surge protector/power bar	Recommended	Protects from unstable venue power
Outlet tester	Recommended	Detects wiring problems before damage
Cable ties/velcro	Required	Cable management and strain relief
Microfiber cloth	Required	Clean screen before photos/client walkthrough

CAUTION: Missing required items

If you are missing a REQUIRED item, stop and notify the PM before travel or before going onsite.

TASK 01 - Structural Integrity & Unboxing (NO POWER)

- Photograph flycase (all sides) -> proof of shipping condition.
- Open flycase carefully -> avoid cable snags and scratches.
- Two-person lift (equal pull) -> prevent frame twist.
- Physical shake test -> detect loose internal parts.
- Open service panel -> access PC ports and internal connectors.
- Connect USB keyboard and mouse -> reliable control for setup.

WARNING: Red Flag stop conditions

If you hear internal rattling, see exposed wiring, or find major damage: DO NOT POWER ON.

Post photos to Basecamp and wait for PM/IT Lead instruction.

Structural Integrity Checklist	Pass/Fail	Notes
Flycase photos captured		
Kiosk removed with 2-person equal pull		
Shake test passed (no rattling)		
Service panel opened with key		
Keyboard/mouse connected		
No cracks/dents/loose parts		

TASK 02 - Compliance Gate (Mandatory Before Power)

WARNING: No compliance log = no deployment

Create Basecamp log and update asset.bm3group.com before powering on for client use.

Basecamp Log Template (Copy/Paste)

```
Title: Deployment Start - <Serial> - <Client/Event> - <YYYY-MM-DD>

Include:
- Model + Serial
- Location (office / client / venue)
- Photos: flycase closed, flycase open, service panel open, accessories laid out
- Damage/missing parts notes
- Your name + time started
```

asset.bm3group.com field	Requirement
Serial number	Required
Model	Required
Status (In Prep / Deployed / Returned / In Repair)	Required
Location	Required
Project code	Required
Notes + photos	Required

TASK 03 - Scenario Confirmation (Fork in the Road)

- Scenario A: Brand new unit -> ETL gate + OS provisioning.
- Scenario B: From storage -> OS health validation + content update.
- Scenario C: Failed unit -> OS recovery/reinstall.
- Scenario D: Online -> URL mode.
- Scenario E: Offline -> localhost mode.
- Scenario F: Live event support.
- Scenario G: Post-event / warranty.

PRO TIP: Scenario coding

Write your scenario at the top of your notes (example: B + D).

This prevents confusion when multiple kiosks are being deployed.

TASK 04 - Power Safety & First Boot

- Inspect power cable for cuts or bent pins -> reduces short risk.
- Use surge protector whenever possible -> protects from unstable power.
- Test the outlet if you have an outlet tester -> prevents frying hardware.
- Plug into surge first, then wall -> reduces spike risk.
- Power ON kiosk -> confirm boot.

WARNING: Immediate power red flags

Burning smell, popping noise, flickering power, or abnormal heat -> unplug immediately and report.

Power-On Safety Checklist	Pass/Fail	Notes
Power cable inspected		
Outlet tested or surge used		
Screen turns on		
No unusual noise/smell		

TASK 05 - OS Health Validation (Scenario B: From Storage)

Run these commands (Terminal)

```
df -h
lsb_release -a
google-chrome --version
```

OS Health	Pass/Fail	Notes
Ubuntu 20.04 detected		
Chrome opens normally		
Disk space free (>=20%)		
Date/time and timezone correct		
No update popups blocking kiosk		
System stable for 5 minutes		

CAUTION: If OS health fails

Treat the unit as Scenario C (Failed) and proceed to TASK 06 (OS Provisioning/Reinstall).

TASK 06 - OS Provisioning / Reinstall (Scenario A + C)

- Create Ubuntu 20.04 installer USB.
- Boot from USB (press boot menu key repeatedly: often ESC/F12).

- Select 'Try Ubuntu' first -> hardware smoke test.
- Install Ubuntu 20.04 -> clean baseline.
- Create standard user 'kiosk' + enable auto-login.

PRO TIP: Why 'Try Ubuntu' first

It verifies hardware before wiping the drive. If touch/screen fails here, the problem is hardware, not software.

WARNING: Account standardization

Do not create personal accounts. Always use the standard kiosk account.

TASK 07 - Baseline Configuration (All Units)

- Disable sleep and screen lock -> prevents kiosk stopping mid-event.
- Set timezone -> correct logs and analytics.
- Install/verify Chrome -> kiosk runtime.
- Install required tools -> scripts and diagnostics.
- Install MeshCentral or ZeroTier only if required.

Baseline packages

```
sudo apt update
sudo apt install -y apt-transport-https ca-certificates curl software-properties-common
openssh-server unclutter wget
```

NOTE: Portrait kiosks

If the screen is portrait, you may need xrandr (rotation) and xinput (touch mapping). See Appendix: Display Rotation & Touch.

TASK 08 - Online Mode (Scenario D: Live Internet URL)

- Get the approved URL from the PM.
- Update kiosk config file to point to the URL.
- Reboot -> validate autostart.
- Unplug/replug network -> validate reconnect.

Kiosk Mode Validation	Pass/Fail	Notes
Kiosk launches automatically after reboot		
User cannot exit to desktop		
Idle return works		

TASK 09 - Offline Mode (Scenario E: WordPress/Localhost)

- Run the approved offline deployment script -> installs WordPress and restores content.
- Set kiosk start URL to http://localhost.

- Disconnect internet and retest -> proves offline readiness.
- Validate analytics logging -> enables post-event reporting.

WARNING: Credentials handling

Do NOT post WordPress credentials in Basecamp.

Use the approved secure source for credentials (per SOP).

TASK 10 - Final QA + Client Handover (Go-Live Gate)

Go-Live QA Tests	Pass/Fail	Evidence/Notes
Touch zig-zag test (full screen)		
Touch 10-point (corners + center)		
Network reconnect after unplug (online)		
Idle return works		
QR scanner works		
Printer prints test page (if present)		
Audio output ok + mic ok (if required)		
App/URL stable for 5 minutes		
Offline works with internet unplugged (if required)		

- Log QA results in the QA checklist sheet.
- Take photos of passing screens (home screen, QA pass).
- Export QA as PDF and upload to Basecamp.
- Client walkthrough + sign-off.

WARNING: No partial go-live

If any mandatory test fails, the kiosk is NOT live. Fix, retest, then log.

TASK 11 - Live Event Support (Scenario F)

- Every 2 hours: physical check (touch, QR, printer, network).
- If MeshCentral is enabled: verify device is online and screen is correct.
- Log issues immediately in Basecamp with photos and steps attempted.
- No experimental changes onsite.

CAUTION: Change control

Only apply approved fixes during events. Escalate if unsure.

TASK 12 - Post-Event Pack-Down / Return / After-Sales (Scenario G)

- Shut down safely -> prevents OS corruption.
- Disconnect peripherals -> avoid port damage.
- Coil and label cables -> faster next deployment.
- Wipe content if required -> client privacy.

- Photograph packed case (inside and outside) -> proof of return condition.
- Red/Green tag: Green ready; Red needs repair.
- Update asset status + location in asset.bm3group.com.

NOTE: Offline analytics

If WordPress offline was used, export analytics before wiping content.

6. Scenario Playbooks (A-G)

After you identify your scenario, follow that playbook. SOP links are included for deeper steps.

Scenario A - Brand New Kiosk (First-Time Deployment)

Definition

- Hardware has never been deployed.
- Fresh from supplier or factory.
- No prior OS, or OS must be re-installed.
- ETL certification required.

Execution Flow

- TASK 01 Physical Phase (No Power) -> TASK 02 Compliance Gate.
- ETL Certification Gate (Mandatory) -> contact ETL and wait for confirmation.
- TASK 06 OS Provisioning -> TASK 07 Baseline.
- TASK 08 (Online) or TASK 09 (Offline).
- TASK 10 Final QA + Client Handover.

SOPs Used (Source of Truth)

- OS Imaging / Baseline Setup SOP: https://3.basecamp.com/4938325/buckets/20576475/vaults/9409240123
- Kiosk Mode SOP (Online): https://3.basecamp.com/4938325/buckets/20576475/vaults/9409264331
- Kiosk Mode Assets / Uploads: https://3.basecamp.com/4938325/buckets/20576475/uploads/9409277758
- Pre-Rental Inspection & Preparation SOP: https://3.basecamp.com/4938325/buckets/20576475/vaults/9409270125
- Event Kiosk Rental IT Operations Checklist: https://3.basecamp.com/4938325/buckets/20576475/vaults/9409270125

Scenario B - Kiosk From Storage (Re-Deployment)

Definition

- Kiosk was previously deployed and returned to storage.
- OS boots normally.
- Hardware expected to be intact.

Execution Flow

- TASK 01 Physical Phase -> TASK 02 Compliance Gate -> TASK 04 Power safety.
- TASK 05 OS Health Validation.
- TASK 07 Baseline verification.
- TASK 08 (Online) or TASK 09 (Offline).
- TASK 10 Final QA.

SOPs Used (Source of Truth)

- Pre-Rental Inspection & Preparation SOP: https://3.basecamp.com/4938325/buckets/20576475/vaults/9409270125
- Event Kiosk Rental IT Operations Checklist: https://3.basecamp.com/4938325/buckets/20576475/vaults/9409270125
- Kiosk Mode SOP (Online): https://3.basecamp.com/4938325/buckets/20576475/vaults/9409264331
- Scripts Repo (Health Checks): https://3.basecamp.com/4938325/buckets/20576475/vaults/9409146592

Scenario C - Failed Kiosk (OS Corrupted / Unstable)

Definition

- Kiosk does not boot, reboots repeatedly, or kiosk mode crashes.
- System unstable or unusable.

Execution Flow

- TASK 01 Physical Phase -> TASK 02 incident log + set asset status.
- TASK 06 OS Recovery/Reinstall -> TASK 07 Baseline.
- TASK 08/09 Kiosk mode + content -> TASK 10 Full QA.

SOPs Used (Source of Truth)

- OS Imaging / Baseline Setup SOP: https://3.basecamp.com/4938325/buckets/20576475/vaults/9409240123
- Scripts Repo (Recovery & Health): https://3.basecamp.com/4938325/buckets/20576475/vaults/9409146592
- Pre-Rental Inspection SOP: https://3.basecamp.com/4938325/buckets/20576475/vaults/9409270125

Scenario D - Online Kiosk (Live Internet)

Definition

- Stable internet available.
- Kiosk points to live URL.
- Used for events, offices, showrooms.

Execution Flow

- Validate network (LAN -> Wi-Fi -> LTE).
- Configure kiosk URL + whitelist domains if required.
- Validate reconnect after unplug.
- Monitor uptime.

SOPs Used (Source of Truth)

- Kiosk Mode SOP (Online): https://3.basecamp.com/4938325/buckets/20576475/vaults/9409264331
- Live Event Support SOP: https://3.basecamp.com/4938325/buckets/20576475/vaults/9409270125

Scenario E - Offline Kiosk (WordPress / Localhost)

Definition

- No reliable internet.
- Content must work offline.
- Analytics stored locally.

Execution Flow

- Deploy offline WordPress package.
- Point kiosk to http://localhost.
- Disconnect internet and re-test.
- Validate analytics logging.
- Export analytics post-event.

SOPs Used (Source of Truth)

- WordPress Offline Deployment SOP (attached).
- Pre-Rental Inspection SOP: https://3.basecamp.com/4938325/buckets/20576475/vaults/9409270125

Scenario F - Live Event Support

Definition

- Kiosk already deployed; event running.
- Focus is uptime and user experience.

Execution Flow

- Physical check every 2 hours.
- Remote monitoring via MeshCentral (if enabled).
- Immediate issue logging.
- No experimental changes onsite.

SOPs Used (Source of Truth)

- Live Event Support SOP: https://3.basecamp.com/4938325/buckets/20576475/vaults/9409270125

Scenario G - Post-Event / After-Sales

Definition

- Event completed; kiosk returning to storage, sale, or repair.

Execution Flow

- Safe shutdown.
- Content wipe (if required).
- Analytics export (offline only).
- Red/Green tagging.
- Asset status update.

SOPs Used (Source of Truth)

- After Sales Support - Warranty Service SOP (attached).
- Live Event Support SOP: https://3.basecamp.com/4938325/buckets/20576475/vaults/9409270125

7. Troubleshooting (Pathfinding Matrix)

Symptom	First action	Next action	Escalate when
No power / dead screen	Try a different outlet + verify surge	Inspect power cable + internal power connection	Stop if smell/heat/sparks
Touch not responding	Check USB touch cable path + reboot	Reapply touch mapping (Appendix: xinput)	Escalate if device not detected
Touch misaligned after rotation	Confirm output name with xrandr	Apply correct xinput matrix	Escalate if still inverted
No internet	LAN -> Wi-Fi -> LTE fallback	Check DHCP lease / DNS	Escalate if venue blocks access
Kiosk exits to desktop	Reboot + validate autostart	Reapply kiosk script startup configuration	Escalate if repeats
Offline site not loading	Open http://localhost	Check apache2/mysql status	Escalate if restore fails
QR scanner not working	Replug USB / change port	Check detection (lsusb)	Escalate hardware failure
Printer not printing	Check power/cable + test page	Confirm driver and correct port	Escalate if driver missing

NOTE: Evidence

Take a photo of errors and attach to Basecamp. It speeds remote support.

8. Technical Appendix (Deep-Dive)

Use this section when provisioning or when a task fails. It contains deeper procedures and scripts.

Appendix 8A - OS Imaging & Baseline Setup (Ubuntu 20.04)

PRO TIP: Minimum safe install pattern

Boot installer USB -> Try Ubuntu (smoke test) -> Install Ubuntu 20.04 -> Create kiosk user -> Enable auto-login -> Disable sleep/lock -> Install Chrome/tools.

Verbatim text from 'How to set up a kiosk'

```
How to set up a kiosk
Step 1: Download the Installation Media
1. In a web browser, visit the Ubuntu download page and pick the Ubuntu version suitable
for your machine. The most popular versions include:
Ubuntu Desktop
2. Once you find the version you need, click the green Download button. You’ll be taken
to a thank-you page, and your download should start. (We will download and install
Ubuntu 20.04 for desktops.)
The download is an .iso file. You can use it to create a bootable USB drive.
3. Save the file to a location of your choice.
Step 2: Create Bootable USB
You will need a USB drive with 4GB or more. This process will delete all data on the USB
drive. Make sure to backup any existing data on the USB drive.
Option 1: Create a Bootable USB Drive on Ubuntu
Use the Create startup disk tool:
Open a search dialog, and type create startup.
If it’s not installed, the Software Center will offer the option to install it - choose
the option for USB drive, then open the utility.
In the top pane, click Other, then browse and select the Ubuntu 20.04 .iso file you
downloaded.
In the bottom pane, select your USB drive.
Click Make startup disk.
Option 2: Create Bootable USB Drive on Windows
You’ll need to install a third-party utility called Rufus to create a USB bootable
drive.
1. Download the Rufus utility. Scroll down to the download section and click the link to
download the latest version of Rufus.
2. Run the file once downloaded.
3. A pop-up dialog opens. You will be prompted whether you want to check for online
updates. Select No.
4. The Rufus utility launches. Plug in the USB drive - you should see the drive pop up
in the device field.
Set the USB as the device you wish to write to.
In the Boot Selection drop-down, click Disk or ISO Image.
Click the Select button to the right.
Browse and select the .iso Ubuntu file you downloaded earlier.
5. Click Start.
Step 3: Boot up Ubuntu from USB
1. Turn off your system. Make sure you remove all other USB devices, such as printers,
memory cards, etc.
2. Insert the Ubuntu USB drive into the system and turn on your machine.
There are two possible scenarios:
The computer boots the USB drive automatically.
You need to manually configure USB booting in the Boot Menu or BIOS/UEFI.
3. To manually configure the boot order, tap the boot menu key about once or twice per
second as soon as the computer powers on.
The boot menu key may be different depending on your computer manufacturer. Below is a
list of common boot keys associated to a brand:
```

```
4. Once you see your boot menu, use the arrows to pick the Ubuntu media to boot from.
For a DVD, the entry will usually have DVD or Optical in the name. USB is usually
labeled USB.
Your system should start loading the Ubuntu live disc menu.
Step 4: Run Ubuntu
You can test Ubuntu 20.04 before you commit to installing it. The .iso includes a live
mode that only runs in memory.
Launch this mode by clicking Try Ubuntu.
Step 5: Install Ubuntu 20.04 LTS Desktop
To begin the installation, click Install Ubuntu.
Choose Keyboard Layout
By default, the system will select English and English.
If you have a non-standard keyboard, you can select it in the list. Alternately, click
Detect Keyboard Layout and the system will automatically choose your keyboard. If you
need to test your keyboard, use the labeled field.
When you’re ready, click Continue.
Choose Starting Applications
Normal Installation - This is the full Ubuntu Desktop experience, with office software,
games, and media players.
Minimal Installation - Choose this to save disk space, especially if you won’t be using
media players or productivity software.
You’ll also be asked to confirm other options:
Download updates while installing Ubuntu - This does the work of downloading large
package files during the installation. Once the installation finishes, the packages will
be ready to apply as updates.
Install third-party software for graphics and Wi-Fi hardware and additional media
formats - Some hardware, like graphics cards and wi-fi cards, do not have open-source
driver support. Also, some media formats, such as .wmv, do not fall under the GPL
license. If you need support for these, you’ll need to agree to additional terms of use.
Disk Partitioning
Next, you’ll be presented with an Installation Type dialog. You can wipe the hard drive
clean prior to installing Ubuntu by clicking Erase disk and install Ubuntu. If you go
this route, skip ahead to the next step.
Click Continue to apply your changes to the drive partitions.
You’ll be asked to Write changes to disks?  None of the options you’ve selected are
permanent until you click Continue on this screen.  Click Continue to proceed.
Select Time Zone
Once the system formats the disk partitions, the installer will ask Where are you?
Type the nearest large city into the box, and the system will set your local time zone.
Click Continue.
Create User Account
Next, you’ll need to configure a user account. Fill in the following fields:
Name: Kiosk
Computer name: <SN>-<model>
Username: kiosk
Password: root
Log in automatically: Check this box
Click Continue to install Ubuntu.
Once the installer finishes, remove the Ubuntu installation media. You’ll be prompted to
Restart Now.
The system should boot into your fresh install of Ubuntu 20.04.
Step 6: Install Updates and programs
The following items will have to be installed on the Computer’
sudo apt install apt-transport-https ca-certificates curl software-properties-common
openssh-server unclutter wget curl google-chrome-stable chrome-gnome-shell
Install VPN
curl -s https://install.zerotier.com | sudo bash
sudo zerotier-cli join 8286ac0e47889b85
Install Docker
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal
stable"
apt-cache policy docker-ce
```

```
sudo apt install docker-ce
sudo systemctl status docker
sudo usermod -aG docker ${USER}
su - ${USER}
Groups
docker info
Install Chrome tools
Open chrome
https://chrome.google.com/webstore
GNOME Shell integration
Kiosk exit
Open chrome
https://extensions.gnome.org/
https://extensions.gnome.org/extension/4049/disable-gestures-2021/
Update Screen settings
Run the following to check the screen
Xinput
#!/bin/sh
# This is a comment!
xrandr --output eDP-1 --rotate left --primary --size 1080x1920
xinput set-prop 'ILITEK Multi-Touch-V5000' 'Coordinate Transformation Matrix' 0 -1 1 1 0
0 0 0 1
xinput set-prop 'ILITEK Multi-Touch-V5000 Mouse' 'Coordinate Transformation Matrix' 0 -1
1 1 0 0 0 0 1
kiosk@bm3sk:~$ export DISPLAY=:0
https://linuxconfig.org/set-wallpaper-on-ubuntu-20-04-using-command-line
```

Appendix 8B - Kiosk Script Architecture & Rules

WARNING: Do not modify production scripts unless approved

Use config files to change URLs/timeouts. If scripts must change, update at the source repo and document the change.

Kiosk Script Starter Guide (verbatim)

```
Kiosk Script Starter Guide
Definitive Reference for New Engineers & Technicians
Version 2.1 | Last Updated: 2024-12-01
1. Introduction & Purpose
This guide is the single source of truth for the kiosk ecosystem. It explains script
logic, dependencies, and operational workflows used in deployment, monitoring, and
maintenance.
All scripts follow the immutable principle-configuration changes only via .cfg files-and
are designed for graceful degradation when offline.
Audience: New engineers/technicians joining the kiosk team.
Goal: Enable you to understand, operate, and troubleshoot the kiosk platform with
confidence.
2. System Architecture & Directory Structure
Before running scripts, understand the environment layout.
All kiosks follow this standardized directory structure:
text
/opt/kiosk/
├── scripts/                    # Primary executable scripts
│   ├── kiosk_online.sh        # Core runtime engine
│   ├── system_inventory_audit.sh
│   ├── kiosk_health.sh
│   └── ubuntu-network-checker.py
```

```
├── config/                     # Configuration files
│   ├── kiosk_schedule.cfg     # Time-based URL scheduling
│   └── URL_Change.CFG.txt     # Main URL & fallback settings
├── assets/                     # Offline content
│   └── Offline_Page.html      # Local fallback page
└── logs/ → symlink → /var/log/kiosk/  # All runtime logs
Key Notes:
Scripts are owned by root:kiosk with 755 permissions.
Logs rotate daily via logrotate; retain for 30 days.
Never modify .sh or .py files directly on a live kiosk.
3. Kiosk Runtime & Content Control
3.1 kiosk_online.sh - The Core Engine
Purpose: Ensures Chrome runs continuously in kiosk mode, switches URLs based on
schedule, and falls back to offline content gracefully.
Operational Logic:
Init - Clears Chrome cache, locks, and Singleton files.
Connectivity Check - Pings 8.8.8.8 and a configurable internal endpoint.
Launch - Opens Chrome with flags:
bash
google-chrome --kiosk --incognito --no-first-run --disable-session-crashed-bubble
Watchdog Loop - Monitors Chrome PID; restarts instantly if crashed.
Schedule Check - Every 5 minutes, reads kiosk_schedule.cfg and switches URL if required.
Schedule Config Example (kiosk_schedule.cfg):
ini
[schedule]
morning_start=07:00
morning_url=https://example.com/morning
evening_start=18:00
evening_url=https://example.com/evening
default_url=https://example.com/default
3.2 URL_Change.CFG.txt & Offline Mode
File Location: /opt/kiosk/config/URL_Change.CFG.txt
Format: Key-value pairs.
ini
main_url=https://client-app.example.com
fallback_delay=30
offline_path=/opt/kiosk/assets/Offline_Page.html
Offline Page Requirements:
Must be self-contained (no external dependencies).
Include a “Reconnecting…” animation or progress indicator.
Auto-refresh every 60 seconds to retry connection.
Branded with client logo and support contact info.
4. Health, Inventory & Audit
4.1 system_inventory_audit.sh
When to Run: After a new build, before shipping, or when verifying hardware against
purchase orders.
Data Collected:
Serial Number (dmidecode -s system-serial-number)
MAC Addresses (ip link show)
Disk Health (smartctl -a /dev/sda)
Display Identification - Parses EDID to detect monitor model vs. “Generic PnP Monitor”.
Installed Software - Lists versions of Chrome, Python, and custom packages.
Usage:
bash
sudo /opt/kiosk/scripts/system_inventory_audit.sh > /var/log/kiosk/inventory-$(date +
%Y%m%d).log
4.2 kiosk_health.sh
Runs: Every 5 minutes via cron; can be executed manually for troubleshooting.
Checks & Alerts:
CPU - Warns if >90% for 5 minutes (suggests Chrome memory leak).
Temperature - Critical if >85°C (outdoor/enclosed units).
Zombie Processes - Lists defunct processes.
Service Health - Verifies kiosk_online.sh is running; restarts via systemctl if dead.
```

```
Disk Space - Warns if <20% free.
Auto-Recovery Actions:
Attempts graceful Chrome restart.
If fails, reboots the kiosk (reboot).
Logs all actions to /var/log/kiosk/health.log.
5. Network Diagnostics & Connectivity
5.1 ubuntu-network-checker.py
Purpose: Deep-dive network analysis when “internet is down” but hardware seems fine.
Checks Performed:
DNS Resolution - Internal vs. external domains.
MTU Issues - Sends large packets to detect fragmentation.
Firewall/Port Blocks - Tests TCP connectivity to ports 80, 443, 10050 (Zabbix).
Gateway Reachability - Validates local router response.
Execution:
bash
python3 /opt/kiosk/scripts/ubuntu-network-checker.py --verbose
Troubleshooting Tip: If kiosk_online.sh is stuck offline but this script reports “All
checks pass,” suspect Chrome cache or DNS caching.
6. Monitoring & Reporting (Zabbix Integration)
6.1 report.py (API Integration)
Purpose: Pull uptime statistics from Zabbix for SLA reporting.
Data Points:
Mean Time Between Failures (MTBF)
Uptime % (99.9% SLA verification)
Last incident timestamp
Usage (from management workstation):
bash
python3 report.py --host kiosk-id-123 --days 30
6.2 Zabbix_Export_JS (Browser-Based Fallback)
When to Use: When Zabbix API is unavailable or access restricted.
Method:
Log into Zabbix web interface.
Open browser Developer Tools (F12).
Paste the contents of Zabbix_Export_JS into the console.
Script scrapes the dashboard and downloads a CSV.
7. Utilities & Recovery (The “Reset” Buttons)
7.1 Chrome & Keyring Reset
Problem: Chrome shows “Restore Pages?” or “Sign in to Keyring” popups.
Fix:
bash
# Remove Chrome locks
rm -rf ~/.config/google-chrome/SingletonLock
rm -rf ~/.config/google-chrome/SingletonSocket
# Clear keyring prompts
rm -rf ~/.local/share/keyrings/*.keystore
# Optional: Reset entire Chrome profile (use with caution)
# mv ~/.config/google-chrome ~/.config/google-chrome.backup
Warning: This logs the kiosk out of any manually authenticated websites.
7.2 Force Offline Test
Simulate network failure to verify offline content:
bash
sudo iptables -A OUTPUT -p tcp --dport 80 -j DROP
sudo iptables -A OUTPUT -p tcp --dport 443 -j DROP
# Wait 30-60 seconds, Chrome should switch to offline page.
# Revert:
sudo iptables -D OUTPUT -p tcp --dport 80 -j DROP
sudo iptables -D OUTPUT -p tcp --dport 443 -j DROP
8. Operational Workflow for New Engineers
9. Key Principles
Immutable Scripts - Modify only .cfg files on live kiosks. Script changes go through Git
and are deployed via Ansible.
Logs Are King - Always check /var/log/kiosk/ before blaming hardware.
Graceful Fallback - A kiosk must never show a browser error. Offline content must be
```

```
branded and reassuring.
Watchdog Mentality - Every script should have built-line monitoring and self-healing
where possible.
Documentation - Update this guide when procedures change.
10. Next Steps
Familiarize yourself with the directory structure on a test kiosk.
Execute each script in a sandbox environment to understand output.
Bookmark the troubleshooting flowchart (see separate document).
Memorize the quick-start command cheat sheet (see separate document).
For Support:
Check logs first: tail -f /var/log/kiosk/kiosk.log
Escalate to team lead if issue persists >15 minutes.
Document ID: KSSG-2.1 | Classification: Internal Use Only
© 2024 Kiosk Operations Team. All rights reserved.
```

Ubuntu Chrome Kiosk Script Setup Guide (verbatim)

```
📄 README: Ubuntu Chrome Kiosk Script Setup Guide
🟢 Overview
This script automatically launches Google Chrome in kiosk mode, continuously monitors
connectivity, and optionally switches to an offline page when internet connectivity
fails.
🧾 Files
Script file:
 /home/kiosk/start-Online-kiosk.sh

Configuration file:
 /home/kiosk/kiosk_config.cfg
 (Contains the main_page and offline_video_page URLs)

Loading screen HTML:
 /home/kiosk/loading.html

💡 Prerequisites
Ubuntu Desktop installed.

Google Chrome installed.

unclutter installed to hide mouse cursor (optional but included).

Install required packages if not already done:
sudo apt update
sudo apt install unclutter wget xdotool
🔐 File Ownership
After copying the script and config files to /home/kiosk, set ownership:
sudo chown kiosk:kiosk /home/kiosk/start-Online-kiosk.sh
sudo chown kiosk:kiosk /home/kiosk/kiosk_config.cfg
sudo chown kiosk:kiosk /home/kiosk/loading.html
🔑 Permissions
Make the script executable:
sudo chmod +x /home/kiosk/start-Online-kiosk.sh
💻 Add to Startup (Ubuntu Desktop GUI)
1️⃣ Log in as kiosk user on the Ubuntu Desktop.
2️⃣ Open the Startup Applications tool.
 (Search “Startup Applications” from the Activities overview.)
3️⃣ Click Add.
4️⃣ Enter:
Name: Chrome Kiosk

Command: /home/kiosk/start-Online-kiosk.sh
```

```
Comment: Launches Chrome in kiosk mode on startup

5️⃣ Click Add and close.
✔️ On next login, the script will start automatically.
🌐 MeshCentral Management
You can remotely connect to the kiosk machine via MeshCentral.

From MeshCentral, you can open a terminal or use file manager to:

Update the kiosk_config.cfg file remotely.

Restart the machine if needed.

Adjust script parameters without local access.

🛡️ Sudo Access
The IT Specialist will provide you with the sudo password when needed (e.g., for initial
setup or package installation).

Once ownership is set to kiosk:kiosk and permissions are correct, daily operation does
not require sudo.

✅ Configuration File Example
File: /home/kiosk/kiosk_config.cfg
main_page="https://your-main-website.com"
offline_video_page="file:///home/kiosk/offline.html"  # Optional: local fallback page
If you do not want an offline fallback, simply provide only main_page and leave
offline_video_page blank or remove logic in the script.
🚀 How to Test
/home/kiosk/start-Online-kiosk.sh
Chrome should open in kiosk mode.

Mouse cursor should hide after 1 second of inactivity.

Config changes should reload automatically.

⚙️ Troubleshooting
💬 Support
If issues persist, please contact IT Specialist using MeshCentral, or provide logs (you
can add logs to /var/log/kiosk.log in the script for advanced debugging).
```

Appendix 8C - Online Kiosk Engine Scripts

start-Online-kiosk.sh (full script)

```
#!/bin/bash

# Configuration
CONFIG_FILE="/home/kiosk/kiosk_config.cfg"
LOADING_SCREEN="file:///home/kiosk/loading.html"  # Path to your loading GIF page

# Static settings
check_interval=5      # Interval between checks (in seconds)
max_failed_checks=3   # Allow 3 consecutive failures before switching offline

# Load configuration
load_config() {
    if [ -f "$CONFIG_FILE" ]; then
        source "$CONFIG_FILE"
        echo "Config loaded - Main: $main_page, Offline: $offline_video_page"
    else
        echo "ERROR: Config file missing!"
        exit 1
    fi
}

# Initial load
load_config

# State variables
current_mode="booting"
chrome_pid=0
failed_checks=0
last_main_page="$main_page"
last_offline_page="$offline_video_page"

# -------------------------------------------------------------------
# FUNCTIONS
# -------------------------------------------------------------------

check_internet() {
    wget --timeout=2 --tries=2 --spider http://clients3.google.com/generate_204
>/dev/null 2>&1
    return $?
}

check_chrome_errors() {
    [ $chrome_pid -eq 0 ] && return 1

    if ! kill -0 $chrome_pid 2>/dev/null; then
        return 1
    fi

    if command -v xdotool >/dev/null; then
        if xdotool search --name "Error|Crash|Aw, Snap!" >/dev/null; then
            return 1
        fi
    fi

    return 0
}

start_chrome() {
    local url="$1"
    local show_loading="$2"

    echo "Preparing to launch Chrome..."

    # Optionally show loading screen
    if [ "$show_loading" = true ]; then
        echo "Showing loading screen..."
        pkill -f "chrome.*--kiosk"
        google-chrome --password-store=basic --kiosk --no-first-run \
            --disable-pinch --disable-infobars --noerrdialogs \
            "$LOADING_SCREEN" >/dev/null 2>&1 &
        sleep 2
    fi

    echo "Loading URL: $url"
    pkill -f "chrome.*--kiosk"
    google-chrome --password-store=basic --kiosk --no-first-run \
        --disable-pinch --disable-infobars --noerrdialogs \
        --disable-session-crashed-bubble --no-default-browser-check \
        --disable-component-update --check-for-update-interval=31536000 \
        "$url" >/dev/null 2>&1 &
    chrome_pid=$!

    command -v unclutter >/dev/null && unclutter -idle 1 &
}

# ---------------------------------------------------------------------
# MAIN LOOP
# ---------------------------------------------------------------------

# Disable screen blanking and power management
xset s off
xset s noblank
xset -dpms

# Start Chrome initially
start_chrome "$main_page" true
current_mode="online"

while true; do
    load_config

    # URL change detection
    if [ "$last_main_page" != "$main_page" ] || [ "$last_offline_page" !=
"$offline_video_page" ]; then
        echo "URL config changed - reloading..."
        last_main_page="$main_page"
        last_offline_page="$offline_video_page"
        start_chrome "$main_page" true
        current_mode="online"
        failed_checks=0
        continue
    fi

    # Chrome health check
    if ! check_chrome_errors; then
        echo "Detected Chrome issue - restarting..."
        start_chrome "$main_page" true
        current_mode="online"
        failed_checks=0
        continue
    fi

    # Internet state machine
    if check_internet; then
        failed_checks=0
        if [ "$current_mode" != "online" ]; then
            echo "Internet restored - switching to main page"
            start_chrome "$main_page" true
            current_mode="online"
        fi
    else
        echo "Internet offline (consecutive failures: $failed_checks)"
        ((failed_checks++))

        if [ "$current_mode" = "online" ] && [ $failed_checks -ge $max_failed_checks ];
then
            echo "Switching to offline content..."
            start_chrome "$offline_video_page" true
            current_mode="offline"
            failed_checks=0
        fi
    fi

    sleep $check_interval
done
```

kiosk_config.cfg (example)

```
main_page="https://bm3demos.com/cme2024/digital-ads/"
offline_video_page="file:///path/to/offline.html"
```

Appendix 8D - Health Check Script (kiosk_health.sh)

How to run

```
sudo bash kiosk_health.sh
```

- CPU load -> detects runaway processes
- Memory usage -> detects low RAM conditions
- Disk usage -> ensures >=20% free space
- CPU temperature -> detects overheating risk
- Disk SMART health -> detects failing drives
- Critical logs -> finds severe OS errors

kiosk_health.sh (full script)

```
#!/bin/bash

# Function to install packages if missing
install_if_missing() {
    local cmd="$1"
    local pkg="$2"
    if ! command -v "$cmd" &> /dev/null; then
        echo "[*] Installing $pkg..."
        sudo apt update -qq && sudo apt install -y "$pkg"
    fi
}

echo "===== Checking/Installing Required Tools ====="
install_if_missing "sensors" "lm-sensors"
install_if_missing "smartctl" "smartmontools"
install_if_missing "iostat" "sysstat"

echo -e "\n===== Running System Health Checks ====="

# --- CPU Load ---
echo -e "\n[CPU Load]"
cpu_load=$(uptime | awk -F 'load average:' '{print $2}' | awk '{print $1}')
cpu_cores=$(nproc)
echo "CPU Load (1min avg): $cpu_load (Cores: $cpu_cores)"
echo "Normal range: Below $cpu_cores (High load if > $cpu_cores)"
```

```
# --- Memory Usage ---
echo -e "\n[Memory Usage]"
free -h | grep "Mem:"

# --- Disk Usage ---
echo -e "\n[Disk Usage]"
df -h --exclude-type=tmpfs --exclude-type=devtmpfs

# --- CPU Temperature ---
echo -e "\n[CPU Temperature]"
if command -v sensors &> /dev/null; then
    sensors | grep -E "Core|Package"
else
    echo "[!] 'sensors' still missing after install attempt."
    echo "    Tip: You may need to run 'sudo sensors-detect' once to enable full sensor support."
fi

# --- Disk Health (SMART) ---
echo -e "\n[Disk Health (SMART)]"
if command -v smartctl &> /dev/null; then
    root_disk=$(lsblk -ndo NAME,TYPE | awk '$2=="disk"{print $1; exit}')
    echo "Checking /dev/$root_disk:"
    sudo smartctl -a /dev/"$root_disk" | grep -E "Model|Reallocated|Pending|Temperature|Health"
else
    echo "[!] 'smartctl' still missing after install attempt."
fi

# --- System Logs (Critical Errors) ---
echo -e "\n[System Logs - Critical Errors]"
log_output=$(journalctl -p 3 -xb --no-pager | tail -n 10)
if [ -z "$log_output" ]; then
    echo "No critical errors found in logs."
else
    echo "$log_output"
fi

echo -e "\n===== Health Check Complete ====="
```

Health check reporting notes (verbatim)

```
📝 Notes for Report & SOP - Kiosk Health Check Script
Purpose
This script is designed to verify system readiness for 24/7 kiosk operation (Ubuntu 20.04 + Chrome in kiosk mode).
 It runs a preflight check before events, focusing on:
System uptime & load (ensures the system isn’t overloaded).

Disk usage & cleanup (prevents crashes from running out of space).

Memory & swap availability (checks RAM and backup space).

System errors (alerts if too many recent critical errors).

Network connectivity (verifies that internet is stable).

Color Coding in Report
🟢 PASS (Green) = Healthy, no action needed.
```

```
🟡 WARNING (Yellow) = Monitor or take preventive action.

🔴 FAIL (Red) = Critical issue, must be fixed before event.

Automatic Fixes Included
If disk usage ≥ 80%, the script will automatically:
Clean unused packages (apt autoremove).

Clear cached packages (apt clean, autoclean).

Limit system logs to 100MB.

Clear Chrome cache for the kiosk user.

Remove old/disabled Snap packages.

It then shows how much space was freed:
Disk cleanup complete: freed space from 12G -> 18G
How to Run
Save script as kiosk_health_check.sh.

Make it executable:

 chmod +x kiosk_health_check.sh
Run it:

 ./kiosk_health_check.sh
Review the color-coded output.

Example Interpretation
>>> [2] Disk Space
 Root Disk Usage: 83% -> WARNING (Clean up soon)
 Running cleanup tasks...
 Cleared Chrome browser cache
 Removed old Snap packages
 Disk cleanup complete: freed space from 12G -> 18G
✅ Disk was almost full but the script fixed it automatically.
>>> [4] Recent System Errors
 System Errors: 3 -> WARNING (Some errors logged, usually safe)
⚠️ A few errors were logged. Usually safe, but if the number is high (>10), escalate to
IT.
>>> [5] Network Check
 Active Interface: wlp2s0
 Internet: Latency 25ms -> PASS
✅ Network is connected and stable.
SOP Actions
If all checks PASS → System is READY.

If WARNING appears → Note it, rerun script after cleanup, or escalate if repeated.

If FAIL appears → Must be resolved before kiosk is used (replace cable, free space,
check hardware).

⚡ With these notes, anyone running the script will know:
What’s being checked.

What the results mean.

What actions are taken automatically.

When to escalate to IT support.
```

Appendix 8E - Google Chrome Auto-Installer & Updater

README (verbatim)

```
📄 README: Google Chrome Auto-Installer & Updater Script
✅ Overview
This script automates the installation and silent updating of Google Chrome Stable on
Debian/Ubuntu systems. It ensures Chrome is installed, always up to date with no user
prompts, and configured for kiosk or managed environments.
💡 Features
Installs Google Chrome Stable if not already installed.

Updates Chrome to the latest stable version silently on each run.

Disables first-run and crash recovery dialogs.

Configured for non-interactive, kiosk-friendly environments.

Idempotent: installs once, then updates as needed.

🧾 What the script handles
✅ Installs Google Chrome Stable if missing.

🔁 Updates Chrome to the latest stable version if already installed.

🚫 Disables first-run and metrics dialogs.

💻 Configured for kiosk or managed environments.

♻️ Safe to run multiple times.
⚙️ Running the script via MeshCentral
1️⃣ Upload the script
Upload Chrome_update.sh to the device (e.g., via MeshCentral’s Files → Upload) into:
/home/kiosk/Chrome_update.sh
2️⃣ Make it executable
Open a terminal session (via MeshCentral), then run:
chmod +x /home/kiosk/Chrome_update.sh
3️⃣ Change ownership to kiosk:kiosk
sudo chown kiosk:kiosk /home/kiosk/Chrome_update.sh
🚀 Adding to Startup Applications (graphical method)
If you want this script to run every time the kiosk user logs in, add it to Startup
Applications:
1️⃣ Open Startup Applications
Press Super (Windows) key and search for Startup Applications, or
Go to Settings → Session and Startup
Add the chrome_update.sh file from home/kiosk directory.

2️⃣ Click Add
Name: Chrome Installer & Updater

Command: bash /home/kiosk/Chrome_update.sh
Comment: Installs or updates Google Chrome silently on login

3️⃣ Save
On each login, the script will check for Chrome and update it if needed.
👤 Ownership
The script file should be owned by kiosk:kiosk:
sudo chown kiosk:kiosk /home/kiosk/Chrome_update.sh
💬 Example usage summary
chmod +x /home/kiosk/Chrome_update.sh
sudo chown kiosk:kiosk /home/kiosk/Chrome_update.sh
Then add it to Startup Applications as described above.
```

```
⚠️ Important Notes
To install or update system packages, root privileges are required.

By default, Startup Applications runs as the logged-in user, without sudo.


If you'd like, I can prepare the passwordless sudo configuration steps or modify the script to support both root and non-root scenarios.
```

chrome_update.sh (full script)

```
#!/bin/bash

set -e

# Ensure script runs as root
if [[ "$EUID" -ne 0 ]]; then
  echo "Please run this script as root"
  exit 1
fi

# Check if Google Chrome is already installed
if command -v google-chrome >/dev/null 2>&1; then
  echo "Google Chrome is already installed. Updating..."
  apt update -y
  apt install --only-upgrade -y google-chrome-stable
  exit 0
fi

# Proceed with fresh installation
echo "Installing Google Chrome..."

apt update -y
apt install -y wget gnupg ca-certificates apt-transport-https software-properties-common

wget -q -O - https://dl.google.com/linux/linux_signing_key.pub | gpg --dearmor -o /usr/share/keyrings/google-chrome.gpg

echo "deb [arch=amd64 signed-by=/usr/share/keyrings/google-chrome.gpg] http://dl.google.com/linux/chrome/deb/ stable main" > /etc/apt/sources.list.d/google-chrome.list

apt update -y
apt install -y google-chrome-stable

mkdir -p /etc/opt/chrome/policies/managed
cat <<EOF > /etc/opt/chrome/policies/managed/auto_config.json
{
  "HideWebStorePromo": true,
  "MetricsReportingEnabled": false,
  "RestoreOnStartup": 4,
  "BrowserAddPersonEnabled": false,
  "BrowserGuestModeEnabled": false,
  "ImportBookmarks": false,
  "ImportHistory": false,
  "ImportSavedPasswords": false,
  "ImportSearchEngine": false
}
EOF
```

```
echo "Google Chrome installed and configured."
```

Appendix 8F - Offline WordPress Deployment (Localhost)

CAUTION: Disk space

Offline WordPress deployments can consume significant disk space. Confirm >=20% free disk space before deploying.

Offline Deployment SOP (verbatim)

```
WordPress Migration & Kiosk Deployment Guide
WordPress Migration & Kiosk Deployment Guide
This guide covers the automated processes performed by the wordpress-db-restore2.sh
script and the manual steps required to finalize the site migration and Kiosk mode
setup.
Part 1: Automated Script Actions
The script handles the environment setup and data restoration to ensure the site is
ready for local use without manual file transfers.
WP-CLI Installation: Downloads and installs the WordPress Command Line Interface to
allow for automated site management.
Core Setup: Installs a fresh WordPress database at http://localhost with pre-defined
admin credentials.
All-in-One Migration Setup:
Installs and activates the standard All-in-One WP Migration plugin.
Downloads and installs the Unlimited Extension zip file from a secure S3 bucket to
bypass default upload size limits.
Backup Restoration: * Downloads the site backup file (cme-2024-v3.wpress) directly to
the server.
Automatically triggers the ai1wm restore command to overwrite the clean install with the
actual site data.
System Configuration: * Sets directory permissions to www-data and 755 for security and
stability.
Updates the database and sets the permalink structure to /%postname%/.
Part 2: Manual Post-Installation Steps
Once the script has finished executing, follow these steps to verify the site and
activate Kiosk mode.
1. Finalize the WordPress Migration
Login to Dashboard: Go to http://localhost/wp-admin/ in your browser.
Note: Use the credentials from the original site that was backed up, as the script's
temporary credentials will have been replaced during restoration.
Verify Permalinks: Go to Settings > Permalinks. Scroll to the bottom and click Save
Changes. This is a critical step to ensure all internal links and pages resolve
correctly.
Check Plugins: Ensure both the "All-in-One WP Migration" and the "Unlimited Extension"
are active in the Plugins menu.
2. Kiosk Mode & URL Configuration
To reflect the local WordPress site on the kiosk browser, you must update the system's
configuration file.
Edit the .cfg File: Locate the configuration file (e.g., kiosk.cfg or the system's
browser startup script) used by your kiosk software.
Update the URL: Set the primary startup URL to: http://localhost
Verification: Restart the kiosk or the browser service. The browser should now
automatically launch in a restricted, full-screen mode displaying your migrated
WordPress site.
3. Clean Up
Remove Installation Files: Once the site is verified, you can safely delete the .sh
script and the .wpress file from the /wp-content/ai1wm-backups/ directory to save disk
```

```
space.
Wordpress Offline deployment
Step 1:- Download the wordpress offline deployment script from below basecamp link
wordpress-offline-deploy2 (1).sh
Step 2:- Transfer the script to the kiosk using meshcentral and execute below commands.
sudo chmod +x wordpress-offline-deploy.sh
sudo ./wordpress-offline-deploy.sh
Step 3:- Once the script is executed go to this URL http://localhost/wp-admin/ on
browser login to wordpress.
username :- jolo@brandm3dia.com
password :- %DY@+^Q39a38Cse
Step 4:- Go to Settings -> plug-ins and install WordPress all-in-one extension and
unlimited extension plugin.
You can get unlimited extension from below link
https://bm3-wordpress-files.s3.ca-central-1.amazonaws.com/all-in-one-wp-migration-unlimited-extension.zip
Step 5:- Go to Settings -> plugins -> All in one Wordpress migration plugin and click on
update.
Step 6:- Go to settings -> plugins -> Select All in one wordpress migration plugin and
restore database from .wpress file.
Step 7:- Go to settings -> permalink -> scroll down and click on save.
```

wordpress-offline-deploy (full script)

```
#!/bin/bash

WORDPRESS_PATH="/var/www/html"
APACHE_CONF="/etc/apache2/sites-available/wordpress.conf"
DOMAIN_NAME="127.0.0.1"
WORDPRESS_FILE=cme-2024-v3.wpress
WORDPRESS_FILE_URL=https://bm3-wordpress-files.s3.ca-central-1.amazonaws.com/cme-2024-v3.wpress


# Update and upgrade the system
sudo apt update && sudo apt upgrade -y

# Install Apache
sudo apt install apache2 -y
sudo systemctl enable apache2
sudo systemctl start apache2

# Install MySQL
sudo apt install mysql-server -y

# Secure MySQL installation
sudo mysql_secure_installation <<EOF

y
0
y
y
y
y
EOF

# Install PHP
sudo apt install php libapache2-mod-php php-mysql php-cli php-curl php-zip php-gd php-mbstring php-xml php-soap php-intl php-bcmath -y
```

```
# Download and Install WordPress
cd /var/www/html
sudo wget https://wordpress.org/latest.tar.gz
sudo tar -xvzf latest.tar.gz
sudo mv wordpress/* .
sudo rm -r wordpress latest.tar.gz

# Set permissions
sudo chown -R www-data:www-data /var/www/html/
sudo chmod -R 755 /var/www/html/

# Create MySQL Database and User
DB_NAME="wordpress_db"
DB_USER="wordpress_user"
DB_PASSWORD="your_password"

sudo mysql -u root -p <<MYSQL_SCRIPT
CREATE DATABASE $DB_NAME;
CREATE USER '$DB_USER'@'localhost' IDENTIFIED BY '$DB_PASSWORD';
GRANT ALL PRIVILEGES ON $DB_NAME.* TO '$DB_USER'@'localhost';
FLUSH PRIVILEGES;
EXIT;
MYSQL_SCRIPT

# Configure WordPress
sudo cp wp-config-sample.php wp-config.php

# Update wp-config.php with database information
sudo sed -i "s/database_name_here/$DB_NAME/" wp-config.php
sudo sed -i "s/username_here/$DB_USER/" wp-config.php
sudo sed -i "s/password_here/$DB_PASSWORD/" wp-config.php



# Configure Apache 2 for WordPress
if ! grep -q "$WORDPRESS_PATH" "$APACHE_CONF"; then
    echo "Configuring Apache 2 for WordPress..."
    sudo tee "$APACHE_CONF" > /dev/null <<EOL
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot $WORDPRESS_PATH
       ServerName localhost
       ServerAlias localhost

    ErrorLog \${APACHE_LOG_DIR}/error.log
    CustomLog \${APACHE_LOG_DIR}/access.log combined

    <Directory $WORDPRESS_PATH>
        AllowOverride All
       Require all granted
    </Directory>
</VirtualHost>
EOL
    sudo a2enmod rewrite
  sudo a2ensite wordpress.conf
    sudo systemctl restart apache2
fi

sudo mv /var/www/html/index.html /tmp/

# download curl package
sudo apt-get install curl
```

```
# Download wp-cli
curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar

# Make the file executable
chmod +x wp-cli.phar

# Move the file to a location that is in your system's PATH
sudo mv wp-cli.phar /usr/local/bin/wp

# Verify the installation
wp --info

#installing core database of wordpress
wp --allow-root core install --url="http://localhost" --title="taccom" --
admin_user="jolo@brandm3dia.com" --admin_password="%DY@+^Q39a38Cse" --
admin_email="jolo@brandm3dia.com" --path=/var/www/html/ --skip-email
wp plugin install all-in-one-wp-migration --activate --allow-root --path=/var/www/html/
wget https://bm3-wordpress-files.s3.ca-central-1.amazonaws.com/all-in-one-wp-migration-
unlimited-extension.zip
# Set permissions
sudo chown -R www-data:www-data /var/www/html/
sudo chmod -R 755 /var/www/html/
wp plugin install ./all-in-one-wp-migration-unlimited-extension.zip --allow-root --
activate --path=/var/www/html
wget $WORDPRESS_FILE_URL
cp $WORDPRESS_FILE /var/www/html/wp-content/ai1wm-backups/

# Set permissions
sudo chown -R www-data:www-data /var/www/html/
sudo chmod -R 755 /var/www/html/

wp plugin update all-in-one-wp-migration-unlimited-extension --allow-root
--path=/var/www/html/

echo "y" | wp ai1wm restore $WORDPRESS_FILE --allow-root --path=/var/www/html

# Set permissions
sudo chown -R www-data:www-data /var/www/html/
sudo chmod -R 755 /var/www/html/

wp --allow-root core update-db --path=/var/www/html
wp option update permalink_structure "/%postname%/" --allow-root --path=/var/www/html
wp rewrite structure '/%postname%/' --allow-root --path=/var/www/html/

wp --allow-root rewrite flush --path=/var/www/html

# Set permissions
sudo chown -R www-data:www-data /var/www/html/
sudo chmod -R 755 /var/www/html/
```

Appendix 8G - Pre-Rental Inspection & Preparation (QA)

Pre-Rental Inspection & Preparation SOP (verbatim)

```
Purpose
To ensure all kiosk systems are properly inspected, configured, and prepared before
rental deployment, maintaining security, functionality, and compliance with Brandm3dia
standards.
Scope
```

```
Applies to all IT operations related to kiosk and Robot inspection, preparation, and
readiness checks prior to rental deployment.
Responsibilities
IT Head: Perform inspections, health checks, content uploads, and functional tests.
Project Manager:- Provide content and confirm requirements and Approves configurations
and ensures compliance.
Owner Department
IT Department
General Purpose
To standardize the inspection, configuration, and preparation of all kiosk systems
(standard and robotic) before rental deployment, ensuring:
Functionality: Hardware/software operational integrity
Offline Capability: WordPress/local content fallback
Analytics: Tracking verification
Security: remote monitoring
Inspection & Preparation Steps
Physical Inspection
Exterior: No cracks, dents, or cosmetic damage.
Cables/Ports: All connections secure (power, USB, peripherals).
Display: No dead pixels, brightness adjusted.
System Health Check
OS & Software:
Confirm Ubuntu 20.04 LTS (or approved version).
Update Chrome to the stable kiosk mode version.
Verify GNOME extensions (Caffeine, Hide Top Bar, etc.).
Storage: Minimum 20% free disk space.
Network: Stable Wi-Fi/Ethernet (test upload/download speeds).
Content Upload
Source: Use content provided by the Project Manager through Basecamp (via USB/cloud).
Destination: Copy to /home/kiosk/ (overwrite old files if needed).
Files to Check:
loading.html & loading.gif (offline fallback).
kiosk_config.cfg (ensure correct settings).
Edit kiosk_config.cfg:
url=http://digitl signage url
If you are offline, please follow the steps below for WordPress.
Start-Online-kiosk.sh (test execution).
WordPress Offline Mode & Analytics
Localhost Setup:
Edit kiosk_config.cfg:
ini
url=http://localhost
Verify:
loading.html displays before the redirect.
Apache/MySQL are running:
sudo systemctl status apache2 mysql
Analytics Verification:
Google Analytics:
Check real-time reports for activity.
Validate tracking code in /var/www/html/wp-header.php.
Local Logs:
bash
tail -10 /var/log/apache2/access.log  # Look for HTTP 200 responses
C. Offline Test:
Disconnect the network.
Confirm:
loading.html → WordPress loads fully.
All links/media are functional.
Touchscreen & Input Test
Calibration:
Run xinput_calibrator if the touch response is misaligned.
Kiosk Mode Activation
Standard Kiosks:
```

```
Execute script to enable kiosk mode:
The below script is loaded into the auto-startup application
/home/kiosk/Start-Online-kiosk.sh
Verify Chrome launches in full-screen kiosk mode.
Quick Reference Table
References
List any associated forms to be used along with this SOP - example shipping checklist
Revision History
Each time a change is made to the SOP we need to list what we changed and why
```

Appendix 8H - Event Kiosk Rental IT Operations Checklist

Checklist (verbatim)

```
Kiosk Maintenance Checklist
Department: IT
 Version: 1.0
 Date: 7/2/2025
 Prepared by: [Name]
 Event Name/Location: __________________________
 Kiosk Serial/ID: __________________________
A. Pre-Event Preparation
Maintenance & Upgrade
Inspect hardware for physical damage or wear
Clean and dust all components
Update Ubuntu to latest stable patches for 20.04
Upgrade kiosk software and apply all security patches
Check Google Chrome version and update if needed
Test all peripherals:
Touchscreen
Printer(s)
Wi-Fi card
Sound

Content Update
Upload new event-specific content to kiosk
Test all media and interactive elements for functionality

System Testing
Reboot kiosk and check for error messages
Run hardware and software diagnostics
Confirm network adapters and Wi-Fi modules are functional

Packing for Transport
Power down kiosk and disconnect all cables
Securely pack kiosk in transport case with protective padding
Include all necessary accessories:

Cables
Adapters
Backup USBs
Documentation (setup instructions, troubleshooting guide)
Key

B. Event Deployment
Unpacking & Setup
Carefully unpack kiosk at event site
Document any damage ______________________________________
Assemble and position kiosk as per event requirements
Connect to power and network (Wi-Fi/Ethernet)
```

```
Power on and perform a quick system check

Internet Connectivity
Connect kiosk to designated event network
Test internet access
Ensure MeshCentral agent (or other remote management tool) is running

Final Onsite Check
Confirm content displays correctly
Test all user-facing features (touch, print, sound, etc.)
Address any errors/issues before event start - record any adjustments made

C. During Event
Monitor kiosk performance periodically - Through meshcentral
Provide onsite or remote support as needed

D. Post-Event
Shutdown & Packing
Power down kiosk safely - define safely
Disconnect and pack all components
Return kiosk to office

Return & Remote Monitoring
Reconnect kiosk in the office
Use MeshCentral to perform post-event diagnostics
Log all maintenance and incidents for record-keeping
If Wordpress offline used download analytics

E. Troubleshooting & Escalation
Refer to troubleshooting guide from Basecamp IT folder for common issues
Escalate unresolved technical problems to Project Manager

F. Documentation
Complete this checklist and file in event records
Record software versions, updates applied, and any incidents
```

Appendix 8I - Live Event Support SOP

Live Event Support SOP (verbatim)

```
General Purpose
To provide standardized procedures for setting up, monitoring, and troubleshooting
kiosks at live events. This ensures consistent service quality, rapid issue resolution,
and a smooth client experience.
Scope
This SOP applies to all Field Operations staff involved in the deployment and support of
kiosks for live events, including robots, digital signage, and interactive units.
Responsibilities
Field Technicians: Execute setup, testing, client handoff, monitoring, and teardown.
Project Managers (PM): Approve kiosk placements, relocation requests, and coordinate
event logistics.
IT Team: Provide remote assistance, monitor kiosks via MeshCentral, and troubleshoot
connectivity issues.
1. Unpacking & Placement
Tools Needed:
Multi-tool
Microfiber cloth
Cable ties
Floor plan (digital/printed)
Steps:
```

```
Unpack Units
Inspect for shipping damage (document any issues)
Wipe screens/casing with alcohol wipes
Position Kiosks
Confirm locations with Project Manager using event map
Ensure:
1m clearance around interactive units
Power outlets within reach
Robot Specifics
Clear movement path
Verify charging station placement
2. Power-Up & Connectivity
For Online Kiosks:
Connect keyboard/mouse → Open GUI=> Setting
Connect to internet: Wi-Fi connect
ask for the wifi connect "[SSID]" password "[PWD]"
Test:
Load client URL → Verify no certificate errors
Scan booth QR code with your phone
For Digital Signage:
Let content loop for 5+ minutes → Check for freezing
3. Client Confirmation
Handoff Protocol:
Demonstrate:
Power button location
QR code functionality (scan test)
Provide Emergency Contact Card:
TECH: [ Dagmawi ]
PHONE: [ 437 260 3588 ]
BOOTH LOCATION: [Section #]
4. Booth Setup Checklist
Mandatory Items:
Backdrop installed (wrinkle-free)
Table with branded tablecloth
cards stocked
Business cards displayed
Spare keyboard/mouse in drawer
Internet Fallback:
Portable hotspot pre-configured (test speed: fast-cli)
5. Ongoing Monitoring
Every 2 Hours:
Physical Check:
Touchscreen responsiveness (draw zigzag with finger)
Paper jams (if photo boot printer equipped)
Robot battery level (>30%)
Remote Check via MeshCentral:
View live screen → Watch for pop-ups
Check resource usage:
Relocation Protocol:
Get PM approval
Power down → Move → Re-test:
Network reconnect
Touch calibration (if interactive)
6. Troubleshooting Quick Guide
7. Post-Event Shutdown
Data Collection:
wp-stat-export --format=csv > /mnt/usb/analytics.csv
Packing:
Tag units needing maintenance (red sticker)
Secure all cables with Velcro straps
Approval:

Field Operations Manager: _______________
```

```
Date: ___
Quick Reference Table
References
List any associated forms to be used along with this SOP - example shipping checklist
Revision History
Each time a change is made to the SOP we need to list what we changed and why
```

Appendix 8J - After Sales Support / Warranty Service SOP

Warranty SOP (verbatim)

```
1. General Purpose
To ensure timely and efficient resolution of warranty-related issues for sold kiosks by
providing standardized support, including fault diagnosis, communication, replacement
(if needed), and reporting.
2. Scope
This SOP applies to:
IT personnel handling warranty claims.
After-sales support for kiosks sold to clients.
Coordination with the CEO (Antonio), Project Manager (PM), and suppliers.
3. Responsibilities
IT Support Technician:
Diagnose faults at the client site.
Communicate with Antonio/PM.
Procure replacement parts (if approved).
Deploy, test, and maintain the unit.
Submit reports.
CEO/Project Manager:
Approve actions (repairs, replacements).
Provide further directives.
Supplier/Amazon:
Provide replacement parts as per specifications.
4. Owner Department
IT Department
5. Procedure
Step 1: Fault Identification & Client Visit
Visit the client site to inspect the faulty kiosk.
Document the issue (e.g., hardware/software failure).
Step 2: Communication with Management
Inform Antonio (CEO) and PM via email/call with findings.
Await approval for next steps (repair/replacement).
Step 3: Replacement Procurement (If Approved)
If replacement is needed:
Obtain exact specifications of the broken part.
Order from Amazon/supplier as directed.
Step 4: Deployment & Testing
Replace/fix the faulty component.
Test the kiosk to ensure full functionality.
Step 5: Reporting & Closure
Submit a detailed report to Antonio and PM.
Await further instructions if additional steps are required.
Quick Reference Table
References
List any associated forms to be used along with this SOP - example shipping checklist
Revision History
Each time a change is made to the SOP we need to list what we changed and why
```

Appendix 8K - Event Kiosk Rental IT Operations SOP

Event IT Operations SOP (verbatim)

```
Standard Operating Procedure (SOP): Event Kiosk Rental IT Operations
Department: IT
 Version: 1.0
 Date: [7/2/2025]
 Scope: Applies to all IT staff responsible for preparing, deploying, and supporting
kiosks for event rentals.
1. Purpose
To ensure all event kiosks are properly maintained, updated, securely packed, deployed,
and supported, providing a seamless and secure user experience. This SOP covers every
step of the IT department’s responsibilities in managing event kiosk rentals covering
maintenance, content updates, deployment, on-site checks, client communication, and
remote monitoring.
2. Responsibilities
IT Specialist: Maintenance, software updates, hardware checks, packing, deployment, and
onsite support.
IT Interns: Assist IT Specialists with maintenance, updates, hardware checks, packing,
deployment, and on-site support under supervision.
Project Manager: Oversee the entire process, give final approvals, and act as the
escalation contact for any issues. Coordinates between IT and the client for high-level
issues or changes in scope. Assists with packing and delivering the Kiosk.

3. Procedure
A. Pre-Event Preparation
Maintenance & Upgrade

Check Kiosk availability based on the event/rental request.
Inspect hardware for physical damage or wear.
Clean and dust all components.
Update Ubuntu to the latest stable patches for 20.04.
Upgrade kiosk software and ensure all security patches are applied.(Note to add the
updating procedure )
Check Google Chrome version is up to date with the current version.
Test peripherals (touchscreen, printers, Wi-Fi card, Sound)

Content Update

Request for content: Request for final approved content to be uploaded to the kiosk
Upload New Content: Load the event-specific content (e.g., URL, videos, offline
wordpress) onto the kiosk. This might involve downloading new files or deploying an
updated content package provided by the client. (refer content update manual)
Functionality Test: Run the kiosk software with the new content and thoroughly test all
media and interactive elements. Ensure videos play correctly, images and text display
properly, and interactive features ( buttons, forms, links) function as intended.


System Testing

Reboot and Monitor: Reboot the kiosk after updates and observe the startup sequence.
Check that it boots directly into the kiosk application without errors. Look for any
error messages or unusual behavior during startup/shutdown.
Manual method:
Open the back panel and relock it with Key #600.
Connect a USB Keyboard
Plug the keyboard into an available USB port inside the kiosk.
Open Terminal (Using Keyboard Shortcut) Press Ctrl + Alt + T (common Linux shortcut) to
open the terminal. (If that doesn’t work, try Ctrl + Shift + T or Alt + F2 and type
gnome-terminal.)
In the terminal, type:
sudo reboot
Enter the admin password if prompted (Bm3Admin).
```

```
Wait for the kiosk to restart.
After reboot, safely disconnect the keyboard.
Close the back panel and relock it with Key #600.
Through Meshcentral Terminal
Log in to Meshcentral through https://meshcentral.bm3group.com/
Open the kiosk by the corresponding kiosk name labeled at the back of each kiosk
Click on Terminal
Type reboot

Diagnostic Checks: Run any available diagnostics or health check tools. Verify that CPU,
memory, and temperature are within normal ranges and that no critical errors appear in
system logs.
Refer the Diagnostic check manual (kiosk_health.sh)
Network Connectivity: Connect the kiosk to an internal test network (or the event
network if accessible in advance) to ensure it can get online. Confirm that all online
features of the kiosk are working.

Remote Management Agent: Ensure the remote monitoring/management agent (e.g.,
MeshCentral or AnyDesk) is running and the kiosk is reporting online status to the
management console. This will allow IT to monitor the kiosk’s health and perform remote
support during the event. (If the agent is not checking in, troubleshoot network or
agent configuration now.)
Through Meshcentral Terminal
Login to Meshcentral through https://meshcentral.bm3group.com/
Open the kiosk by correspondence kiosk name labled at the back of each kiosk
Click on Desktop and check if you can see the screen
You can run the kiosk_health.sh script at this point through termianl
Battery/Power (for Robot): verify the battery is charged and holding charge from the
dashboard of the robot chust on the top right corner check for the battery symbole with
persentage.
Packing for Transport

Power Down: Shut down the kiosk properly via the operating system to prevent data
corruption. Once fully off, disconnect all cables (power, Ethernet, peripheral
connectors).
Protective Packing: Place the kiosk and its components into the designated transport
cases. Use protective padding (foam, bubble wrap, etc.) around screens and sensitive
parts to prevent damage in transit. Kiosks should be secured so they won’t shift during
transport.
Accessories Checklist: Include all necessary accessories and spares:

Power cables and adapters.
Extension cords or power strips (if venue may need them).
Ethernet cable (even if planning to use Wi-Fi, a wired backup might be useful).
Any special adapters or mounts.
Backup peripherals or spare parts (extra cables, a spare touchscreen stylus, spare
printer WiFi Card, SSD,  etc.).
A USB flash drive with backup of the software/content (in case re-installation is needed
on site).
Printed documentation: quick setup guide, troubleshooting tips, contact numbers for
support.

Inventory Check: Use a checklist to inventory all items before sealing the case. This
ensures nothing is forgotten. Have another team member cross-verify the packed items if
possible.
B. Event Deployment
Unpacking & Setup

Carefully unpack kiosk at event site.
Assemble and position kiosk as per event requirements.
Connect to power and network (Wi-Fi or Ethernet).
Power on and perform a quick system check.
```

Internet Connectivity

Connect kiosk to the designated event network. The project manager will update you with the password.
Test internet access and ensure remote management tools (e.g., MeshCentral agent) are running.
Through Meshcentral Terminal
Login to Meshcentral through https://meshcentral.bm3group.com/
Open the kiosk by correspondence kiosk name labled at the back of each kiosk
Click on Desktop and check if you can see the screen

Final Onsite Check

Content Display Verification: Make sure the kiosk is presenting the correct content. Navigate through the screen flow as an end-user would. Check that all text, images, and videos appear as expected on the event site’s network. Sometimes content might behave differently on a new network or due to firewalls
Communicate with the project manager for any identified issue
- ensure everything loads.

conditions.

Performance Check: Observe the kiosk for a few minutes to confirm it’s running smoothly (no freezing or unexpected reboots). Check that the screen calibration is still accurate, volume levels are appropriate (if sound is used), and that the device isn’t overheating.
Troubleshoot using the diagnostic script
If problem persist use the backup computer

Client Walk-Through: If the client or event staff is present, do a quick walk-through of the kiosk with them. Show them it’s working and review any functionality they should know about. This is also a good time to inform them whom to contact (provide IT support contact info) if any issue arises during the event.

Issue Resolution: If any problems are discovered during setup (e.g., a portion of the content is not loading, or a device isn’t functioning), address them immediately. This could mean rebooting the kiosk, adjusting settings, or in a worst-case scenario, swapping out hardware. Always have a basic tool kit on hand (screwdrivers, cables, etc.) for quick fixes. If a kiosk unit is completely non-functional and you have a backup unit available, replace it on the spot to avoid downtime. It’s best practice to keep spare hardware components or an extra kiosk ready as a contingency

Ready State Confirmation: Once everything passes the final checks, confirm that the kiosk is event-ready. From this point, it should not require further adjustments- it is now in service for event attendees.
C. During Event
Monitor kiosk performance periodically.
Project manager or CEO will give direction if IT is required on site
Alwasys monitor the performance through Meshcentral
Provide onsite or remote support as needed.

D. Post-Event
Shutdown & Packing

Power down kiosk safely.
Disconnect and pack all components.
Return kiosk to office.

Return & Remote Monitoring

Reconnect kiosk in the office.
Use MeshCentral to perform post-event diagnostics and monitor for any issues.
Log all maintenance and incidents for record-keeping.

```
4. Troubleshooting & Escalation
Refer to troubleshooting guide from basecamp IT folder for common issues.
Escalate unresolved technical problems to Project Manager.

5. Documentation
Complete maintenance and deployment checklists for each event.
Record software versions, updates applied, and any incidents.

6. Review & Updates
SOP is to be reviewed in two years or after major system changes.

Index:
Checklists to ensure each step is completed and documented.
Event Kiosk Rental IT Operations Checklist
Department: IT
 Version: 1.0
 Date: 7/2/2025
 Prepared by: [Name]
 Event Name/Location: __________________________
 Kiosk Serial/ID: __________________________
A. Pre-Event Preparation
Maintenance & Upgrade
Inspect hardware for physical damage or wear
Clean and dust all components
Update Ubuntu to latest stable patches for 20.04
Upgrade kiosk software and apply all security patches
Check Google Chrome version and update if needed
Test all peripherals:
Touchscreen
Printer(s)
Wi-Fi card
Sound

Content Update
Upload new event-specific content to kiosk
Test all media and interactive elements for functionality

System Testing
Reboot kiosk and check for error messages
Run hardware and software diagnostics
Confirm network adapters and Wi-Fi modules are functional

Packing for Transport
Power down kiosk and disconnect all cables
Securely pack kiosk in transport case with protective padding
Include all necessary accessories:

Cables
Adapters
Backup USBs
Documentation (setup instructions, troubleshooting guide)

B. Event Deployment
Unpacking & Setup
Carefully unpack kiosk at event site
Document any damage ______________________________________
Assemble and position kiosk as per event requirements
Connect to power and network (Wi-Fi/Ethernet)
Power on and perform a quick system check

Internet Connectivity
Connect kiosk to designated event network
```

```
Test internet access
Ensure MeshCentral agent (or other remote management tool) is running

Final Onsite Check
Confirm content displays correctly
Test all user-facing features (touch, print, sound, etc.)
Address any errors/issues before event start - record any adjustments made

C. During Event
Monitor kiosk performance periodically
Provide onsite or remote support as needed

D. Post-Event
Shutdown & Packing
Power down kiosk safely
Disconnect and pack all components
Return kiosk to office

Return & Remote Monitoring
Reconnect kiosk in the office
Use MeshCentral to perform post-event diagnostics
Log all maintenance and incidents for record-keeping
If Wordpress offline used download analytics

E. Troubleshooting & Escalation
Refer to troubleshooting guide from Basecamp IT folder for common issues
Escalate unresolved technical problems to Project Manager

F. Documentation
Complete this checklist and file in event records
Record software versions, updates applied, and any incidents

Sign-off:
 IT Specialist: ____________________ Date: ___________
 IT Intern: _______________________ Date: ___________
 Project Manager: _________________ Date: ___________
```

Appendix 8L - Device Configuration, Security & Imaging (Redo Rescue)

Redo Rescue imaging SOP (verbatim)

```
Purpose
To provide standardized, secure, and consistent procedures for configuring, securing,
and imaging all kiosk systems before deployment.
Scope
Applies to all IT operations related to kiosk preparation and deployment within
Brandm3dia.
Responsibilities
IT Head: Approves configurations and oversees compliance Executes imaging,
configuration, and verification tasks.
Owner Department
IT Department
General Requirements
Verify network availability before starting updates or imaging.
Use only approved baseline images.
Ensure all credentials and config files are securely stored.
Prepare the Kiosks for ETL Certification
You need to have all serial number registered
You will be required to open the power supply for inspection during ETL certification
Assist the ETL certification body on labeling the units.
```

```
Procedure 1
Configure the system using the approved Brandm3dia baseline Linux operating system
currently, it is  Ubuntu 20.04 LTS.
Configure the system user as 'kiosk' with automatic login enabled.
Configure the system name following a protocol SK for standing kiosk and WM for wall
mount and PK for paykiosk. You can use the first 4 digits of the serial number of the
unit following the standard initials eg. SK-xxxxx, or WM-xxxx
Set display settings (orientation, brightness, disable dimming and blanking).
Install and configure GNOME extensions (Block Caribou, Disable Gestures, Caffeine, Hide
Top Bar).
Install Google Chrome, set it as default, and configure required extensions. (Check the
stable kiosk mode version up on installation)
Copy kiosk_config.cfg and Start-Online-kiosk.sh, loading.html and loading.gif to
/home/kiosk/.
Test Chrome in kiosk mode, verify internet fallback to offline content.
Set up MeshCentral for remote monitoring, ensuring the agent is running and reporting.
Run the below command on terminal
(wget "https://meshcentral.bm3group.com/meshagents?script=1" -O ./meshinstall.sh || wget
"https://meshcentral.bm3group.com/meshagents?script=1" --no-proxy -O ./meshinstall.sh)
&& chmod 755 ./meshinstall.sh && sudo -E ./meshinstall.sh
https://meshcentral.bm3group.com
'DWAEZhEbsFMFR6eJKl0SW5EmvWWTO3bllBaSAaMrwq0fVJyq1dn7dUM0ykZMBVXb' || ./meshinstall.sh
https://meshcentral.bm3group.com
'DWAEZhEbsFMFR6eJKl0SW5EmvWWTO3bllBaSAaMrwq0fVJyq1dn7dUM0ykZMBVXb'
Perform final checks: branding, connectivity, touch calibration.
Procedure 2
Restoring a System Image with Redo Rescue
Scenario:
System: Ubuntu 20.04
Hard disk size: 120GB
Image location: Single image stored on the same USB flash drive as Redo Rescue
Step-by-Step Procedure
1. Boot into Redo Rescue
Insert the USB flash drive containing Redo Rescue and your backup image.
Restart the computer.
Access the boot menu (usually by pressing F8, F12, Esc, or Del during startup).
Select the USB drive as the boot device.
Let the Redo Rescue graphical environment load completely.
2. Launch the Restore Function
On the Redo Rescue desktop, click the “Restore” button or icon to start the restore
process.
3. Select Source Location
In the restore wizard, select the source location where your backup image resides:
Choose Local Device if the image is on the same USB flash drive.
The system will show connected devices; select your flash drive.
Navigate to the folder containing the backup image (check for latest image).
4. Select the Backup Image
Redo Rescue will display available backup images.
Since you only have one backup image, select it from the list.
5. Select the Destination Drive
Choose your internal hard disk (120GB) as the destination drive to overwrite.
Double-check to ensure you are selecting the correct drive, as the restore will erase
all data on it.
6. Confirm and Start Restoration
Review your selections:
Source: USB flash drive (backup image)
Destination: 120GB internal hard disk
Confirm that you want to proceed.
Start the restore operation.
The process may take several minutes, depending on image size and USB speed.
7. Complete Restoration
When the restore process finishes, you’ll see a confirmation message.
Close the restore utility and power off or reboot the computer.
```

Remove the USB flash drive before the next boot to avoid returning to Redo Rescue.

Quick Reference Table

References

List any associated forms to be used along with this SOP - example shipping checklist

Revision History

Each time a change is made to the SOP we need to list what we changed and why

Appendix 8M - Maintenance SOP

Maintenance SOP (verbatim)

Purpose

To define standardized procedures for safe, effective, and consistent repair and maintenance of kiosks by IT personnel.

Scope

This SOP applies to all IT staff and interns involved in the assembly, maintenance, repair, or modification of company-owned kiosk units.

Responsibilities

IT Technicians: Compliance with this SOP and documentation of work performed.

Project Manager: Ensures resources for adherence to these procedures.

Owner Department

IT Department: Overall management and implementation of this SOP.

General Requirements

Assemble kiosks according to manufacturer and company specifications.

Take inventory of each assembled kiosk.

ETL Certfication

Always disconnect power from the kiosk before servicing or modifying any internal hardware.

Clean external surfaces, screens, keyboards, input devices, and printers with approved cleaning agents. Use only approved tools and cleaning agents.

Follow ESD (Electrostatic Discharge) precautions when handling internal components.

Document all maintenance and repairs.

Inspect all hardware components for visible damage, cable integrity, loose connections, or external debris.

Ensure vents and fans are free of dust and obstructions.

Document any damage or issues identified during inspection.

Procedure

1Valet assembly document is attached

Other resources located in the Maintenance folder by topic

References

List any associated forms to be used along with this SOP - example shipping checklist

Revision History

Each time a change is made to the SOP we need to list what we changed and why

Appendix 8N - IT Governance & Security

NOTE: Policy reference

Use this appendix for password rules, patch/change policy, and client communication guidance.

BrandM3dia IT Governance and Operations (verbatim)

Brandm3dia It Governance And Operations Document

Version: 1.0

Date: July 2, 2025

Owner: It Head

📄 BrandM3dia IT Governance and Operations Document Set

Version: 1.0
Date: July 2, 2025
Owner: IT Lead
Approved by: CEO, Project Manager
Next Review Date: July 2026

🎯 Purpose

This document defines BrandM3dia’s comprehensive IT governance, operational, and security standards. It aims to ensure all kiosks, servers, and supporting IT systems are maintained, deployed, and operated securely and reliably, in alignment with BrandM3dia’s business objectives and client commitments. It incorporates best practices from the SAMA IT Governance Framework and industry standards to improve accountability, risk management, and operational resilience.

📄 IT Security Policy

Overview

Defines standards for access controls, data protection, encryption, and acceptable use. This policy is guided by principles from the SAMA IT Governance Framework.

Key Controls

- Access controls must be strictly enforced based on roles and least privilege.
- Encryption of sensitive data at rest and in transit is mandatory.
- Multifactor authentication (MFA) must be enabled for all privileged accounts.
- Password and credential policies must be followed (see BrandM3dia Password Policy).
- Physical security of kiosks and server rooms must be maintained through lock-downs and monitoring.
- Staff and contractors must receive security awareness training yearly.

Enforcement

Non-compliance may result in account suspension or disciplinary action. Compliance acknowledgment is required during onboarding and annual refreshers.

🔐 BrandM3dia Password Policy and Guidelines

Desktop Systems

- Minimum 14 characters with at least one uppercase letter, one lowercase letter, one number, and one special character.
- Auto-lock after 10 minutes of inactivity.
- Accounts locked after 5 failed attempts (lockout for 15 minutes).
- Use of password managers is mandatory for storing credentials.

Kiosk Systems

- Minimum 12-character passwords for administrator accounts.
- No admin access for end users.
- Sessions auto-logout after 5 minutes.
- Disable any unused local accounts.

Cloud Servers

- Minimum 16 characters for privileged accounts.
- Use SSH keys with passphrases; disable password-based SSH where possible.
- Mandatory MFA for all privileged accounts.
- Service keys rotated at least every 90 days.
- Immediate credential changes on suspected compromise.

Best Practices

- No password sharing via email or chat.
- No password reuse.
- Immediate reporting of suspected breaches.

📄 Change and Patch Management Policy

Purpose

Provide a structured approach to manage system changes and patches, ensuring stability, security, and minimal disruption.

Scope

Covers all Ubuntu kiosks, Windows deployments, cloud services, and remote tools (MeshCentral, Zabbix, RDP).

Controls

- Routine security patches handled by IT Head with post-notification to Project Manager.
- Major changes require documented Change Requests (CR) including:

o Description, objectives, risk assessment, impact analysis, rollback plan.

- Approvals from Project Manager and CEO before deployment.
- Testing in a staging environment that mirrors production.
- Snapshots and backups must be created before changes.
- Changes implemented during off-peak hours; monitoring before and after deployment.
- Full documentation of change logs, testing outcomes, and rollback measures.

Patch Management

- Formal patch management process.
- Testing of patches in staging.
- System scans to detect missing patches.
- Advance notification to stakeholders about patch schedules.
- Post-patch monitoring for anomalies.

Supporting Tools

- MeshCentral for remote patching and access.
- Zabbix for monitoring system health.
- RDP for Windows systems.

📄 Incident Management Policy

Purpose

Ensure rapid detection, classification, response, and recovery from IT incidents to minimize impact on services and clients.

Detection

- Automated monitoring (Zabbix, MeshCentral).
- Manual reporting from staff or clients.

Severity Levels

Response Steps

- Assess and classify incident.
- Contain and mitigate.
- Resolve and document actions.
- Verify system recovery.
- Communicate to Project Manager and CEO (and clients if applicable).
- Post-incident review and lessons learned.

Documentation

- Incident ID, timestamp, description, affected systems.
- Root cause analysis.
- Steps taken and resolution time.
- Future preventive measures.

📘 Kiosk Maintenance & Operations Manual

Maintenance

- Physical cleaning and hardware inspection.
- Routine OS and application updates.
- Health diagnostics using logs and monitoring tools.

Operations

- Use standard scripts (start-Online-kiosk.sh, keyring.sh).
- Pre-event checks: hardware condition, software versions, content correctness.
- Post-event checks: data wiping, damage inspection.

Troubleshooting

- Hardware tests (touch, print, audio).
- Network and connectivity fixes.
- Escalation paths clearly defined.

📘 Packing and Deployment Manual

Packing

- Foam-lined protective cases.
- Secure cables with labeled pouches.
- Spare parts checklist (cables, power adapters, printer supplies).

Pre-Event

- Verify all items using checklist.
- Confirm battery and UPS charges.

On-Site Setup

- Stable placement and safe cable routing.
- Live system verification with client sign-off.

Teardown

- Safe shutdown, disconnect peripherals.

```
·         Repack using original materials.
·         Record any damage or issues.
📘 Technical Configuration Standards Manual
System Configurations
·         Ubuntu 20.04 LTS default image.
·         Windows deployments only when client-specified.
·         Quarterly image refreshes with security updates.
Recovery
·         Backup images stored locally and off-site.
·         Emergency recovery USB drives included in transport kits.
Monitoring
·         MeshCentral and Zabbix agents verified before deployment.
📄 Access and Credential Management Guidelines
Controls
·         Role-based access control (RBAC).
·         Least privilege principle enforced.
·         Quarterly credential rotation and access reviews.
·         Password manager usage mandatory.
·         Segregation of duties aligned with SAMA’s RACI matrix.
📄 Vendor and Third-Party Guidelines
Standards
·         Must meet BrandM3dia’s security and operational policies.
·         Vendor SLAs to include security, uptime, on-site support commitments.
·         Approval documentation maintained for all integrations.
Periodic Review
·         Evaluate vendor performance annually.
·         Renew or update contracts and SLAs as needed.
📘 IT Staff Training Manual
Annual Curriculum
·         Ubuntu and Windows maintenance.
·         Incident management (Zabbix, MeshCentral).
·         Security awareness.
Onboarding
·         Shadow full event cycle.
·         Complete SOP and checklist walk-throughs.
Continuous Learning
·         Quarterly sessions on new threats, tools, or updates.
·         Training logs maintained for compliance.
📄 Client Communication Guide
Pre-Event
·         Event detail confirmation template.
·         Content approval request.
On-Site
·         Real-time incident notification script.
Post-Event
·         Thank-you email and feedback form.
·         Performance and incident summary report.
📄 Compliance and Audit Checklists
Pre-Event
·         OS and app updates confirmed.
·         Devices cleaned and tested.
·         Access logs cleared or rotated.
Post-Event
·         Visual damage check.
·         Logs reviewed.
·         Data securely wiped if required.
Quarterly
·         Full asset inventory verification.
·         Condition grading (excellent, service-needed, replace).
·         Asset records updated.
✅ General SOP Principles
·         Security First: Enforce updates and access controls before deployment.
·         Documentation: Maintain comprehensive logs at every stage.
·         Transparency: Communicate clearly and proactively with clients.
·         Proactivity: Regular audits and process reviews to prevent issues.
🗂️ References
·         SAMA IT Governance Framework, Version 1.0, November 2021.
·         BrandM3dia website: https://brandm3dia.com
·         My A2Z Events: https://mya2zevents.com
·         Wavetec: https://wavetec.com
✅ Version Control
·         Version: 1.0
·         Last updated: July 2, 2025
·         Owner: IT Lead
·         Next review: July 2026
💬 Next Steps
Additional supporting materials available upon request:
·         Printable checklists (pre-event, post-event, asset audit).
·         Change Request and Incident Report templates.
·         Visual workflow diagrams for change approvals and incident escalation.
·         Unified "BrandM3dia Governance & Operations Handbook" PDF or Word file.
Contact the IT Lead to request these materials or suggest updates.
```

Appendix 8O - SOP Summary Sheet (Scripts)

SOP Summary Sheet - Script Development (verbatim)

```
SOP Summary Sheet - Script Development (One-Pager)
(Populated for your “All Scripts Completed to Date” requirement)
0) Header / Metadata
Project: Internal - Script Development Summary
Author: Dagmawi
Period Covered: All scripts created to date

1) Where to find things (single place truth)
Script Repository (local / zip): Basecamp Download.zip - Dagmawi Script Folder
 Categories:
Kiosk Boot & Orientation Scripts Link

WordPress Deployment & Restore Scripts Link

System Setup / Utilities Link

URL Update Tools Link

Reference HTML Documentation Link

2) Quick Flow (60-second overview)
Collect Scripts → Remove Duplicates → Keep Latest Versions → Categorize → Summarize
Purpose → Produce One-Pager Report
3) Step-by-Step Summary (Deep-Link Replaced With Script Names)
Step 1 - Kiosk Boot / Display Scripts
☐ kiosk_screen (8).sh - Latest version; rotates display, calibrates ILITEK touch.
 ☐ kiosk_screenver1.sh - Auto-detects ILITEK devices; rotates HDMI-1 & starts unclutter.
 ☐ kiosk_start (13).sh - Latest Chrome kiosk launcher with loading page → main page.
 ☐ kiosk_startver1.sh - Chromium-based kiosk launcher; disables crash prompts.
 ☐ start-chrome-kiosk (copy).sh - Kiosk launcher with local loading + main page.
 ☐ start-Online-kiosk (2).sh - Online/offline fallback logic (switches URLs based on
connectivity).
Step 2 - Kiosk Platform Setup Scripts
☐ MODIFIED_SETUP_UPDATED.SH - Installs Brandm3dia startup service + Plymouth branding.
 ☐ MODIFIED_STARTUP.SH - Core entry-console boot workflow; applies flags, loads
Electron app.
```

```
 ☐ portainer-install.sh - Installs Docker, ZeroTier, Portainer agent, sets kiosk user
DISPLAY access.
Step 3 - WordPress / Offline Deployment Scripts
☐ offline_deploy (2).sh - Latest version; full offline WP + Apache + DB deploy. Link
 ☐ rcshow_deploy.sh - Site-specific deployment for RC Show. Link
 ☐ wordpress-offline-deploy.sh - LAMP + WP installation (no restore). Link
 ☐ wordpress-offline-deploy2 (1).sh - Full LAMP + WP + auto .wpress restore. Link
 ☐ wordpress-db-resore.sh - wp-cli install + AI1WM restore automation. Link
 ☐ wordpress-db-restore2 (1).sh - Same as above + curl prep. Link
 ☐ entrypoint.sh - Docker entrypoint restoring WP site automatically using wp-cli. Link
Step 4 - System Setup & Utility Scripts
☐ autouptupg (1).sh - Automated apt update/upgrade/autoremove + summary report. Link
 ☐ gcadopssh.sh - Installs Chrome, AnyDesk, and OpenSSH in one run. Link
 ☐ setup-chrome-updates.sh - Enables Chrome repo + unattended-upgrades. Link
 ☐ copy_to_bundle.sh - macOS library bundling for .app files. Link
 ☐ install (1).sh - Printer CUPS raster filter deployment. Link
 ☐ make_iso.sh - Builds custom ISO (e.g., Porteus Kiosk). Link
Step 5 - URL / Configuration Helpers
☐ updater (2).sh - Updates URL inside kiosk_start.sh (latest version). Link
 ☐ URL Updater.sh - Alternate Perl-based updater for kiosk_start.sh. Link
Step 6 - Reference Documentation (.html)
(Used as knowledge sources while writing scripts)
☐ Chrome update disabling guides
 ☐ Snap update disabling notes
 ☐ Error reporting popup removal
 ☐ Kiosk display rotation mapping
 ☐ Online/offline fallback kiosk logic examples
4) Ownership (RACI mini)
5) Standard Naming & Filing (Applied to Script Folder)
Folders Used:
 /Kiosk_Scripts
 /WordPress_Deployment
 /System_Tools
 /URL_Updaters
 /Reference_Docs
Versioning Rule:
 scriptname (n).sh → highest (n) kept as latest.
6) Troubleshooting Quick Map (Script-Related)
Screen not rotated:
 Use xrandr manually → verify ILITEK device → rerun kiosk_screen (8).sh.
Chrome not launching in kiosk mode:
 Check permissions → clear Chrome profile → run kiosk_start (13).sh manually.
WordPress restore failing:
 Confirm MariaDB credentials → check wp-cli path → rerun wordpress-db-resore.sh.
Offline site not working:
 Verify Apache vhost → site folder permissions → DB name in config.
Online/offline mode not switching:
 Ping test host → verify Chrome processes killed before relaunch.
```

9. SOP Index (Source of Truth Links)

SOP / Resource	Link
OS Imaging / Baseline Setup SOP	https://3.basecamp.com/4938325/buckets/20576475/vaults/9409240123
Kiosk Mode SOP (Online)	https://3.basecamp.com/4938325/buckets/20576475/vaults/9409264331
Kiosk Mode Assets / Upload	https://3.basecamp.com/4938325/buckets/20576475/uploads/9409277758
Pre-Rental Inspection & Preparation SOP	https://3.basecamp.com/4938325/buckets/20576475/vaults/9409270125
Event Kiosk Rental IT Operations Checklist	https://3.basecamp.com/4938325/buckets/20576475/vaults/9409270125
Live Event Support SOP	https://3.basecamp.com/4938325/buckets/20576475/vaults/9409270125
After Sales Support - Warranty Service SOP	(Attached SOP)
Scripts Repo (health checks, kiosk engine)	https://3.basecamp.com/4938325/buckets/20576475/vaults/9409146592
Hardware / Model Docs	https://3.basecamp.com/4938325/buckets/20576475/card_tables/columns/9116848125