DS Site Kiosk Guide

Setting up and running a public sign-in kiosk at the site gate

DS Site · Deep Space Horizon · October 2026

1. What a site kiosk is

A site kiosk is a tablet at the site gate showing one **public DS Site page**. The page has four large buttons, and each one opens an existing form:

- **Site Induction**: completed once per site by each new worker.
- **Site Sign In**: completed by inducted workers when they arrive each day.
- **Sign Off**: completed by everyone (workers and visitors) when they leave.
- **Visitor Sign In**: completed by visitors, who don't need an induction.

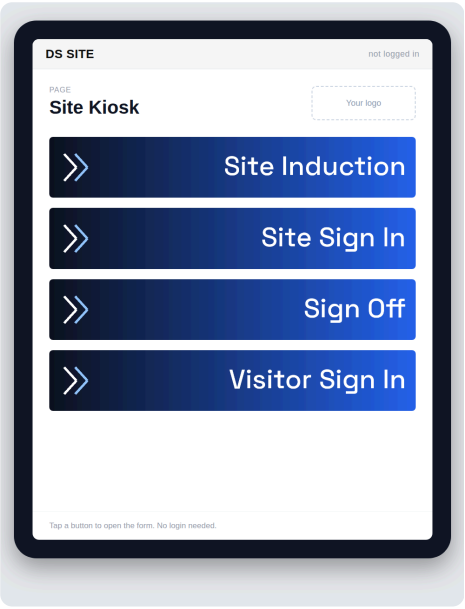


Illustration: the kiosk page on a logged-out tablet. Your channel's own button artwork and logo replace the placeholders.

Nobody logs in. The tablet stays logged out, and every form on it is filled in anonymously, so one tablet serves everyone who comes through the gate. Sign-ins land in Horizon's **Site Sign On** register in the same way as sign-ins made from a personal phone.

Under the hood, the kiosk is an ordinary DS Site **template plus page**. The template has four sections, and each section holds a single *Form* field with a fixed value (the form to open) and a button image. This is the same pattern as a channel's Site Sign On page. Fixed Form references keep working when the page is saved for offline use in the DS Site app.

2. How it fits together

Button	Opens	What happens to the submission
Site Induction	The site's induction form: the standard `site_induction` form or a site-specific copy	Stored as the worker's induction for that site.
Site Sign In	Sign-in form, alias `signing_in`	A worker row in Site Sign On, matched to their induction.
Sign Off	Sign-out form, alias `signing_out`	Closes that person's open sign-in for the day.
Visitor Sign In	Visitor form, alias `visitor_signing_in`	A visitor row in Site Sign On, with induction shown as *Not required*.

The form aliases matter. Sign-in, visitor sign-in and sign-out submissions are picked up by the **Deep Space Form Integration** webhook, and its `site-sign-in` module handles exactly these three aliases: `signing_in`, `visitor_signing_in` and `signing_out`. Horizon keys every sign-in on the person's **mobile number**, not on a DS Site account. That's why anonymous kiosk submissions work.

> **The kiosk is an addition, not a replacement.** The QR codes on Horizon → HSEQ → Site Sign On still work for people who'd rather sign in on their own phone. Both routes write to the same register.

3. Before you start

- **The four forms exist in the channel and are activated.** On each one, *login required* is off, *restricted* is off, and there's no access code.
- **Visitor Sign In forces anonymous submission** and has the alias `visitor_signing_in`. If the channel doesn't have one yet:
  1. Copy the Sign In form, which keeps its field aliases.
  2. Set the alias to `visitor_signing_in`.
  3. Remove the induction dependency.
  4. Turn on anonymous submission.
  5. Keep the Project field.
- **The Deep Space Form Integration webhook is active for the channel**, so sign-in submissions reach Horizon. Check this with the Deep Space team if Site Sign On isn't already in use.
- **You have channel administrator access:** a DS Site admin login for the manual method, or the channel's admin API key for the builder script.
- **The channel has button artwork**: see section 5.

4. Creating the kiosk

Option A: the builder script (recommended)

The DS Site tooling includes a kiosk builder. Each kiosk lives in its own project folder, `page-edit-and-create/page-projects/<slug>/`, which holds `build_site_kiosk.py` and a README. Start a new kiosk by copying an existing kiosk project folder.

1. **Set the channel details** at the top of `build_site_kiosk.py`:
   - the channel alias;
   - the default form id for each of the four buttons;
   - the master id of the channel's Site Sign On template, which the script copies the Form field type from;
   - the button artwork file.
2. **Add the API key.** Put it in `keys.env` as `DSSITE_TARGET_API_KEY=<key>`. That file is git-ignored, so the key is never committed.
3. **Dry run:** `python build_site_kiosk.py`. It checks that each form can be read publicly and is activated, with login off, restricted off and no access code. It renders the button images and writes `mappings/plan.json`. If any form fails a check, it stops and lists the problem. Nothing is written to the channel.
4. **Apply:** `python build_site_kiosk.py --apply`. It:
   - uploads the button images as channel resources, or reuses them;
   - creates the *Site Kiosk* template and page;
   - activates the page;
   - checks each button points at the right form, and that the page can be read without logging in.

   The page id and its public URL are written to `mappings/result.json`.

Option	Use it to
`--induction-form <id>`	Choose which induction form the Site Induction button opens, for example a site-specific induction instead of the standard one.
`--apply`	Write to the channel. Without it, the script only checks and plans.
`--api-key <key>`	Pass the key directly instead of using `keys.env`.

Safe to re-run. The script finds the template (`site_kiosk_template`) and page (`site_kiosk`) by alias. If they exist, it backs up the live versions and upgrades them in place, so you never get a second kiosk. Use a re-run to change a button's form, swap artwork or repair the page. Every call is recorded in the project folder:

- `downloaded-jsons/`: backups taken before each change, and copies of the result after it;
- `uploaded-jsons/`: exactly what was sent;
- `response-jsons/`: what DS Site returned.

Option B: manually in DS Site admin

For support staff without the script. The quickest route is to **copy the channel's Site Sign On template**, which already uses this pattern, then adjust it:

1. **Admin → Resources:** upload the four button images and activate each one.
2. **Admin → Templates:** create (or copy) a template named *Site Kiosk* with four sections: Site Induction, Site Sign In, Sign Off and Visitor Sign In.
   - Give each section one *Form* field. Set its fixed value to the target form and its button image to the matching resource.
   - Hide section names, so only the buttons show.
   - Leave login required, restricted and access code **off**.
3. **Admin → Pages:** create a page from the template, with the same public settings (no login, not restricted, no access code). Save it, then **activate** it.
4. The public URL is `https://app.qin.world/<channel>/pages/<page id>`.

> **Point at master forms.** Make sure each Form field points at the form's current master version, not an archived copy, or the button will open an old version. If you later change a button image, re-save the template and page. Pages store a copy of the resource as it was when they were saved.

5. Button artwork

Match the channel's existing menu buttons, so the kiosk looks like the rest of the site app. The builder takes one existing button image from the channel and **re-letters** it:

- It keeps the logo and colour gradient.
- It paints over the old text area, extending the gradient across it.
- It writes the new label in white in the DS Site font (Space Grotesk, Medium), right-aligned and scaled to fit.

This is the same method used for site-specific induction menus. If the channel already has a button for one of the four actions (often Site Induction), reuse that resource instead.

6. Testing before go-live

1. Open the public page URL in a **private or logged-out browser window**. The page must load with no login prompt.
2. Tap each button and check it opens the right form, with no login and no “join channel” prompt.
3. Submit **one test submission per form**, named “TEST”, using a test mobile number. Pick the real project.
4. In Horizon, open **HSEQ → Site Sign On** for that project. Check that:
   - the test worker sign-in appears;
   - the visitor row shows induction as *Not required*;
   - Sign Off closes the matching sign-in.
5. Archive the test submissions afterwards, so they don't show in the site's records.

7. Using the kiosk on site

Setting up the tablet

- Mount the tablet at the gate with power and a reliable connection (Wi-Fi or mobile data).
- Open the kiosk's public URL and save it to the home screen. If the connection is patchy, make the page and its forms available offline in the DS Site app. Submissions are then held on the tablet and sent when it's back online.

> **The tablet must stay logged out.** Sign In, Sign Off and Induction forms that don't force anonymous submission record the logged-in user as the submitter. A logged-in kiosk would record every gate submission as that one person. If someone logs in on the tablet, log out again. As a safeguard, a channel admin can turn on *force anonymous submission* on those forms too. Only do that if nobody needs to submit them while logged in elsewhere.

Everyday use

Who	What they do
New worker	Tap *Site Induction* and complete it once for this site, then *Site Sign In*.
Inducted worker	Tap *Site Sign In* on arrival each day and *Sign Off* when leaving.
Visitor	Tap *Visitor Sign In* on arrival (no induction needed) and *Sign Off* when leaving.
Site manager	Check who's on site in Horizon → HSEQ → Site Sign On, which updates as people sign in and off.

Ask everyone to use the same mobile number every time. Horizon matches inductions, sign-ins and sign-offs by mobile number. If someone uses a different number, they show up as a new person.

8. Limitations and what's next

- **No “must be inducted” check on the kiosk.** DS Site's form dependencies (for example, “must have completed the induction to sign in”) are tied to the logged-in account. Anonymous kiosk submissions have no account, so the dependency can't be checked on the tablet. Horizon still shows each worker's induction status against their sign-in.
- **Returning workers retype their details.** A returning-worker lookup has been proposed but not built: the worker enters their mobile and surname, the lookup runs on the server, and the sign-in form opens already filled in. It would be switched on per workspace.
- **Never put a lookup field for people in a public form.** A lookup field publishes its whole CSV to anyone who opens the form, which would expose every worker's details.
- **Filled-in links match on field labels.** Links that open a form already filled in match fields by their labels. Renaming a label (for example “First Name”) breaks those links.

9. Troubleshooting

Problem	Check
A button asks for a login	The target form has login required on, is restricted, or has an access code. Fix the form, then run the builder's dry run: it lists any form that isn't public.
The page itself asks for a login	The template or page isn't public, or the page isn't activated. Re-run the builder with `--apply`, or fix the settings in Admin → Pages.
A button opens an old version of a form	The Form field points at an archived version. Re-run the builder, which always uses the master form, or re-save the template and page.
A submission doesn't appear in Site Sign On	Check: • the form alias is exactly `signing_in`, `visitor_signing_in` or `signing_out`; • a project and mobile number were entered; • the Form Integration webhook is active for the channel. Allow a minute for it to arrive.
Submissions are recorded against one person	The tablet is logged in. Log out (see section 7).
A new button image doesn't show	Pages keep the image as it was when they were saved. Re-run the builder, or re-save the template and page.
Offline: forms don't open or submissions don't arrive	Make the page and all four forms available offline in the DS Site app while connected. Queued submissions send when the tablet reconnects. After changing the kiosk, refresh the offline copy.