

Complete EU Compliance Bundle

Six Practical Guides for Small Web Agencies

Version 1.0 · August 2026
A Clean Copy Publication · Mahope

Free edition — licensed under Creative Commons BY-NC 4.0

Contents

NIS2 Compliance for Small Web Agencies

A complete, plain-English guide for agencies with 1–50 employees: does NIS2 apply to you, the security measures that count, incident reporting templates, and a day-by-day 30-day compliance plan.

GDPR Compliance for Small Web Agencies

Controller vs processor roles explained in plain English, the three documents that matter (DPA, RoPA, incident plan), data subject rights handling, and a 14-day action plan with ready-to-paste DPA clause language.

EAA Compliance Checklist for WordPress Sites

A 10-point accessibility checklist covering color contrast, alt text, headings, keyboard navigation, screen readers, forms, multimedia and zoom — plus testing tools and a 14-day fix plan.

EAA Compliance for Shopify Stores

Shopify-specific EAA guidance: product image alt text systems, theme accessibility (contrast, focus, keyboard), checkout and forms, the required accessibility statement.

Cookie Consent & Privacy Compliance for Small Websites

What the law actually requires, how to audit your site for cookies and tracking, building a compliant consent banner, consent record-keeping, privacy policy structure under Articles 13–14.

Build Your First Chrome Extension

A step-by-step, hands-on guide that takes you from "I have never built an extension" to live in the Chrome Web Store. Manifest V3, popups, context menus, storage, options pages and publishing.

NIS2 Compliance for Small Web Agencies

From: Complete EU Compliance Bundle — Version 1.0

NIS2 Compliance for Small Web Agencies

A Practical Guide to Meeting EU Cybersecurity Requirements Without a Compliance Team

Version 1.0 -- August 2026

Foreword

If you run a small web agency in the EU, this guide is for you.

NIS2 (the EU's Network and Information Security Directive 2) became enforceable in 2025. Unlike the original NIS directive, NIS2 casts a much wider net -- it covers more sectors, more company sizes, and imposes stricter requirements. Most significantly for you: digital service providers, managed service providers, and their subcontractors are explicitly in scope.

This is not a legal document. It is a practical field guide written by someone who works with agencies like yours. Every recommendation has been tested with real agencies. You will not find abstract theory here -- only what works.

Let's get you compliant.

Chapter 1: Does NIS2 Apply to Your Agency?

This is the first question, and it's the one most agencies get wrong. Many assume they are too small to be in scope. Under NIS2, that assumption can be costly.

The Scope Test

NIS2 applies to medium-sized and large enterprises in covered sectors. The size thresholds are:

- Large enterprise: 250+ employees OR €50M+ annual turnover AND €43M+ balance sheet
- Medium enterprise: 50-249 employees OR €10M-€50M turnover
- Small enterprise: Below 50 employees AND below €10M turnover

Good news: If your agency has fewer than 50 employees and less than €10M annual turnover, you are a "small enterprise" and generally NOT directly in scope for NIS2.

Bad news: You still need to read this guide. Here's why.

Why Small Agencies Must Still Comply

- Your clients are in scope. If you serve medium or large enterprises that ARE subject to NIS2, they must assess their supply chain -- and that includes you. Your security posture becomes their problem.
- Contractual requirements. Enterprise clients are already adding NIS2 clauses to their vendor contracts. If you cannot demonstrate basic compliance, you lose the contract.
- Downstream liability. Under NIS2 Article 21(2)(c) on supply chain security, your clients are required to assess the security practices of their suppliers. If you are the weak link, they will be penalized for it -- and they will not hesitate to pass that liability to you.
- Competitive disadvantage. Agencies that can demonstrate NIS2 alignment win more pitches. It is becoming a differentiator.

Quick Self-Assessment

Answer these three questions:

- Do any of your clients have more than 50 employees? → If yes, they are likely in NIS2 scope.
- Do you handle any of the following for clients: hosting, server management, DNS, email infrastructure, SSL/certificate management, backup services, or security monitoring? → If yes, you are a digital service provider in their supply chain.
- Do you have access to your clients' networks, data, or administrative systems? → If yes, you are a vector they must secure.

If you answered yes to any of the above, you need to be NIS2-ready -- even if the directive does not name you directly.

Important vs. Essential Entities

NIS2 divides in-scope entities into two categories:

Category	Threshold	Fines	Example
-----	-----	-----	-----
Essential	250+ emp OR €50M turnover	Up to €10M or 2% of global turnover	Telecoms, energy, banking
Important	50+ emp OR €10M turnover	Up to €7M or 1.4% of global turnover	Digital services, hosting, managed services

As a small agency, you will likely never be an Essential or Important entity. But your clients will be -- and they will require you to meet equivalent security standards.

Verdict: Read this guide, implement the measures, and document everything. You will not face direct NIS2 fines, but you will face client contract losses if you are not prepared.

Chapter 2: The 10 Essential Security Measures (Article 21)

Article 21 of NIS2 requires in-scope entities to implement "appropriate and proportionate technical, operational, and organisational measures." The directive lists 10 categories. Below, I translate each into practical actions for a small agency.

2.1 Risk Analysis and Security Policies

What NIS2 says: Entities must perform risk analyses and establish security policies.

What to do:

- Write a one-page Information Security Policy covering: acceptable use, password requirements, device security, remote work rules.
- Review it annually (set a calendar reminder).
- Store it in a shared location all employees can access.

Template: A sample Information Security Policy is included in the appendices.

2.2 Incident Handling

What NIS2 says: Entities must have processes for detecting, handling, and reporting incidents.

What to do:

- Designate one person as Security Contact (can be you or a team member).
- Create a simple Incident Response Plan in 6 steps: Detect → Triage → Contain → Eradicate → Recover → Review.
- Use a free monitoring tool (e.g., UptimeRobot, Better Stack) to detect outages.
- For WordPress agencies: install a security plugin (e.g., Wordfence, iThemes Security) and enable real-time monitoring.

2.3 Business Continuity and Backup Management

What NIS2 says: Entities must ensure business continuity, including backup management and disaster recovery.

What to do:

- Implement the 3-2-1 backup rule: 3 copies, 2 different media, 1 offsite.
- For client sites: use automated backup (UpdraftPlus, BlogVault, or server-level snapshots).
- Test restoration quarterly -- a backup that cannot be restored is worthless.
- Document your backup procedure in one page.

2.4 Supply Chain Security

What NIS2 says: Entities must assess and manage security risks in their supply chain.

What to do:

- List every third-party service you use: hosting providers, DNS, email, plugins, payment processors.
- For each, document: what data they handle, their security certifications (SOC 2, ISO 27001), and their data processing location.
- Require your own subcontractors to meet equivalent security standards.

2.5 Security in System Acquisition and Maintenance

What NIS2 says: Security must be part of how you acquire, develop, and maintain systems.

What to do:

- Keep all systems updated: WordPress core, plugins, themes, server software.
- Remove unused plugins and themes -- they are attack surface.
- Use a maintenance schedule: weekly updates for security patches, monthly full review.

2.6 Vulnerability Handling and Disclosure

What NIS2 says: Entities must have processes for handling vulnerabilities.

What to do:

- Subscribe to security advisories for all major software you use.
- For WordPress: use WPScan or a security plugin to detect known vulnerabilities.
- Establish a disclosure channel: create a security@yourdomain.com email address.

2.7 Testing and Auditing

What NIS2 says: Entities must test and audit their security measures regularly.

What to do:

- Run an automated security scan quarterly (e.g., HackerTarget, Detectify, or WPScan).
- Review access logs monthly for suspicious activity.

- Document all tests and their results.

2.8 Use of Cryptography and Encryption

What NIS2 says: Entities must use appropriate cryptography and encryption.

What to do:

- Ensure HTTPS everywhere (LetsEncrypt is free -- use it on all client sites).
- Encrypt all backups (at rest and in transit).
- Use a password manager (Bitwarden, 1Password) -- never share passwords in email.
- Enable MFA on all administrative accounts.

2.9 Human Resources Security

What NIS2 says: Entities must ensure personnel understand and follow security practices.

What to do:

- New employee onboarding: 30-minute security briefing covering: password hygiene, phishing awareness, data handling, incident reporting.
- Annual refresher (send a 5-minute guide, no need for formal training).
- Include confidentiality clauses in employment contracts (or independent contractor agreements).

2.10 Access Control

What NIS2 says: Entities must control access to systems and data.

What to do:

- Implement least privilege: every user gets only the access they need to do their job.
- Use role-based access: separate admin access from day-to-day accounts.
- Review access quarterly -- remove accounts for former employees or contractors immediately.
- For WordPress: use a user role editor plugin to create granular permissions.

Chapter 3: Incident Reporting -- What, When, and How

NIS2 imposes strict incident reporting timelines. For Important entities: 24 hours for early warning, 72 hours for full notification.

What Qualifies as a Reportable Incident

Under NIS2, an incident is "any event having an actual adverse effect on the security of network and information systems." For a web agency, this includes:

- Successful intrusion into client hosting environment
- Ransomware attack on your systems
- Data breach involving client data (customer PII, credentials, financial data)
- Extended service outage caused by a security event
- Compromise of a client's WordPress admin account

The Reporting Timeline

Time	Action
------	--------

-----	-----
Within 24h	Send early warning to your security contact or client
Within 72h	Submit detailed notification (what happened, impact, ongoing risks)
Upon resolution	Final report (root cause, actions taken, preventive measures)

Incident Report Template

Early Warning (within 24h):

To: [Client Security Contact]
 Subject: Security Incident Notification – [Brief Description]

We are writing to inform you of a security incident affecting [systems/services].

Status: Ongoing / Contained / Resolved
 Initial impact: [Brief description]
 Current actions: [What we are doing]
 Next update: [Date/time]

Contact: [Your name], [Your email/phone]

Full Notification (within 72h):

To: [Client Security Contact]
 Subject: Security Incident Report – [Reference Number]

Incident type:
 Date and time detected:
 Date and time occurred:
 Systems affected:
 Data impact:
 Root cause (preliminary):
 Actions taken:
 Ongoing risks:
 Recommended client actions:

What Not to Do

- Do not assume you are "too small to report" -- under NIS2, under-reporting carries penalties.
- Do not wait for full clarity before sending the early warning. The 24-hour window is short.
- Do not report without preserving evidence (logs, screenshots, timestamps).

Chapter 4: Supply Chain Security Requirements

This is the chapter that matters most to web agencies. Under NIS2, your clients are required to assess you.

What Your Clients Will Ask For

Based on actual procurement questionnaires from EU enterprises in 2025-2026, expect these questions:

- Do you have a documented Information Security Policy?
- What backup procedures do you follow?
- Are you certified under ISO 27001, SOC 2, or equivalent?
- Where is client data stored and processed?
- What access controls do you have in place?
- Do you conduct regular security testing?

- How do you handle incidents?
- Do you conduct background checks on employees?
- Do you use subcontractors or third parties that handle client data?
- What is your password policy?

Building Your Vendor Security Questionnaire Response

Create a single PDF document (call it "Vendor Security Profile") that answers all 10 questions. Update it annually. Share it proactively in pitches -- it signals professionalism.

What to Require from Your Own Vendors

Your hosting provider, DNS provider, email provider, and any SaaS tools should meet equivalent standards. At minimum, they should have:

- SOC 2 Type II or ISO 27001 certification (for major providers)
- A published Security Page on their website
- Data processing agreements (DPA) available on request
- EU data residency options (for GDPR/NIS2 alignment)

Chapter 5: Contract Clauses Every Agency Needs

Add these clauses to your client contracts now. They protect both you and your client under NIS2.

Clause 1: Security Obligations

The Provider shall maintain appropriate technical and organisational security measures to protect the Client's data and systems, including but not limited to: access controls, encryption in transit and at rest, regular security updates, and backup procedures.

Clause 2: Incident Notification

The Provider shall notify the Client within 24 hours of becoming aware of any security incident affecting the Client's data or systems, and shall provide a full incident report within 72 hours, including root cause analysis and remediation steps.

Clause 3: Subcontractor Security

The Provider shall ensure that any subcontractor or third-party service provider with access to Client data or systems maintains security measures equivalent to those described in this agreement, and shall provide a list of subcontractors upon request.

Clause 4: Data Processing

The Provider shall process Client data only in accordance with the Client's documented instructions. Data shall be stored and processed within the European Economic Area unless otherwise agreed in writing.

Clause 5: Compliance Verification

Upon the Client's reasonable request (not more than once per 12-month period), the Provider shall complete a security questionnaire or provide documentary evidence of the security measures implemented under this agreement.

Chapter 6: 30-Day Compliance Checklist

Follow this day by day. By day 30, you will have a documented compliance framework that satisfies most client requirements.

Week 1: Foundation

Day	Task	Done?
-----	-----	-----
1	Write your Information Security Policy (1 page)	■
2	Designate a Security Contact person	■
3	Enable MFA on all admin accounts (email, hosting, DNS, WordPress)	■
4	Set up a password manager and audit all shared passwords	■
5	Implement HTTPS on all client sites (LetsEncrypt)	■
6	Create asset inventory: all servers, domains, SaaS tools	■
7	Review and document Week 1. Secure this document.	■

Week 2: Operations

Day	Task	Done?
-----	-----	-----
8	Set up automated backups (3-2-1 rule)	■
9	Test a backup restoration (restore one site from backup)	■
10	Install security monitoring (Wordfence or equivalent) on all client sites	■
11	Subscribe to security advisories for WordPress, plugins, server software	■
12	Create Incident Response Plan (6-step one-pager)	■
13	Clean up: remove unused WordPress plugins, themes, user accounts	■
14	Review and document Week 2	■

Week 3: Contracts & Vendors

Day	Task	Done?
-----	------	-------

-----	-----	-----
15	Add NIS2 clauses to your standard client contract (see Chapter 5)	■
16	Audit your vendors: request DPAs or security docs from hosting/DNS/email providers	■
17	Create your Vendor Security Profile document	■
18	Update employee/contractor agreements with confidentiality clauses	■
19	Set up role-based access controls (WordPress user roles, hosting accounts)	■
20	Review all third-party integrations and remove unused ones	■
21	Review and document Week 3	■

Week 4: Testing & Completion

Day	Task	Done?
-----	-----	-----
22	Run a full security scan of your infrastructure (WPScan, HackerTarget)	■
23	Review access logs for suspicious activity (last 30 days)	■
24	Conduct a 30-minute team security briefing (even if it's just you)	■
25	Create security@yourdomain.com and document disclosure process	■
26	Set up uptime monitoring for all client sites (UptimeRobot, free tier)	■
27	Document everything in a single "Security Compliance" folder	■
28	Final review: does anything from Week 1 need updating?	■
29	Send your Vendor Security Profile to your top 3 clients proactively	■
30	Set calendar reminders: quarterly review, annual policy update	■

Ongoing Schedule

Interval	Task
----------	------

-----	-----
Weekly	Apply security updates (WordPress core, plugins, themes)
Monthly	Review access logs, verify backups are running
Quarterly	Run security scan, test backup restoration, review access controls
Annually	Update Information Security Policy, review vendor agreements

Chapter 7: Key Resources

Free Security Tools for Small Agencies

Tool	Purpose	Cost
-----	-----	-----
LetsEncrypt + Certbot	Free SSL/TLS certificates	Free
Wordfence	WordPress security (firewall, malware scan)	Free tier available
UpdraftPlus	WordPress backups (auto to cloud)	Free tier available
Bitwarden	Password manager, shared team folders	Free tier (2 users)
UptimeRobot	Uptime monitoring, 5-minute intervals	Free (50 monitors)
HackerTarget	External vulnerability scanning	Free tier
WPScan	WordPress vulnerability database	Free for basic use
Better Stack	Incident alerting and status pages	Free tier
Have I Been Pwned	Check leaked credentials	Free

Standards to Reference

- ISO 27001:2022 -- International standard for information security management
- NIST Cybersecurity Framework -- US framework, widely referenced globally
- OWASP Top 10 -- Web application security risks
- WCAG 2.1 AA -- Accessibility standard (referenced alongside EAA)

Appendices

Appendix A: Sample Information Security Policy

[AGENCY NAME] – INFORMATION SECURITY POLICY
Version: 1.0 | Date: [DATE]

1. PURPOSE
This policy defines the security requirements for all employees, contractors,

and systems at [AGENCY NAME].

2. SCOPE

All employees, contractors, systems, and data.

3. POLICY

- a) Access Control: Access is granted on a least-privilege basis. All access is reviewed quarterly. Accounts are terminated within 24 hours of departure.
- b) Authentication: Multi-factor authentication is required on all administrative systems. Passwords must be minimum 16 characters and stored in Bitwarden.
- c) Data Protection: Client data is stored within the EU. All data in transit is encrypted via TLS 1.2+. Backups are encrypted at rest.
- d) Incident Response: All security incidents must be reported to [CONTACT] within 1 hour. See Incident Response Plan for procedures.
- e) Software Updates: All systems receive security patches within 7 days of release.
- f) Third Parties: All vendors with access to our systems must meet equivalent security standards.

4. REVIEW

This policy is reviewed annually by [ROLE/TITLE].

Appendix B: Incident Response Plan (One-Pager)

[AGENCY NAME] – INCIDENT RESPONSE PLAN

Version: 1.0 | Date: [DATE]

STEP 1: DETECT

- Automated alerts (Wordfence, UptimeRobot, Better Stack)
- User reports suspicious behavior
- Client reports issue

STEP 2: TRIAGE (within 15 min)

- Severity: Low / Medium / High / Critical
- Impact: Single client / Multiple clients / Internal only
- Assign: Response lead (rotate if needed)

STEP 3: CONTAIN

- Isolate affected systems (remove from network, disable account)
- Preserve evidence (logs, screenshots)
- Notify affected clients (within 24h for security incidents)

STEP 4: ERADICATE

- Remove root cause (malware, compromised account, vulnerability)
- Apply security patches
- Reset all affected credentials

STEP 5: RECOVER

- Restore from clean backup
- Verify system integrity
- Monitor for 48 hours post-recovery

STEP 6: REVIEW (within 7 days)

- Root cause analysis
- What worked / what didn't
- Update procedures to prevent recurrence

CONTACTS:

Security Lead: [NAME], [PHONE]

Backup: [NAME], [PHONE]

IT Emergency: [NAME], [PHONE]

Appendix C: Vendor Security Profile Template

Answer these 10 questions and compile into a PDF:

- Our Information Security Policy is documented and reviewed annually.
- We follow the 3-2-1 backup rule and test restoration quarterly.
- We are [ISO 27001 / SOC 2 / self-assessed] aligned.
- Client data is stored within the EU at [hosting provider].

- Access control is least-privilege with quarterly reviews.
- We run security scans quarterly and apply patches within 7 days.
- Incident response: 24h notification, 72h full report.
- All employees sign confidentiality agreements.
- Our subcontractors are [list] and are held to equivalent standards.
- Our password policy requires 16+ characters and MFA on all systems.

Free tools from the publisher

- NIS2 Scope Checker -- answer five questions and see whether NIS2 likely applies to your agency:
<https://hermes-passiv.pages.dev/nis2-check>
- Website compliance scan -- free automated check of privacy policy, cookie banner and security headers on any site: <https://hermes-passiv.pages.dev/scan>
- Ask the AI compliance assistant -- free answers to NIS2 questions, no signup:
<https://hermes-passiv.pages.dev/books/nis2-for-agencies#bookAi>

Final Words

NIS2 compliance for a small web agency is not about building a Fort Knox. It is about being organized, documented, and transparent. The agencies that will thrive under NIS2 are not the ones with the biggest security budgets -- they are the ones that can show their clients: "Here is what we do, here is how we do it, here is proof it works."

Start today. Follow the 30-day checklist. You do not need a compliance team. You just need one afternoon and this guide.

About this guide

Written by a compliance automation specialist. Published by Mahope / Hermes Passiv.

This guide is updated regularly as NIS2 enforcement evolves. The latest version is always available on Amazon Kindle.

Version 1.0 -- August 2026

Disclaimer: This guide provides practical guidance based on publicly available information about NIS2. It does not constitute legal advice. For specific legal questions, consult a qualified attorney in your jurisdiction.

GDPR Compliance for Small Web Agencies

From: Complete EU Compliance Bundle — Version 1.0

GDPR Compliance for Small Web Agencies

A Practical Guide to Client Data Protection Without a Legal Department

By Mahope

Every agency that builds or maintains websites for European clients is already processing personal data -- and already responsible for GDPR compliance, whether anyone has said so out loud or not. This guide turns that vague obligation into a concrete checklist you can complete in two weeks.

It is written for agencies of one to fifty people: no legal department, no DPO on payroll, no budget for outside counsel. Everything here can be done by a technically competent founder with a spreadsheet and an afternoon.

What this guide gives you:

- A plain-language map of which GDPR rules actually apply to an agency
- The three documents every client relationship needs (DPA, processor list, incident plan)
- A 14-day action plan that takes you from "nothing written down" to "defensible"
- Contract clauses you can paste into your next agreement
- Templates and checklists referenced throughout

This guide is practical, not exhaustive. Where the law leaves room for judgement, we tell you what most agencies do and what regulators expect. It is general information about compliance practice -- not legal advice for your specific situation.

Chapter 1 -- Why This Applies to You (Yes, You)

The myth that gets agencies in trouble

The most common misconception among small agencies is some version of: "We just build websites. Our clients own the data."

Under GDPR, that's only half true. When your client asks you to build a contact form, set up analytics, or configure email marketing, you are processing personal data on their behalf. That makes you a data processor under Article 28, with direct legal obligations of your own -- obligations that exist regardless of what your contract says.

Regulators have been explicit about this. Processing without a written contract that meets Article 28 requirements is itself a violation -- for both parties. If your client gets audited and cannot produce a compliant contract with you, that is their violation. If you process data without the right terms in place, it is yours.

What "processing" means in practice

Almost everything an agency does touches personal data:

Activity	Personal data involved
-----	-----
Building a contact form	Name, email, message content
Setting up Google Analytics	IP addresses, device identifiers

Configuring email/newsletter tools	Subscriber lists
Managing hosting	Access logs, error logs with IPs
Website maintenance	Database contents, user accounts
Handling staging/backup environments	Full copies of production data

If you do any of these for EU-based clients (or their EU visitors), you're in scope. There is no size threshold for GDPR -- unlike NIS2, a one-person shop is fully covered.

What actually happens when something goes wrong

The dramatic fines make headlines, but they are not the realistic risk for a small agency. The realistic risks, in order:

- A client procurement review fails you. Larger clients increasingly audit vendors before signing. No DPA template? You lose the deal or sign their one-sided paper.
- A data breach you can't document. You must notify the affected client (and often the authorities) within 72 hours of becoming aware. If you don't know where data lives or who to call, you miss the deadline.
- A subject access request lands. A website visitor demands their data. Your client calls you in a panic because the request involves systems you built.
- Insurance won't cover you. Professional indemnity insurers now routinely ask for documented GDPR processes.

None of these require a regulator to be involved. All of them cost real money and reputation. Compliance is cheaper than any of them.

The good news

For a small agency, GDPR is mostly documentation discipline, not technology. You almost certainly already do most of the right things technically -- encrypted connections, access control, backups. What you lack is the paperwork that proves it. This guide fixes exactly that.

Chapter 2 -- Your Two Roles: Controller vs Processor

GDPR assigns every organisation one of two roles in each processing activity, and the roles carry different duties. Getting this straight is the foundation for everything else.

When you're a controller

You are a controller when you decide why and how personal data is processed -- i.e., it's your own business purpose. For an agency this typically means:

- Your own CRM and client records
- Your own marketing and newsletter
- Your own website's analytics and contact form
- Job applications you receive
- Freelancer and supplier records

As a controller you owe individuals the full set of duties: lawful basis, privacy notice, data subject rights handling, breach notification to authorities, records of processing.

When you're a processor

You are a processor when you handle personal data on someone else's instructions -- your clients' data. Building and maintaining their sites, hosting their databases, administering their email accounts.

As a processor you owe narrower but still direct duties:

- Process only per documented instructions (the contract)
- Keep records of what you process for whom
- Secure the data appropriately
- Notify each client of a breach without undue delay (your contracts should say within 24-48 hours)
- Get authorisation before using sub-processors (e.g., your hosting provider)
- Delete or return data when the engagement ends

Why the distinction matters commercially

Two consequences worth internalising:

- Your clients owe you DPAs. Every client for whom you touch personal data should have signed a data processing agreement with you. Most small agencies have none -- meaning they've been non-compliant since day one, along with their clients.
- You owe your sub-processors the same. Your hosting provider is your sub-processor when working on client sites; you need their DPA on file (all major providers publish theirs) and your clients need to be told roughly who they are.

One role per activity

Note that you can be both at once for the same client: controller for your invoicing data about them, processor for their website users' data. Document them separately. Chapter 4's register handles this cleanly.

Chapter 3 -- The Three Documents That Matter

Small-agency compliance boils down to having three documents current and in use. Skip the forty-page compliance manual; these are what get requested, checked, and relied upon.

Document 1: The Data Processing Agreement (DPA)

The single most important piece of paper in your stack. It's required by Article 28 whenever a controller engages a processor -- which means between your clients and you, and between you and your hosts/tools.

A compliant DPA covers:

- Subject matter, duration, nature and purpose of processing
- Types of data and categories of data subjects
- Your obligation to process only on documented instructions
- Confidentiality commitments for staff
- Security measures (reference an annex)
- Rules on sub-processors (with a list + notice period for changes)
- Assistance with data subject rights requests
- Breach notification commitment (we recommend 24 hours)
- Deletion or return of data at termination
- Audit/cooperation rights

How to get there fast: Use the standard EU Commission SCCs annexed as a DPA module, or start from your national DPA guidance body's template. Don't draft from scratch. Send your template to new clients as part of the standard contract pack -- negotiating DPAs one by one is where small agencies lose weeks.

Appendix B includes clause language you can adapt.

Document 2: The Record of Processing Activities (RoPA)

Article 30 requires controllers to keep a written record of all processing activities. Processors must keep one too. For a small agency, this is a spreadsheet with one row per activity, not software.

Columns we recommend:

Column	Example entry
-----	-----
Activity	Hosting & maintenance of Client X website
Role	Processor
Controller	Client X ApS
Categories of data	Contact-form submissions: name, email, message
Data subjects	Website visitors
Where stored	[Host] EU region, backups in same region
Retention	Per client instructions; deleted 30 days after termination
Transfers	None outside EEA
Security measures	TLS, 2FA admin access, encrypted backups

One afternoon fills it in for a typical agency. Update it whenever you take on a new client or tool -- make it part of project kickoff.

Document 3: The Incident Response Plan

When a breach happens you have hours, not days. Write the plan now, while calm. Minimum content:

- Definition of a breach: any accidental or unlawful destruction, loss, alteration, unauthorised disclosure of, or access to personal data -- including ransomware, misdirected emails, exposed backups, and compromised admin accounts.
- Who does what: one person leads, one contacts affected clients, one preserves evidence. In a tiny agency, one name per task is fine.
- First-hour actions: contain (revoke credentials, isolate), assess scope (which data, whose, how many), preserve logs.
- Notification clock: clients within 24h of confirming a breach involving their data. Controllers then have 72h to notify supervisory authorities; help them meet it.
- Template messages: pre-draft the client notification email and the facts you'll need (what happened, when detected, data involved, mitigation, next update time).
- Post-incident review: what failed, what changes.

Test it once a year with a tabletop exercise -- an hour, one fake ransomware scenario. Write down what went wrong in your response and fix the plan.

Chapter 4 -- Your Privacy Infrastructure

Beyond the big three documents, four pieces of infrastructure keep you out of trouble day-to-day.

Privacy notices

Controllers need a privacy notice wherever personal data is collected. Two you likely owe today:

- Your own site's privacy policy -- covering your contact form, analytics, newsletter. Most agencies either lack one or copied one that describes activities they don't actually have. Rewrite it against reality.
- Client-facing statements -- your clients owe notices on their sites; offering a review of their privacy policy as part of projects is both good service and a differentiator (and pairs naturally with the accessibility statement work from our companion guide).

Lawful bases -- keep it boring

For your own operations: legitimate interest for B2B records and security logging (document the balancing test briefly), contract for serving clients, consent for your newsletter.

Do not overthink consent banners for your own brochureware site beyond what your analytics actually requires. Do not rely on consent for anything essential -- consent can be withdrawn and then you must stop.

Data minimisation and retention

The cheapest way to reduce breach impact is holding less data. Practical rules:

- Collect nothing on forms you don't act on ("company size" fields nobody reads)
- Purge old prospect and applicant data annually -- calendar it
- Set log retention explicitly (e.g., 30-90 days), don't let defaults accumulate forever
- Staging environments: use anonymised data, never full production copies left lying around

Sub-processors and transfers

Keep a simple sub-processor list: hosting, backup, email delivery, analytics, support tooling. For each, note the DPA location (most major providers self-host their SCCs-backed DPA online -- link it) and whether data stays in the EEA.

Transfers outside the EEA need a transfer mechanism -- in practice, Standard Contractual Clauses plus a Transfer Impact Assessment note. Nearly every mainstream SaaS provider has this handled; your job is to record it, not renegotiate it.

Chapter 5 -- Security Measures That Actually Count

Article 32 requires security "appropriate to the risk." Regulators don't expect enterprise SOC2 from a five-person agency; they do expect the fundamentals done consistently.

The baseline twelve

- TLS everywhere, HSTS on
- Unique admin accounts -- no shared logins, ever
- Two-factor authentication on all admin panels, hosters, code repos
- Password manager for the team
- Least privilege: developers don't have standing production database write access
- Encrypted backups, tested restore at least twice a year
- Automatic updates for CMS cores and critical plugins; a documented patching routine
- Offboarding checklist: revoke access same-day when someone leaves
- Encryption of laptops (FileVault/BitLocker) -- default-on, verify once
- Logging on production systems, retained long enough to investigate an incident

- Vendor review: before adopting a new tool that touches client data, confirm its DPA and security page exist
- Client handover: documented credential transfer, keys rotated after handover

That list costs almost nothing and satisfies the overwhelming majority of what an auditor or client questionnaire will ask. Everything above it is proportional response to specific risk.

The staging environment trap

The single most common small-agency failure: production data cloned to staging servers with weaker access controls, forgotten for years. Fix with policy: staging uses synthetic or anonymised data; if production data is truly necessary, the staging environment inherits production-grade access controls and is destroyed after use.

Chapter 6 -- Handling Data Subject Rights

Individuals can exercise eight rights; in agency life you will realistically encounter three.

Access requests (DSARs)

Someone asks what data you hold about them. As controller for your own data, you must respond within one month. As processor, you assist your client within contractual timescales.

Practical procedure:

- Log the request (date, identity verification method, scope)
- Search the places you know: CRM, email, website databases, backups (note: you generally inform the requester about backups rather than restoring them if restoration would be disproportionate)
- Compile and respond in plain language
- Deliver securely

Erasure requests

Delete where no overriding obligation requires retention. Watch for: legal retention duties (invoices -- typically 5 years in Denmark), ongoing contracts, and legitimate-interest overrides. Document your decision either way.

Portability and rectification

Portability applies to data given by consent or contract, delivered machine-readably -- usually a CSV export from the relevant system. Rectification is simply correcting the record everywhere it appears.

Agency angle: your clients will fumble these requests. Offering a defined service -- "we handle DSAR fulfilment for sites we maintain, 48h SLA" -- is a genuine productised upsell that deepens retention.

Chapter 7 -- The 14-Day Action Plan

Everything in this guide compressed into two working weeks. Assume 1-2 hours per day.

Week 1 -- Map and document

- Day 1: List every system you use that holds personal data (yours and clients'). Spreadsheet.
- Day 2: Classify each row: are you controller or processor?
- Day 3: Draft your RoPA using the Chapter 4 columns.
- Day 4: Find and file DPAs from your hosting, backup, and email providers. Note gaps.
- Day 5: Adopt or adapt a DPA template (Appendix B) for your client contracts.

- Day 6: Write your incident response plan skeleton (Chapter 3).
- Day 7: Buffer / catch-up.

Week 2 -- Close gaps and prove it

- Day 8: Run the baseline-twelve security checklist; note failures.
- Day 9: Fix authentication gaps first (unique accounts + 2FA).
- Day 10: Fix staging-environment issues; purge stale data copies.
- Day 11: Write/rewrite your own privacy policy.
- Day 12: Set retention rules and calendar annual purges.
- Day 13: Draft client-notification templates for breaches.
- Day 14: Review everything with fresh eyes; schedule quarterly review.

After two weeks you will have more documented compliance than the large majority of similarly sized agencies. Maintain with one hour per month.

Chapter 8 -- Selling Compliance as a Service

Once your own house is in order, the knowledge becomes revenue.

Products agencies successfully sell

- Compliance starter package: RoPA setup + DPA signing + privacy policy review for new retainers
- Annual compliance review: fixed-price audit against the Chapter 7 checklist
- DSAR handling retainer: rights-request fulfilment for maintained sites
- Breach readiness: incident plan drafting + tabletop exercise for clients

Each is scoped, repeatable, and uses documents you already built for yourself. Clients buying NIS2 or accessibility work (see our companion guides) will frequently add GDPR scope -- the sales conversation is warm, not cold.

Positioning

Lead with risk removal, not law: "Your vendor reviews ask for DPAs and records of processing. We set those up and keep them current." Procurement friction is the pain buyers feel; cite it directly.

Appendix A -- Quick Reference Checklist

Documents

- ☐ DPA template adopted and included in contract pack
- ☐ Signed DPA with every active client
- ☐ Provider DPAs filed (hosting, backup, email, analytics)
- ☐ RoPA current, updated at each new engagement
- ☐ Incident response plan written, tested annually
- ☐ Own-site privacy policy accurate

Security

- ☐ Unique accounts + 2FA everywhere
- ☐ Password manager in use
- ☐ Backups encrypted, restores tested twice yearly
- ☐ Patching routine documented and running

- [] Staging free of raw production data
- [] Offboarding checklist executed for every departure

Ongoing

- [] Quarterly: RoPA + sub-processor list review
- [] Annually: purge stale data, tabletop exercise, policy refresh
- [] Per project: DPA attached, data flows added to RoPA

Appendix B -- Core DPA Clause Language

Adapt with counsel as needed; these reflect common market standards.

Instructions. Processor shall process personal data solely on documented instructions from Controller, including with regard to transfers, unless required otherwise by Union or Member State law.

Sub-processors. Processor may engage sub-processors listed in Annex III. Processor shall give Controller thirty (30) days' written notice of any intended change and an opportunity to object on reasonable grounds.

Assistance. Processor shall promptly notify Controller if it receives a request from a data subject to exercise any right, and shall assist Controller in responding, taking into account the nature of the processing.

Breach notification. Processor shall notify Controller without undue delay, and in any case within twenty-four (24) hours, after becoming aware of a personal data breach affecting Controller's personal data.

Return and deletion. Upon termination, Processor shall, at Controller's choice, delete or return all personal data and delete existing copies, except where Union or Member State law requires storage.

Audit. Processor shall make available to Controller all information necessary to demonstrate compliance with Article 28 obligations and allow for and contribute to audits, including inspections.

Appendix C -- Glossary

Controller -- decides purposes and means of processing. Processor -- processes on behalf of a controller. DPA (Data Processing Agreement) -- mandatory Article 28 contract between controller and processor. RoPA -- Record of Processing Activities (Art. 30). DSAR -- Data Subject Access Request. Personal data breach -- any security failure exposing personal data, including loss of availability. SCCs -- Standard Contractual Clauses for transfers outside the EEA. Supervisory authority -- national regulator (in Denmark: Datatilsynet). Pseudonymisation -- replacing identifying fields so data can't be attributed without additional information held separately.

Free tools from the publisher

- Website compliance scan -- free automated check of privacy policy, cookies and security headers:
<https://hermes-passiv.pages.dev/scan>
- Cookie banner checker -- see what cookies a site sets before consent:
<https://hermes-passiv.pages.dev/cookie-check>
- Ask the AI compliance assistant -- free answers to GDPR questions, no signup:
<https://hermes-passiv.pages.dev/books/gdpr-for-agencies#bookAi>

EAA Compliance Checklist for WordPress Sites

From: Complete EU Compliance Bundle — Version 1.0

EAA Compliance Checklist for WordPress Sites

A Practical Guide to Meeting EU Accessibility Requirements Under the European Accessibility Act

Version 1.0 -- August 2026

Foreword

If you manage WordPress sites for EU clients, the European Accessibility Act (EAA) affects you directly.

The EAA requires that digital services -- including websites, mobile apps, and e-commerce platforms -- meet specific accessibility standards. The deadline (June 28, 2025) has now passed. Enforcement is ramping up across EU member states.

This is not about "nice to have" accessibility. This is about legal compliance.

This guide gives you a practical, step-by-step checklist for making WordPress sites EAA-compliant. No theory, no abstract standards -- just actionable tasks you can work through site by site.

Chapter 1: Is Your Site in Scope?

The EAA applies to digital services offered in the EU. If your agency builds, hosts, or maintains any of the following, you are in scope:

Service type	Examples	In scope?
-----	-----	-----
E-commerce	WooCommerce stores, membership sites, booking systems	■ Yes
Public sector	Municipality sites, government services	■ Yes (also covered by EN 301 549)
Banking/Insurance	Financial service sites	■ Yes
Transport	Booking portals, schedule sites	■ Yes
Telecom	ISP portals, mobile operator sites	■ Yes
Private blogs	Personal sites, portfolio sites	■ No (unless commercial)
Brochure sites	Small business info sites	■■ Likely yes if selling

The Key Question

Ask this for every site you manage:

Does this site offer a service or product to the EU public?

If yes, it needs to be EAA-compliant. Content management systems (WordPress), e-commerce platforms (WooCommerce), and booking systems are explicitly included.

Chapter 2: The 10-Point EAA Compliance Checklist

This is the core of the guide. Each item is a concrete check that you can run on any WordPress site. Pass all 10, and the site is demonstrably EAA-compliant.

1. Color Contrast (WCAG 2.1 AA)

Requirement: Text must have a contrast ratio of at least 4.5:1 (normal text) or 3:1 (large text 18pt+ / 14pt bold+).

How to check:

- Use the WebAIM Contrast Checker (webaim.org/resources/contrastchecker/)
- Enter the text color hex and background color hex
- Pass/fail result instantly
- Testing tool: Accessibility Insights browser extension (free, Microsoft)
- Testing tool: WAVE browser extension (free, WebAIM)
- Quick check: Toggle "High Contrast" mode in browser dev tools

Common WordPress violations:

- Light gray text on white backgrounds (most common)
- Blue link text on blue-ish backgrounds
- Footer text on dark backgrounds
- Button text against button colors

Fix: Adjust colors in the theme customizer, or add custom CSS. If the theme is locked, create a child theme.

2. Alt Text on All Images

Requirement: Every meaningful image must have alt text. Decorative images must have alt="" (empty alt).

How to check:

- Run the WAVE extension -- it lists all missing alt attributes
- Browser DevTools > Accessibility tab > inspect images
- For WooCommerce: check product images, category images

Common violations:

- Gallery images uploaded without alt text
- Logo in header/ footer without alt
- Social media icons without alt
- Decorative spacer images without alt=""

Fix in WordPress:

- Edit Media > Alternative Text field
- For existing sites with hundreds of images: use a plugin like Accessibility Checker to bulk-fill missing alt text
- Add alt="" for decorative images via filter functions

3. Heading Structure (h1-h6 Hierarchy)

Requirement: Headings must form a logical hierarchy. No skipped levels. No heading used purely for styling.

How to check:

- Install the HeadingsMap browser extension
- It shows the full heading outline -- look for:
- No h1 → fail
- h1 → h3 (skipping h2) → fail
- Multiple h1s → fail
- Text styled as heading but not marked as heading → fail

Common violations:

- Theme logo is a div, not an h1
- Widget headings jump from h2 to h4
- Page builders create headings at arbitrary levels
- Content editors use bold text instead of heading tags

Fix: Map the site's heading outline. Rename heading levels in the theme files or use a plugin that enforces accessibility-ready heading blocks.

4. Keyboard Navigation

Requirement: Every interactive element must be reachable and operable via keyboard alone. No "keyboard trap" where focus gets stuck.

How to check:

- Navigate the entire site using only Tab, Shift+Tab, Enter, and Escape
- Can you reach every link, button, and form field?
- Is there a visible focus indicator on every element? (blue outline or similar)
- Can you open all menus, dropdowns, and modals?
- Can you close them with Escape?

Common violations:

- Mega menus that don't work with keyboard
- Modal popups that trap focus
- Custom select/dropdown elements that aren't keyboard accessible
- "Skip to content" link missing or broken

Fix:

- Add a "Skip to content" link as the first focusable element
- Ensure all custom JavaScript components handle keyboard events
- Use native HTML elements (button, a) instead of divs with click handlers

5. Screen Reader Compatibility

Requirement: Content must be readable by screen readers. This means proper ARIA labels, semantic HTML, and logical reading order.

How to check:

- Turn on VoiceOver (macOS: Cmd+F5) or NVDA (Windows, free)
- Navigate the page -- does it make sense when read linearly?
- Do images have alt text that describes the content?

- Do form fields have proper labels?
- Are dynamic content updates announced?

Common violations:

- "Read more" links without context (should say "Read more about [page title]")
- Form fields without elements
- Buttons with icon-only content and no aria-label
- Auto-playing video/audio without pause controls

Fix:

- Add descriptive link text or use aria-label
- Ensure every has a matching
- Add aria-live regions for dynamic content

6. Forms and Input Validation

Requirement: Forms must be accessible. Error messages must be clear and programmatically associated with the field.

How to check:

- Submit a form with invalid data -- are errors communicated clearly?
- Are error messages linked to the input field via aria-describedby?
- Does the focus move to the first error?
- Are required fields marked with aria-required or text?

Common violations:

- Inline validation messages that disappear too fast
- Color-only error indicators (red border without text)
- CAPTCHAs that are not accessible (reCAPTCHA v2 has audio fallback, but test it)
- Confirmation messages that are not detected by screen readers

Fix:

- Use inline error messages with clear text
- Add aria-describedby connecting error to field
- Use accessible CAPTCHA alternatives (honeypot + time-based check)
- Use aria-live="polite" for confirmation messages

7. Links and Navigation

Requirement: Links must be distinguishable and meaningful. No "click here", no "read more" without context.

How to check:

- Scan all links on the page -- does each link text describe its destination?
- Are links visually distinguishable from surrounding text (underlined or high contrast color)?
- Do navigation menus have consistent structure across pages?

Common violations:

- "Click here" / "Learn more" / "Read more" as link text
- Color-only distinction for links (no underline)

- Same link text pointing to different URLs
- Broken anchor links

Fix:

- Rewrite link text to describe the target page
- Add text-decoration: underline to links (user preference style)
- Run a link checker (W3C Link Checker or browser extension)

8. Multimedia: Video and Audio

Requirement: All pre-recorded video must have captions. Audio must have transcripts. Live video must have captions where feasible.

How to check:

- Does every embedded video (YouTube, Vimeo) have captions turned on?
- Do audio players have visible play/pause/volume controls?
- Are auto-playing videos controllable?
- Is there a transcript for podcast/audio content?

Common violations:

- YouTube videos without auto-captions (auto-generated are insufficient for compliance)
- Background video with no pause button
- Audio content with no text alternative
- Animated GIFs with no description

Fix:

- Upload corrected captions to YouTube/Vimeo
- Add a "Pause" button for all moving content
- Provide text transcripts below audio embeds
- Add aria-label to describe GIF content

9. Resize and Zoom

Requirement: Content must be readable when zoomed to 200%. No horizontal scrolling. No text clipping.

How to check:

- Open the site
- Zoom to 200% (Cmd++ on macOS, Ctrl++ on Windows)
- Does the layout reflow or require horizontal scrolling?
- Is all text readable?
- Do buttons and links still have enough padding to be clickable?

Common violations:

- Fixed-width containers that don't scale
- Text in images that pixelate at 200%
- Off-canvas menus that overflow at zoom
- Tables with many columns that break layout

Fix:

- Use relative units (rem, %) instead of fixed px widths

- Test the theme's responsive breakpoints
- For complex tables: use responsive table techniques (horizontal scroll on mobile, or stacked rows)

10. Accessibility Statement

Requirement: EAA requires a published accessibility statement on the site. This includes current compliance status, contact information, and a process for reporting issues.

How to check:

- Is there an accessibility statement page?
- Does it include:
 - Compliance status (e.g., "Partially compliant with WCAG 2.1 AA")
 - Contact method for accessibility issues
 - Description of known limitations
 - Expected response time (typically 5 business days)

Common violations:

- No accessibility statement at all
- Statement is generic boilerplate with no actual commitment
- No contact form or email for reporting issues
- Statement claims "fully compliant" without evidence

Fix:

- Create a /accessibility-statement/ page
- Use the EAA-compliant statement generator (w3.org/WAI/accessibility-statement/)
- Include a contact form or dedicated email
- Update when compliance status changes

Chapter 3: Testing Tools (All Free)

Tool	What it checks	Cost
-----	-----	-----
WAVE Browser Extension	All 10 points, visual overlay	Free
Axe Browser Extension	Automated WCAG 2.1 AA checks	Free
Lighthouse (Chrome DevTools)	Integrated audit with scores	Free (built into Chrome)
WebAIM Contrast Checker	Color contrast ratios	Free
NVDA Screen Reader	Full screen reader test	Free (Windows)
VoiceOver	Full screen reader test	Free (macOS)
Accessibility Insights	Guided manual testing	Free (Microsoft)
HeadingsMap	Heading structure outline	Free

Recommended workflow:

- Run Lighthouse accessibility audit → fix automated findings

- Run WAVE → fix reported errors and contrast issues
- Run HeadingsMap → fix heading hierarchy
- Manual keyboard navigation test
- Test with NVDA or VoiceOver on 3 key pages

Chapter 4: 14-Day Fix Plan

Day	Task	Time
-----	-----	-----
1	Run Lighthouse + WAVE audit. Document all issues	1h
2	Fix all color contrast issues (check 1)	1h
3	Add alt text to all images (check 2)	1-3h
4	Fix heading hierarchy (check 3)	1h
5	Test and fix keyboard navigation (check 4)	2h
6	Test screen reader flow (check 5)	1h
7	Fix form accessibility (check 6)	1h
8	Fix links and navigation (check 7)	1h
9	Fix multimedia issues (check 8)	1h
10	Test resize and zoom (check 9)	30min
11	Create accessibility statement (check 10)	1h
12	Final full audit	1h
13	Client handover: document what was done	1h
14	Set up monitoring: quarterly re-scans	30min

Chapter 5: Making EAA Compliance a Revenue Stream

For agencies, EAA compliance is not just a legal obligation -- it is a service you can sell.

Service Pricing Guide

Service	Description	Price range
-----	-----	-----
EAA Audit	Full automated + manual audit with report	€500-1,500
EAA Remediation	Fix all 10 checklist items	€1,000-5,000

EAA Maintenance	Quarterly re-scans + fixes	€200-500/mo
Accessibility Statement setup	Create + publish	€150-300

Selling Compliance as a Service

Use the checklist in this guide as your service scope. When a client asks "what does EAA compliance cost?", hand them the 10-point checklist and say: "Each of these needs to be checked and fixed. We can do it all."

The 14-day fix plan becomes your project timeline. The accessibility statement becomes your deliverable.

Appendix A: Quick Reference Card

Print this and keep it at your desk.

EAA QUICK CHECK (WordPress)

- Color contrast $\geq 4.5:1$ → WebAIM tool
- All images have alt text → WAVE tool
- Heading hierarchy logical → HeadingsMap
- Full keyboard navigation → Tab test
- Screen reader compatible → VoiceOver/NVDA
- Forms have error messages → Submit test
- Links are descriptive → Visual scan
- Videos have captions → Player check
- 200% zoom no horizontal scroll → Zoom test
- Accessibility statement published → Site check

Appendix B: Sample Accessibility Statement (Template)

ACCESSIBILITY STATEMENT

Last updated: [DATE]

[AGENCY/SITE NAME] is committed to ensuring digital accessibility for people with disabilities. We are continually improving the user experience for everyone and applying the relevant accessibility standards.

Compliance Status

We strive to conform to WCAG 2.1 Level AA, which is the standard required under the European Accessibility Act.

Current Status: [Full compliance / Partial compliance]

Known limitations: [List any known issues]

Feedback

We welcome your feedback on the accessibility of this site.

Please contact us:

Email: accessibility@[domain]

Phone: [optional]

Response time: Within 5 business days

Technical specifications

- Accessibility relies on the following technologies: HTML, WAI-ARIA, CSS, JavaScript
- These technologies are relied upon for conformance with accessibility standards

Assessment approach

We assess accessibility using automated tools (Lighthouse, WAVE) and manual testing (keyboard navigation, screen reader testing).

This statement was last reviewed on [DATE] and will be updated annually.

Free tools from the publisher

- EAA / accessibility scanner -- run a free automated accessibility check on any WordPress site:
<https://hermes-passiv.pages.dev/scan>
- Contrast and alt-text are covered in the checklist -- test your live site here:
<https://hermes-passiv.pages.dev/page-profile>
- Ask the AI compliance assistant -- free answers to accessibility (EAA/WCAG) questions, no signup:
<https://hermes-passiv.pages.dev/books/ea-checklist#bookAi>

Final Words

EAA compliance for WordPress sites is achievable without a massive budget. The 10-point checklist covers the vast majority of issues found on the average WordPress site. Most fixes take an hour or less.

The agencies that treat accessibility as a feature -- and learn to sell it -- will win more clients, charge higher rates, and avoid the fines that are now being enforced across Europe.

Start with one site. Run the checklist. Fix what's broken. Document what you did. That is compliance.

Version 1.0 -- August 2026

Published by Mahope / Hermes Passiv

This guide provides practical guidance based on publicly available information about the European Accessibility Act. It does not constitute legal advice. For specific legal questions, consult a qualified attorney in your jurisdiction.

EAA Compliance for Shopify Stores

From: Complete EU Compliance Bundle — Version 1.0

EAA Compliance for Shopify Stores

A Practical Guide to Meeting EU Accessibility Requirements Under the European Accessibility Act

If your Shopify store sells to customers in the European Union, the European Accessibility Act (EAA) applies to you -- regardless of where your business is registered. This guide walks through exactly what you need to check, fix, and document.

Chapter 1: What the EAA Means for Your Shopify Store

The European Accessibility Act (Directive 2019/882) took full effect on June 28, 2025. It requires that products and services sold in the EU meet common accessibility standards. For e-commerce stores -- including those built on Shopify -- this means your website must be usable by people with disabilities.

Who must comply

Any business selling to EU consumers, regardless of where the business is headquartered. A Shopify store based in New York, Sydney, or Singapore that ships to customers in Germany, France, or Spain must comply.

Covered products and services under the EAA

For an e-commerce store, the EAA covers:

- The entire browsing experience (product catalog, search, navigation)
- Product selection and purchasing (filters, size/color selectors, add-to-cart)
- Checkout and payment flows
- Customer accounts and order tracking
- Contact forms and support interfaces

What happens if you don't comply

Each EU member state sets its own penalties. Current enforcement examples:

- Germany (BFSG): fines up to €100,000 per violation
- France: fines up to €75,000, with daily penalties for ongoing non-compliance
- Netherlands: administrative fines, plus orders to remediate within deadlines
- Spain: fines up to €600,000 for serious infractions
- Sweden: fines up to €900,000

Beyond fines: enterprise partners increasingly ask for proof of EAA compliance before signing contracts. A non-compliant store can lose B2B revenue even without a direct fine.

The standard you need to meet

The EAA references EN 301 549, which in turn references WCAG 2.1 Level AA as the baseline technical standard. From 2026, WCAG 2.2 Level AA is increasingly the expected level. Practically, WCAG 2.1 AA covers everything you need for compliance -- WCAG 2.2 adds a few new criteria that most Shopify themes already meet.

Chapter 2: EAA Compliance Requirements for E-commerce

The EAA's accessibility requirements map to four principles, known as POUR:

Perceivable

Information and user interface components must be presentable to users in ways they can perceive.

For a Shopify store this means:

- Text must have sufficient contrast against backgrounds (WCAG 1.4.3: 4.5:1 ratio for normal text, 3:1 for large text)
- Images must have text alternatives (alt text) -- especially product images
- Content must adapt to different screen sizes and zoom levels
- Colour must not be the only way to convey information (e.g., price changes indicated by colour alone)

Operable

User interface components and navigation must be operable.

For a Shopify store this means:

- All functionality must be available from a keyboard (no mouse-only interactions)
- Users must have enough time to read and use content
- Content must not cause seizures (no flashing animations)
- Navigation must be consistent and predictable
- Focus indicators must be visible when tabbing through the page

Understandable

Information and the operation of the user interface must be understandable.

For a Shopify store this means:

- Text must be readable (plain language where possible)
- Pages must appear and operate in predictable ways
- Input assistance must help users avoid and correct mistakes
- Form labels must be clear and associated with their fields

Robust

Content must be robust enough to be interpreted by a wide variety of user agents, including assistive technologies.

For a Shopify store this means:

- HTML must be valid and semantically correct
- ARIA landmarks and roles must be used correctly where needed
- Custom interactive elements (sliders, accordions, modals) must work with screen readers

Chapter 3: 10 Most Common Accessibility Failures on Shopify

Through automated scanning of hundreds of Shopify stores, these are the most frequent failures:

1. Missing alt text on product images

Product images are the heart of a Shopify store. Alt text is often left empty or filled with generic text like "product image" or the filename.

The fix: In the Shopify admin, edit each product and fill the "Alt text" field (a.k.a. image alt tag in the media manager). Describe the product clearly: "Red cashmere scarf draped over a wooden chair, soft lighting" rather than "scarf.jpg".

2. Low contrast text

Shopify themes frequently use light grey text on white backgrounds -- especially for prices, sale badges, and secondary information. This fails WCAG 1.4.3.

The fix: In Theme Editor (customize), go to Theme Settings → Typography or Colour. Increase contrast of body text. If the theme doesn't allow per-element control, add CSS via Custom CSS: `body { color: #1a1a1a; }` or override specific text colours.

3. Empty buttons and unlabelled links

"Add to cart" buttons that use only an icon without accessible text. Social media icons without aria-labels. "Read more" links that don't describe where they lead.

The fix: Use aria-label on icon-only buttons. For social links: Replace generic links with descriptive text.

4. Missing form labels

Newsletter signup forms, search bars, and contact forms often have placeholder text but no proper element.

The fix: Wrap form fields in tags or use aria-label. In Shopify, edit your newsletter section and ensure labels are visible. Placeholder text alone is not sufficient for legal compliance.

5. Keyboard traps

Product modals (quick-view, size guide, cart drawer) that trap keyboard focus -- users can tab into them but not out. This is a WCAG 2.1.2 failure and a common issue in many Shopify themes.

The fix: Ensure the Escape key closes the modal and returns focus to the trigger element. Test by tabbing through your site without using a mouse.

6. Missing page title and language

Shopify auto-generates page titles, but custom pages often have empty or duplicate titles. Missing attribute is also common.

The fix: In Theme Settings → Search engine listing, set a unique title for each page. Verify the tag includes `lang="en"` (or your store's language).

7. Non-descriptive link text

"Click here", "Shop now", "Learn more" -- these appear everywhere on Shopify stores. Screen reader users tab through links and hear each one in isolation.

The fix: Make link text unique and descriptive. "Shop men's winter jackets" instead of "Shop now". "Read our shipping policy" instead of "Click here".

8. Inaccessible colour swatches

Size, color, and material selectors that rely solely on colour to show the selected state. A colour-blind user or screen reader user cannot tell which option is selected.

The fix: Show both colour and text label on each swatch. Add a visible border or checkmark to the selected swatch. Ensure selected state is announced by screen readers.

9. Insufficient focus indicators

When tabbing through a Shopify store, the default blue outline is often removed by theme CSS (`outline: none`) without providing a visible replacement.

The fix: In Custom CSS, add: `*:focus-visible { outline: 3px solid #0066cc; outline-offset: 2px; }`. Never remove focus outlines without providing a visible alternative.

10. Missing accessibility statement

The EAA requires an accessibility statement on your store. Most Shopify stores don't have one.

The fix: Create a dedicated page at `/pages/accessibility-statement` listing your conformance level, any known issues, how to give feedback, and the supervisory authority for complaints. We provide a template in Chapter 7.

Chapter 4: Fixing Product Images -- The Right Way

Product images are your biggest accessibility surface area. Most Shopify stores have dozens to thousands of images, each needing proper alt text.

Shopify's alt text system

In Shopify admin, each product image can have alt text. Here's where to find it:

- Go to Products → All products
- Click a product
- Scroll to the Media section
- Click an image
- Find the "Alt text" field
- Enter a descriptive text

What good alt text looks like

Bad: "Blue sweater"

Good: "Navy blue merino wool sweater with crew neck, photographed on a grey mannequin against white background -- front view"

Bad: "Product image"

Good: "Handcrafted ceramic coffee mug in matte sage green, 350ml capacity -- shown with saucer from a 45-degree angle"

Bad: "Red shoes 2"

Good: "Women's size 37 red leather ballet flats with gold buckle detail, top-down view on marble floor"

Batch editing alt text

For stores with hundreds of products, consider:

- Exporting products via Shopify's CSV export/import
- Filling alt text in a spreadsheet
- Importing back into Shopify
- Or hiring a virtual assistant for one-off batch work
- AI alt-text generators (several Shopify apps exist -- review output manually, they can miss context)

When alt text isn't needed

Decorative images (background textures, purely decorative icons, spacers) should have empty alt text: `alt=""`. This tells screen readers to skip them. Do not omit the alt attribute entirely -- that causes some screen readers to read the image filename.

Chapter 5: Theme Accessibility

Your Shopify theme controls most of the accessibility surface. Some themes are better than others.

Choosing an accessible theme

Shopify's "Dawn" theme (the default) is a good starting point for accessibility. Other themes from the Shopify Theme Store vary widely. Look for themes that:

- Use semantic HTML (proper heading hierarchy: h1, h2, h3 in order)
- Include visible focus indicators
- Have configurable contrast settings
- Support proper keyboard navigation
- Include ARIA landmarks

Contrast: the most common theme failure

Many premium Shopify themes use designer-approved colour palettes that fail contrast checks. The most common pattern: light grey (#999 or #aaa) body text on white background.

Test every text colour in your theme using a contrast checker. WCAG 2.1 AA requires:

- Normal text (< 18px or < 14px bold): 4.5:1 contrast ratio
- Large text (≥ 18px or ≥ 14px bold): 3:1 contrast ratio
- UI components and icons: 3:1 contrast ratio

Custom CSS for accessibility

Add these rules to your theme's Custom CSS (in Theme Editor → Custom CSS):

```
/* Ensure all text meets contrast */
body, p, li, .text-body {
  color: #1a1a1a;
}

/* Visible focus for keyboard users */
*:focus-visible {
  outline: 3px solid #0066cc !important;
  outline-offset: 2px !important;
}

/* Remove focus outline only when using mouse */
*:focus:not(:focus-visible) {
  outline: none;
}

/* Ensure links are distinguishable */
a { text-decoration: underline; }
a:hover { text-decoration-thickness: 2px; }
```

Testing keyboard navigation

Go through your entire Shopify store using only the Tab, Shift+Tab, Enter, and Escape keys:

- Can you reach every link and button?
- Can you see which element is focused at all times?
- Can you close all modals and drawers?
- Can you complete a purchase without the mouse?

If any step fails, you have a keyboard accessibility issue.

Chapter 6: Checkout and Forms Accessibility

The checkout is your revenue conversion point. An inaccessible checkout loses sales.

Common Shopify checkout issues

- Auto-focus issues: Some themes move focus unexpectedly during checkout, disorienting screen reader users. Test the complete checkout flow.
- Error messages: Inline validation errors that aren't announced. Ensure error messages use `aria-describedby` on the invalid field. Shopify Checkout (Shopify's hosted checkout) is generally good -- custom checkouts are where problems arise.
- CAPTCHA: If you use CAPTCHA on any form, provide an audio alternative. Google reCAPTCHA v3 (invisible) avoids this issue entirely.
- Address auto-complete: Fields that change content dynamically need to announce changes to screen readers.

Shopify's hosted checkout vs. custom checkout

- Shopify Checkout (default): Shopify manages the accessibility of the checkout flow. It's reasonably good and continuously improving. Stick with it if possible.
- Custom checkout (Shopify Plus or third-party): You are fully responsible. Every form field, dropdown, and validation message must meet WCAG.

Form accessibility checklist

Every form on your store should:

- Have visible elements for each field
- Group related fields with and
- Show clear error messages linked to the field via `aria-describedby`
- Not auto-submit on field change without warning
- Support autocomplete attributes (`autocomplete="given-name"`, `autocomplete="email"`, etc.)

Chapter 7: The EAA Accessibility Statement

The EAA requires an accessibility statement. This is the most commonly missed requirement.

What must be in the statement

Field	Required	Details
-----	-----	-----
Name and contact of the merchant	Yes	Business name, address, email
Products/services covered	Yes	"This statement applies to our website at [URL]"
Conformance level	Yes	"WCAG 2.1 Level AA"
Known non-conformances	Yes	List any known accessibility issues, with reasons and remediation timeline
Date of preparation	Yes	When this statement was created
Last review date	Yes	When it was last reviewed/updated
Feedback mechanism	Yes	Email or form for accessibility feedback

Supervisory authority contact	Yes	Contact details for the EU member state's relevant authority
-------------------------------	-----	--

Sample accessibility statement

Accessibility Statement for [Store Name]

>

Last updated: [date]

>

[Store Name] is committed to ensuring digital accessibility for all users, regardless of technology or ability. We are actively working to make our website at [URL] accessible in accordance with the European Accessibility Act (Directive 2019/882) and EN 301 549, which references WCAG 2.1 Level AA.

>

Conformance status

>

As of [date], [URL] is partially compliant with WCAG 2.1 Level AA. We have identified and are working to remediate the following known non-conformances:

>

- [Known issue 1, e.g., "Product images published before [date] may lack descriptive alt text"]
- [Known issue 2, e.g., "Legacy product pages from [version] have heading hierarchy issues"]

>

Feedback

>

If you encounter an accessibility barrier on our website, please contact us at [accessibility@store.com]. We aim to respond within 5 business days and resolve issues within 30 days where feasible.

>

Supervisory authority

>

If you are not satisfied with our response, you may contact:

[Name of national authority]

[Address]

[Email/Website]

>

Preparation date: [date]

Last reviewed: [date]

Next review: [date + 3 months]

Where to publish the statement

Create a page on your Shopify store at /pages/accessibility-statement. Link to it from your footer or legal section alongside your Privacy Policy and Terms of Service.

How often to update

Review the statement every three months. Update it any time there is a material change in your compliance posture -- for example after a theme change, major feature release, or remediation work.

Chapter 8: Testing Your Shopify Store

You don't need expensive tools to check EAA compliance. Here's a tiered approach:

Free -- every store should do this

- Our EAA/WCAG Scanner (hermes-passiv.pages.dev/scan): Paste your store URL, get an instant letter grade with every issue listed by severity. 16 automated checks covering alt text, contrast, form labels, headings, language, and more.
- WAVE Browser Extension: Install the WAVE toolbar (free Chrome/Firefox extension). Scan any page for contrast errors, missing alt text, and structural issues.
- Manual keyboard test: Tab through your entire site. Note any focus loss, keyboard traps, or invisible elements.
- Contrast checker: Use WebAIM's free contrast checker to test all text colours in your theme against their background.
- Screen reader test: Use VoiceOver (macOS, built-in) or NVDA (Windows, free) to navigate your store. Listen for issues: does the screen reader announce product names, prices, and add-to-cart buttons correctly?

Low-cost -- for stores ready to invest

- axe DevTools (free tier): Browser extension that runs automated WCAG checks
- Lighthouse (built into Chrome DevTools): Generates an accessibility score with specific fixes
- SiteImprove (free tier): Scans for accessibility issues across your entire store

Professional -- for enterprise contracts

- Manual accessibility audit by a qualified agency (\$500-5,000)
- Automated monitoring tools (\$10-200/month)
- VPAT (Voluntary Product Accessibility Template) -- often requested by enterprise buyers

Testing frequency

Phase	Frequency
-----	-----
Automated scan	Every theme update or major change
Manual keyboard test	Monthly
Full audit	Quarterly
Accessibility statement review	Every 3 months, or after any material change

Chapter 9: Maintaining Compliance Over Time

EAA compliance is not a one-time project. Your store changes -- new products, new theme versions, new apps -- and each change can introduce new issues.

Build compliance into your workflow

- New product check: Before publishing a new product, verify alt text on all images, check that descriptions don't use colour alone to convey information, and test the product page with Tab navigation.
- Theme updates: Before applying a theme update, test critical pages (homepage, a product page, checkout) with your scanner and keyboard.
- App reviews: Before installing a Shopify app, check whether it uses custom UI elements (modals, sliders, dropdowns) that could introduce accessibility issues. Test each new app.
- Regular scans: Run an automated scan monthly. Set a calendar reminder.
- Statement reviews: Update your accessibility statement every quarter and any time there's a material change.

What happens when enforcement comes

EAA enforcement varies by member state, but the pattern is consistent:

- A customer or competitor files a complaint with the national supervisory authority
- The authority contacts you and requests evidence of compliance attempts
- If your statement is current and shows ongoing remediation, you get a notice period
- If you have no statement and no evidence of effort, the fine is larger

The presence of an up-to-date accessibility statement showing active remediation substantially reduces penalty risk. Most authorities give remediation time if you can show good faith.

Record keeping

Keep records of:

- Accessibility scan results (date, score, issues found)
- Remediation actions taken
- Accessibility statement versions (date and content)
- Any accessibility-related correspondence

These records serve as evidence of good faith if you are ever investigated.

Appendix A: EAA Compliance Quick Checklist

Theme and layout

- [] All text meets WCAG 2.1 AA contrast (4.5:1 normal, 3:1 large)
- [] Colour alone is not used to convey information (price changes, sale indicators)
- [] Page has proper heading hierarchy (one h1, followed by h2, h3)
- [] Skip-to-content link is present at the top of the page
- [] Focus indicators visible on all interactive elements
- [] Page language is set in

Images and media

- [] All product images have descriptive alt text
- [] Decorative images have empty alt text (alt="")
- [] No images of text used where real text could be used
- [] Videos have captions or transcripts

Navigation and interactions

- [] All functionality works with keyboard alone
- [] No keyboard traps in modals, drawers, or quick-view
- [] Focus order matches visual order (left to right, top to bottom)
- [] Link text is unique and descriptive (no "click here" or "read more")
- [] Dropdown menus work with keyboard (Enter opens, Escape closes)
- [] Colour swatches show selected state with text/border, not colour alone

Forms and checkout

- [] Every form field has a visible element
- [] Error messages are linked to the field via aria-describedby
- [] Required fields are clearly marked (visual + text for screen readers)
- [] Autocomplete attributes are set on checkout fields
- [] CAPTCHA has audio alternative or uses invisible version

Documentation

- [] Accessibility statement is published at /pages/accessibility-statement
- [] Statement includes all required fields (see Chapter 7)
- [] Statement is linked from footer
- [] Statement is reviewed quarterly
- [] PDF receipts and invoices are accessible (text-based, not scanned images)

Appendix B: Resources

Free testing tools

- EAA/WCAG Scanner: hermes-passiv.pages.dev/scan -- 16 automated checks
- Ask the AI compliance assistant: <https://hermes-passiv.pages.dev/books/ea-shopify#bookAi> -- free answers to EAA questions, no signup
- WAVE Browser Extension: wave.webaim.org -- visual overlay of issues
- WebAIM Contrast Checker: webaim.org/resources/contrastchecker/
- aXe DevTools: deque.com/axe/devtools/ -- browser extension
- Lighthouse: Built into Chrome DevTools → Lighthouse → Accessibility
- NVDA Screen Reader: nvaccess.org (Windows, free)
- VoiceOver: Built into macOS (System Settings → Accessibility → VoiceOver)

Official resources

- EAA Directive Text: eur-lex.europa.eu (Directive 2019/882)
- EN 301 549: etsi.org -- the European accessibility standard
- WCAG 2.1 Quick Reference: w3.org/WAI/WCAG21/quickref/

- Shopify's accessibility documentation: community.shopify.com

About the author

This guide is published by Mahope, an EU-focused publishing imprint specialising in practical compliance resources for small web agencies and e-commerce businesses. Our tools and e-books are designed for teams without dedicated legal or compliance departments.

Cookie Consent & Privacy Compliance for Small Websites

From: Complete EU Compliance Bundle — Version 1.0

Cookie Consent & Privacy Compliance for Small Websites

A Practical Guide to Meeting GDPR, ePrivacy, and Cookie Requirements Without a Legal Team

If your website serves visitors in the European Union, you need cookie consent -- even if you don't think you use cookies. This guide explains exactly what the law requires, how to audit your site, and how to implement compliant consent without overcomplicating things.

Chapter 1: What the Law Actually Requires

Three pieces of EU law govern cookies and privacy on websites:

GDPR (General Data Protection Regulation)

The GDPR -- Regulation (EU) 2016/679 -- applies whenever you collect or process personal data from EU residents. Personal data includes IP addresses, which means virtually every website tracking visitor behaviour is affected.

What you need: A lawful basis for processing personal data. For most cookie-related processing, consent is the appropriate basis.

ePrivacy Directive (Cookie Law)

The ePrivacy Directive -- Directive 2002/58/EC, amended in 2009 -- is the specific law that requires cookie consent. Article 5(3) states that storing or accessing information on a user's device requires their consent, unless it's strictly necessary for the service.

What you need: Active, informed consent before non-essential cookies are set. A "by continuing to use this site you accept cookies" banner is not compliant.

The interplay

The GDPR sets the general data protection framework. The ePrivacy Directive sets the specific rules for electronic communications. Both must be satisfied. In practice: you need GDPR-compliant consent (freely given, specific, informed, unambiguous) for ePrivacy-covered tracking.

National implementations

EU directives are implemented at the member state level, which means minor differences apply:

Country	Cookie law reference	Key specificity
-----	-----	-----
Denmark	Cookie Executive Order	First cookie must be declined on landing, explicit consent before analytics
Germany	TTDSG	Strict "only strictly necessary without consent" approach
France	RGPD + CNIL guidelines	Heavily enforced; CNIL actively fines non-compliant sites

Netherlands	Telecommunicatiewet	Similar to Danish approach; consent before analytics
UK (post-Brexit)	PECR (UK)	Similar to ePrivacy, enforced by ICO

Chapter 2: Cookies and Tracking Technologies -- An Inventory

Before you can implement compliant consent, you need to know what's running on your site.

Categories of cookies

Category	Examples	Requires consent?
-----	-----	-----
Strictly necessary	Session cookies, CSRF tokens, shopping cart	No — exemption for "strictly necessary"
Functional/preference	Language preference, theme choice	Yes — unless genuinely necessary for the service
Analytics	Google Analytics, Plausible, Fathom	Yes — IP addresses are personal data
Marketing/tracking	Facebook Pixel, Google Ads, TikTok	Yes — always requires consent
Third-party embeds	YouTube videos, Google Maps, Twitter feed	Yes — these set third-party cookies

Analysing your site

- Browser DevTools > Application > Storage > Cookies: Lists all cookies your browser stores for the current site. Check after loading a page.
- Cookie audit tools: Free tools like Cookiebot's scanner, CookieYes scanner, or the PrivacyBee extension can scan your site for cookies.
- Check your third-party scripts: Google Analytics, Facebook Pixel, Hotjar, Microsoft Clarity, and most analytics tools set cookies. So do embedded YouTube videos, Google Maps, Typeform embeds, and social media widgets.

Document your findings

Create a simple table:

Cookie name	Provider	Purpose	Category	Expiry	Requires consent?
-----	-----	-----	-----	-----	-----
_ga	Google	Analytics	Analytics	2 years	Yes
_gid	Google	Analytics	Analytics	24 hours	Yes
PHPSESSID	Self	Session	Necessary	Session	No

This becomes your cookie policy.

Chapter 3: Building a Compliant Cookie Banner

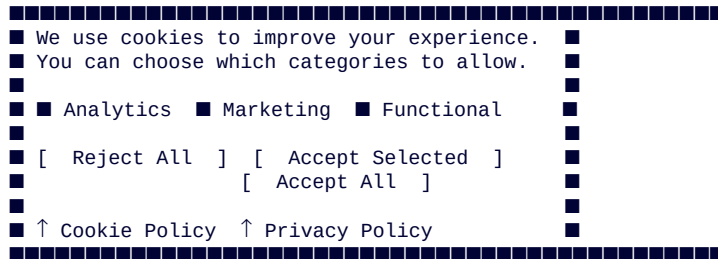
The cookie banner is the first thing many visitors see. It's also the most visibly regulated element.

What a compliant banner needs

- Clear information about what cookies are used and why -- not just "we use cookies"

- Granular consent options -- separate toggles for analytics, marketing, functional. At minimum: analytics vs. marketing.
- Accept all / Reject all -- both must be equally easy. A "reject all" button hidden behind a "settings" link is not compliant.
- No pre-checked boxes -- consent must be active, not assumed.
- No cookie walls -- EDPB guidelines state that denying consent should not deny access to the website. Blocking all content until the user accepts is not compliant.

What the banner looks like



Implementation options

Free options

- Cookie Consent by Osano (open source, free): A simple, lightweight JavaScript library. MIT license, works with any website builder or CMS. Basic compliance.
- Manual implementation: If your site uses very few cookies (e.g., only analytics), you can pause analytics scripts until the user accepts. This requires some JavaScript knowledge.
- Cloudflare Zaraz Auto-consent (if using Cloudflare): Manages cookie consent for Zaraz-deployed scripts. Free tier available.

Paid options (\$5-30/month)

- Cookiebot (by Cybot): Comprehensive scanner + banner. The most widely used. Free for small sites (< 100 pages). Paid plans start at €12/month.
- CookieYes: GDPR-compliant banner + scanner. Free for one domain. Paid from \$10/month.
- CookieFirst: EU-hosted, GDPR-compliant. Free for one domain with limited features. Paid from €9/month.

Platform-specific

- Shopify: Several apps (Shopify's built-in cookie banner, CookieYes for Shopify, Cookiebot)
- WordPress: Plugins like Complianz, GDPR Cookie Consent, Cookiebot
- Wix: Built-in cookie consent (enable in Site Settings)
- Squarespace: Built-in cookie banner (enable in Settings → Cookies)

JavaScript snippet approach

For custom sites, the pattern is:

```
// Only load analytics if user consented
function loadAnalytics() {
  window.dataLayer = window.dataLayer || [];
  // ... Google Analytics loading code
}

// Run when user clicks "Accept"
```

```

document.getElementById('accept-btn').addEventListener('click', function() {
    document.cookie = 'cookie_consent=accepted; max-age=31536000';
    loadAnalytics();
    hideBanner();
});

// Check existing consent
if (document.cookie.includes('cookie_consent=accepted')) {
    loadAnalytics();
}

```

Chapter 4: Consent at Scale -- Record-Keeping

The GDPR requires you to demonstrate compliance. This means documenting what consent you received, when, and how.

What to record

For each consent event, record:

- Unique user identifier (hashed, not raw IP)
- Timestamp of consent
- Which categories were accepted/rejected
- The exact text of the banner shown at the time of consent
- The URL where consent was given
- Proof of user action (clicked "Accept" vs. clicked "Reject")

Storage methods

Method	Pros	Cons
-----	-----	-----
Cookie-based	Simple, no backend needed	Limited to cookie lifetime, not GDPR-proof for enforcement
Local Storage	Simple, no backend	Cleared by user, not permanent
Server-side database	Permanent, fully auditable	Requires backend, more complex
Third-party service	Hosted, maintained, auditable	Monthly cost, dependency

For most small sites, cookie-based consent logging is sufficient if you also keep a log of the consent version shown. The most important record is not the individual consent -- it's the fact that your system only loaded tracking scripts after consent was given.

Consent versioning

When you update your cookie policy or change your tracking tools, all existing consents become invalid. You need a mechanism to re-consent users:

- Add a version number to your consent cookie
- When your policy changes, increment the version
- Show the banner again to users with an older consent version
- Record the new consent

Chapter 5: Privacy Policy -- What Must Be In It

The GDPR requires specific information in your privacy policy. This is separate from the cookie banner but linked.

Required information under Article 13

- Identity and contact details of the data controller (you)
- Contact details of your DPO (if applicable)
- Purposes of processing (why you collect data)
- Lawful basis for processing (consent, legitimate interest, contract, etc.)
- Recipients of personal data (Google, Facebook, Stripe, etc.)
- International transfer safeguards (if you use US-based services, this applies)
- Data retention periods
- The data subject's rights (access, rectification, erasure, restriction, portability, objection)
- Right to withdraw consent at any time
- Right to lodge a complaint with a supervisory authority
- Whether providing data is a contractual requirement (e.g., checkout data)
- Automated decision-making and profiling (if applicable)

Sample structure

```
# Privacy Policy for [Site Name]

## 1. Who we are
[Your business name, address, email, VAT number]

## 2. What data we collect and why
- **Account data**: [description]
- **Purchase data**: [description]
- **Analytics data**: [description]
- **Marketing data**: [description]

## 3. Legal basis
[Consent, legitimate interest, contract – specify which for each category]

## 4. Data sharing
[List third parties, with links to their privacy policies]

## 5. International data transfers
[If you use US services, list the safeguards (SCCs, DPF certification)]

## 6. How long we keep data
[Retention periods per data category]

## 7. Your rights
List all eight rights under GDPR.

## 8. Cookies
[Link to your cookie policy/statement]

## 9. Changes to this policy
[Versioning and update notification]

## 10. Contact
[Email, address]
```

Free privacy policy generators

- GDPR.eu Privacy Policy Generator: Free, basic template
- SECURi Privacy Policy Generator: Free, UK/EU compliant
- Termly: Tiered, free for basic
- iubenda: Freemium, good for small sites

Chapter 6: Handling Data Subject Requests

The GDPR gives individuals the right to request access to their data, request deletion, and more. You must respond within one month.

Types of requests

Request type	Description	Response time
-----	-----	-----
Right to access (SAR)	"Give me all data you have about me"	1 month
Right to erasure	"Delete all data about me"	1 month
Right to rectification	"Fix this incorrect data"	1 month
Right to restriction	"Stop processing my data"	Without delay

Handling a request

- **Verify identity:** Ask for identifying information before releasing data. Only send data to the requestor's verified email address.
- **Acknowledge receipt:** Within 3 business days, confirm you received the request and will respond.
- **Locate the data:** Check your analytics, email marketing, CRM, database, and any third-party services.
- **Compile and respond:** For an access request, provide a structured, machine-readable format (commonly CSV or JSON). For erasure, confirm deletion and ask third parties to do the same.
- **Document:** Keep a record of the request, your response, and the timeline.

Templates

Request acknowledgment:

Subject: Confirmation of Data Subject Request

>

Dear [Name],

>

We confirm receipt of your request to [access/delete/rectify] your personal data on [date]. We will respond within one month. If we need additional information to verify your identity, we will contact you separately.

>

Regards,

[Your Name]

Access response:

Subject: Response to Data Subject Access Request

>

Dear [Name],

>

In response to your request dated [date], please find attached the personal data we hold about you in CSV format.

>

The data includes: [list categories].

>

If you believe any data is incorrect, please contact us to request rectification.

>

Regards,

[Your Name]

Chapter 7: Analytics Without Consent Violations

Most small websites use analytics. Most analytics tools set cookies. Here's how to use analytics compliantly.

Cookie-less analytics options

These analytics tools work without cookies and therefore don't require consent:

Tool	Price	Notes
-----	-----	-----
Plausible	€9/month	Privacy-focused, cookie-less, EU-hosted
Fathom	€12/month	Cookie-less, privacy-first
Matomo (on-premise)	Free	Self-hosted, can be configured cookie-less
Pirsch	€7/month	German-made, GDPR-compliant
Cloudflare Web Analytics	Free	Basic analytics, no cookies

Configuring Google Analytics for consent

If you use Google Analytics and want it to work with consent, you need:

- Consent Mode v2: Google's API that lets GA wait for consent before using cookies. Configure it to respect user choices.
- Disable data sharing: In GA settings, disable "Google signals data" and data sharing with Google products.
- IP anonymization: Set anonymizelp: true in your GA config.
- Disable advertising features: Turn off remarketing, advertising reporting features, and demographics.

Even with these measures, Google Analytics requires consent under the ePrivacy Directive because it stores a client ID in a cookie.

The simplest compliant setup

- Install a consent management platform (Cookiebot, CookieYes, or the free Osano script)
- Configure it to block Google Analytics scripts until consent is given

- Consider Plausible or Fathom as a no-consent alternative for basic analytics
- If you must use Google Analytics: enable Consent Mode v2, anonymize IPs, and disable all advertising features

Chapter 8: Common Compliance Traps

Using "legitimate interest" for analytics

Some sites claim "legitimate interest" as a basis for analytics cookies instead of consent. This is contested. The ePrivacy Directive clearly requires consent for non-essential cookies. Most data protection authorities (DPAs) in the EU reject legitimate interest for analytics cookies. Using it is a bet against enforcement. The safer -- and more honest -- approach is consent.

Implied consent banners

"If you continue browsing, you accept cookies" -- this is not compliant. The user must take an active, affirmative action. Banners that imply consent by inaction are a common cause of CNIL (France) and Datatilsynet (Denmark) fines.

Google Fonts and third-party embeds

Google Fonts can transmit IP addresses to Google servers. YouTube embeds set cookies. Google Maps embeds set cookies. Every third-party embed should be evaluated for data collection.

Cookie walls

Blocking all content until the user accepts cookies (a "cookie wall") is not compliant per the EDPB's 2020 guidelines. Users must have genuine choice. However, this area is evolving -- some paid content sites make a case for legitimate refusal. For most small sites: don't do it.

No cookie policy

Having a cookie banner without a linked, detailed cookie policy is a compliance gap. The banner offers "more information" links -- those should actually lead to information.

Outdated consent

If you update your tracking setup (adding the Facebook Pixel, adding Hotjar, adding a new analytics tool), your existing consent is no longer valid. You must re-consent all users.

Appendix A: Quick Implementation Checklist

Cookie banner

- ☐ Banner appears before any non-essential scripts load
- ☐ Banner explains what cookies are used for
- ☐ Granular on/off toggles for each category (analytics, marketing, functional)
- ☐ Accept All and Reject All buttons are equally prominent
- ☐ No pre-checked boxes
- ☐ No cookie wall (site is accessible without accepting)
- ☐ Banner links to cookie policy and privacy policy
- ☐ Banner respects "Do Not Track" where applicable

Cookie policy

- ☐ Lists every cookie by name, provider, purpose, category, and expiry
- ☐ Explains how to manage cookie preferences
- ☐ State of banner text displayed
- ☐ Includes date of last update

Privacy policy

- ☐ Covers all Article 13 required information
- ☐ Identifies data controller (business name, address, contact)
- ☐ Lists all third parties with whom data is shared
- ☐ Specifies lawful basis for each processing activity
- ☐ Describes data subject rights
- ☐ Describes international transfer safeguards (if applicable)
- ☐ Includes date of last update

Operations

- ☐ Consent records stored (cookie + timestamp + version)
- ☐ Consent versioning system in place
- ☐ Procedure for handling data subject requests documented
- ☐ Privacy policy and cookie policy reviewed at least annually
- ☐ Analytics configured to respect consent
- ☐ Third-party services evaluated for data collection

Appendix B: Resources

Free tools

- Osano Cookie Consent: opensource.cookieconsent.osano.com -- free, open-source banner
- GDPR.eu Privacy Policy Generator: gdpr.eu -- free template
- Cookie audit: Use browser DevTools > Application > Storage > Cookies
- EAA/WCAG Scanner: hermes-passiv.pages.dev/scan

Low-cost solutions

- Cookiebot: €12/month, comprehensive
- CookieYes: \$10/month, good for small sites
- Plausible Analytics: €9/month, cookie-less
- Fathom Analytics: €12/month, cookie-less

Official sources

- GDPR text: eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679
- ePrivacy Directive: eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32002L0058
- EDPB Guidelines 05/2020: edpb.europa.eu -- on consent under GDPR
- Cookie consent guidance (UK ICO): ico.org.uk

About the author

This guide is published by Mahope, an EU-focused publishing imprint specialising in practical compliance resources for small web agencies and e-commerce businesses.

Free tools from the publisher

- Cookie banner checker -- see every cookie and tracker a site loads, before and after consent:
<https://hermes-passiv.pages.dev/cookie-check>
- Ask the AI compliance assistant -- free answers to GDPR/cookie questions, no signup:
<https://hermes-passiv.pages.dev/books/cookie-consent-guide#bookAi>

Build Your First Chrome Extension

From: Complete EU Compliance Bundle — Version 1.0

Build Your First Chrome Extension -- A Complete Practical Guide

By the Clean Copy team · Version 1.0

Preface: Why This Book Exists

Every developer I know has had the same thought at some point: "I wish my browser could just do X." Maybe it's copying text without formatting, checking a page for accessibility issues, or automating a repetitive task. Chrome extensions are how you build that.

I wrote this book because when I built my first extension (Clean Copy -- an HTML-to-Markdown converter that runs in your browser), I had to piece together information from a dozen sources. The official Chrome docs are thorough but scattered. Stack Overflow answers assume you already know the architecture. Blog posts go out of date when Google updates the platform.

This book is what I wish I'd had: a single, practical, step-by-step guide that takes you from "I've never built an extension" to "my extension is live in the Chrome Web Store."

Who This Book Is For

- Web developers who know HTML, CSS, and basic JavaScript
- Anyone who has thought "I could build a browser extension for that"
- Developers who want to publish their first Chrome Web Store listing

You don't need experience with service workers, the Chrome API, or browser extension architecture. You'll learn all of that here.

What You Will Build

By the end of this book, you'll have built and published a working Chrome extension. Along the way, you'll create:

- A popup that interacts with the current page
- A context menu that copies formatted text
- A background service worker that handles events
- An options page that persists user settings
- A keyboard shortcut for power users

The final extension is Clean Copy Lite -- a simplified version of the real Clean Copy extension. It converts selected HTML into clean Markdown, ready to paste into any document.

What You Need

- A computer running Chrome or Edge (Chromium-based)
- A text editor (VS Code recommended)
- Basic familiarity with JavaScript, HTML, and CSS
- About 2-3 hours to build the complete extension

Chapter 1: Understanding Chrome Extension Architecture

1.1 What Makes an Extension Different from a Website

A Chrome extension is a collection of files -- HTML, CSS, JavaScript, images -- packaged together and loaded by the browser. Unlike a website which runs on a remote server, an extension runs locally in the user's browser.

This fundamental difference shapes everything: there's no backend server, no database (unless you use one), and the extension can only interact with the web through the Chrome API. But it also means your extension works offline, has near-zero latency, and costs nothing to host.

1.2 The Three Core Components

Every Chrome extension has at least these three files:

`manifest.json` -- The heart of your extension. This JSON file tells Chrome what your extension is called, what permissions it needs, which files to load, and what version it is. It's the first file Chrome reads when installing your extension.

`background.js` -- A service worker that runs in the background, listening for events. Think of it as the brain of your extension: it handles context menu clicks, keyboard shortcuts, messages from the popup, and any long-running logic. It does NOT have access to the DOM of web pages.

`popup.html` + `popup.js` -- The UI that appears when the user clicks your extension's toolbar icon. This is a regular HTML page (with JavaScript) that can interact with the currently active tab through the Chrome API. The popup only exists while it's open -- once the user clicks elsewhere, it's destroyed.

There are other components too (options pages, content scripts, offscreen documents), but these three are where you start.

1.3 Manifest V3 (MV3)

As of 2023, Chrome requires all new extensions to use Manifest V3 (MV3). The key changes from the older MV2:

- Background pages are replaced by service workers (which can be killed and restarted by Chrome)
- Remote code execution is banned (all code must be in the extension package)
- Network request modification uses `declarativeNetRequest` instead of the old `webRequest` API
- Service workers don't have DOM access (no document, no window)

MV3 is more restrictive but also more secure. This book covers only MV3.

1.4 The Extension Lifecycle

- User installs your extension from the Chrome Web Store
- Chrome reads `manifest.json` and registers all components
- The background service worker starts when needed (on install, on a triggered event)
- The popup loads when the user clicks the toolbar icon
- The extension runs until Chrome decides to unload it (or the user uninstalls it)

Extensions do NOT run continuously. The service worker starts on demand and stops after ~30 seconds of inactivity. This is critical to understand: you cannot rely on persistent state in memory.

1.5 Permissions: Only What You Need

Every extension must declare its permissions in `manifest.json`. Chrome shows these to the user before installation. Common permissions:

```
{
  "permissions": ["activeTab", "contextMenus", "storage", "clipboardWrite"],
  "host_permissions": ["https://*/"]
}
```

The golden rule: ask for the minimum permissions you actually need. Users are wary of extensions that request access to "all websites" or "all your data." The activeTab permission is your best friend -- it gives temporary access to the current tab only when the user interacts with your extension.

Chapter 2: Your First Extension -- Hello World

Let's build something real. Open your text editor and create a new folder called clean-copy-lite.

2.1 Create manifest.json

Inside clean-copy-lite/, create manifest.json:

```
{
  "manifest_version": 3,
  "name": "Clean Copy Lite",
  "version": "1.0.0",
  "description": "Copy selected text as clean Markdown – right-click and paste anywhere.",
  "permissions": ["activeTab", "contextMenus", "clipboardWrite"],
  "background": {
    "service_worker": "background.js"
  },
  "action": {
    "default_popup": "popup.html",
    "default_title": "Clean Copy Lite"
  },
  "icons": {
    "16": "icon-16.png",
    "48": "icon-48.png",
    "128": "icon-128.png"
  }
}
```

Note: We declare clipboardWrite so our extension can copy to the clipboard. activeTab gives access to the current page when the user invokes the extension. contextMenus lets us add a right-click option.

2.2 Create the Background Service Worker

Create background.js:

Wait -- we're getting ahead of ourselves. We need the offscreen document too. Let's create it.

Create offscreen.html:

```
<!DOCTYPE html>
<html>
<head><meta charset="utf-8"></head>
<body>
<script>
// Wait for the background to tell us what to copy
chrome.runtime.onMessage.addListener((request, sender, sendResponse) => {
  if (request.type === 'copy-from-storage') {
    chrome.storage.local.get('_clipboard', (data) => {
      if (data._clipboard) {
        navigator.clipboard.writeText(data._clipboard).catch(console.error);
        chrome.storage.local.remove('_clipboard');
      }
    });
  }
});
</script>
</body>
</html>
```

The offscreen document is an MV3 workaround: service workers can't call navigator.clipboard.writeText(), so we create a hidden HTML page that has DOM access and do the clipboard operation there.


```

padding: 8px;
border: 1px solid #ccc;
border-radius: 4px;
box-sizing: border-box;
resize: vertical;
}
textarea:focus {
outline: none;
border-color: #666;
}
.output {
margin-top: 8px;
min-height: 60px;
padding: 8px;
background: #f5f5f5;
border-radius: 4px;
font-size: 13px;
white-space: pre-wrap;
word-break: break-all;
}
button {
margin-top: 8px;
padding: 6px 16px;
background: #1a73e8;
color: white;
border: none;
border-radius: 4px;
cursor: pointer;
font-size: 13px;
}
button:hover {
background: #1557b0;
}
button.copy-btn {
background: #34a853;
}
.status {
font-size: 12px;
color: #666;
margin-top: 6px;
}
</style>
</head>
<body>
<h2>Clean Copy Lite</h2>
<textarea id="input" placeholder="Paste HTML or messy text here..."></textarea>
<button id="convert">Convert to Markdown</button>
<div class="output" id="output"></div>
<button class="copy-btn" id="copy" style="display:none">Copy to Clipboard</button>
<div class="status" id="status"></div>

<script src="popup.js"></script>
</body>
</html>

```

3.2 Create popup.js

```

// Clean Copy Lite – Popup Logic
document.getElementById('convert').addEventListener('click', async () => {
const input = document.getElementById('input').value;
if (!input.trim()) {
document.getElementById('status').textContent = 'Paste some text first.';
return;
}

// Send to background for conversion
chrome.runtime.sendMessage(
{ type: 'convert-to-markdown', html: input },
(response) => {
if (chrome.runtime.lastError) {
document.getElementById('status').textContent = 'Error: ' + chrome.runtime.lastError.message;
return;
}
document.getElementById('output').textContent = response.markdown;
document.getElementById('copy').style.display = 'inline-block';
document.getElementById('status').textContent = 'Converted!';
}
});

```

```

});

document.getElementById('copy').addEventListener('click', async () => {
  const text = document.getElementById('output').textContent;

  // Store and trigger offscreen copy
  await chrome.storage.local.set({ _clipboard: text });
  try {
    await chrome.offscreen.createDocument({
      url: 'offscreen.html',
      reasons: ['CLIPBOARD'],
      justification: 'Copy Markdown text to clipboard'
    });
  } catch (e) {}
  chrome.runtime.sendMessage({ type: 'copy-from-storage' });
  document.getElementById('status').textContent = 'Copied!';
});

```

3.3 Add the Message Handler to background.js

Add this to your background.js (inside the `chrome.runtime.onInstalled.addListener` or as a separate listener):

```

// Handle messages from the popup
chrome.runtime.onMessage.addListener((request, sender, sendResponse) => {
  if (request.type === 'convert-to-markdown') {
    const markdown = htmlToMarkdown(request.html);
    sendResponse({ markdown });
    return true; // Keep the channel open for async response
  }
  if (request.type === 'copy-from-storage') {
    // Already handled by offscreen.html – but send a response to avoid warnings
    sendResponse({ ok: true });
  }
});

```

Now reload your extension (`chrome://extensions` → reload icon). Click the extension toolbar icon, paste some HTML, and click "Convert to Markdown."

Chapter 4: Keyboard Shortcuts

Power users love keyboard shortcuts. Chrome extensions can declare shortcuts in `manifest.json` and handle them in the background.

4.1 Add Commands to manifest.json

Add to `manifest.json`:

```

"commands": {
  "copy-selection": {
    "suggested_key": {
      "default": "Alt+C",
      "mac": "MacCtrl+C"
    },
    "description": "Copy selected text as Markdown"
  }
}

```

Note: On Mac, Chrome extensions cannot use `Cmd+C` (it's reserved). Use `MacCtrl+C` instead.

4.2 Handle the Command in background.js

```

chrome.commands.onCommand.addListener(async (command) => {
  if (command === 'copy-selection') {
    const [tab] = await chrome.tabs.query({ active: true, currentWindow: true });
    if (!tab) return;

    chrome.scripting.executeScript({
      target: { tabId: tab.id },
      function: getSelectedHtml
    });
  }
});

```

```

}, async (results) => {
  if (chrome.runtime.lastError || !results[0].result) return;
  const markdown = htmlToMarkdown(results[0].result);

  // Store and copy via offscreen document
  await chrome.storage.local.set({ _clipboard: markdown });
  try {
    await chrome.offscreen.createDocument({
      url: 'offscreen.html',
      reasons: ['CLIPBOARD'],
      justification: 'Copy Markdown to clipboard via keyboard shortcut'
    });
  } catch (e) {}
  chrome.runtime.sendMessage({ type: 'copy-from-storage' });
});
});

```

Reload the extension. Select some text on any page, press Ctrl+Alt+C (Windows/Linux) or Ctrl+MacCtrl+C (Mac), then paste. The text should be converted to Markdown automatically.

Chapter 5: Persisting User Settings

Users want to customize their experience. Chrome's storage API makes this easy.

5.1 Add Storage Permission

Update manifest.json permissions:

```

"permissions": [
  "activeTab", "contextMenus", "clipboardWrite",
  "scripting", "offscreen", "storage"
]

```

5.2 Custom Cleanup Rules

Let's let users define custom find-and-replace rules. Update background.js to read rules from storage:

```

// Load custom rules on startup and whenever storage changes
let customRules = [];

async function loadCustomRules() {
  const data = await chrome.storage.local.get('customRules');
  customRules = (data.customRules || []).filter(r => r.pattern && r.replacement);
}

// Apply custom rules to text
function applyCustomRules(text) {
  for (const rule of customRules) {
    try {
      const regex = new RegExp(rule.pattern, 'g');
      text = text.replace(regex, rule.replacement);
    } catch (e) {
      // Skip invalid regex – never break copying
      console.warn('Invalid rule skipped:', rule.pattern);
    }
  }
  return text;
}

// Load rules initially
loadCustomRules();

// Reload rules when they change
chrome.storage.onChanged.addListener((changes) => {
  if (changes.customRules) {
    customRules = (changes.customRules.newValue || []).filter(r => r.pattern && r.replacement);
  }
});

```

Now update the copy chain to apply custom rules before writing to clipboard. Wherever you call `copyToClipboard`, pass the text through `applyCustomRules` first:


```
// Instead of: copyToClipboard(markdown, tab.id);
// Do: copyToClipboard(applyCustomRules(markdown), tab.id);
```

5.3 Saving the User's Mode Preference

Some users always want plain text instead of Markdown. Let's persist that choice:

```
// In the popup:
chrome.storage.local.get('plainMode', (data) => {
  if (data.plainMode) {
    document.getElementById('plain-mode').checked = true;
  }
});

document.getElementById('plain-mode').addEventListener('change', (e) => {
  chrome.storage.local.set({ plainMode: e.target.checked });
});
```

Add a checkbox to popup.html:

```
<label>
  <input type="checkbox" id="plain-mode">
  Plain text only (strip all Markdown formatting)
</label>
```

Chapter 6: The Options Page

An options page gives users a place to configure your extension outside the popup.

6.1 Create *options.html*

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    body {
      font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', sans-serif;
      max-width: 600px;
      margin: 40px auto;
      padding: 0 20px;
    }
    h1 { font-size: 24px; }
    .rule {
      display: flex;
      gap: 8px;
      margin-bottom: 8px;
      align-items: center;
    }
    .rule input {
      flex: 1;
      padding: 6px;
      border: 1px solid #ccc;
      border-radius: 4px;
      font-family: monospace;
    }
    .rule button {
      padding: 6px 10px;
      background: #e74c3c;
      color: white;
      border: none;
      border-radius: 4px;
      cursor: pointer;
    }
    .add-btn {
      padding: 6px 16px;
      background: #1a73e8;
      color: white;
      border: none;
      border-radius: 4px;
      cursor: pointer;
      margin-top: 8px;
    }
  </style>
</head>
<body>
  <h1>Options</h1>
  <div>
    <div>
      <input type="checkbox" /> Plain text only
    </div>
    <button>Save</button>
  </div>
  <button>Add</button>
</body>
</html>
```

```

    .save-btn {
      padding: 8px 24px;
      background: #34a853;
      color: white;
      border: none;
      border-radius: 4px;
      cursor: pointer;
      margin-top: 16px;
      font-size: 14px;
    }
    .status { margin-top: 8px; color: #666; }
  </style>
</head>
<body>
  <h1>Clean Copy Lite Settings</h1>
  <p>Custom find-and-replace rules are applied after every copy:</p>
  <div id="rules"></div>
  <button class="add-btn" id="add-rule">Add Rule</button>
  <br><br>
  <button class="save-btn" id="save">Save Rules</button>
  <div class="status" id="status"></div>

  <script src="options.js"></script>
</body>
</html>

```

6.2 Create options.js

```

let rules = [];

async function loadRules() {
  const data = await chrome.storage.local.get('customRules');
  rules = data.customRules || [];
  renderRules();
}

function renderRules() {
  const container = document.getElementById('rules');
  container.innerHTML = '';
  rules.forEach((rule, i) => {
    const div = document.createElement('div');
    div.className = 'rule';
    div.innerHTML = `
      <input class="pattern" value="${escHtml(rule.pattern || '')}" placeholder="Find (regex)">
      <input class="replacement" value="${escHtml(rule.replacement || '')}" placeholder="Replace">
      <button data-index="${i}" class="remove">x</button>
    `;
    container.appendChild(div);
  });

  document.querySelectorAll('.remove').forEach(btn => {
    btn.addEventListener('click', () => {
      const i = parseInt(btn.dataset.index);
      rules.splice(i, 1);
      renderRules();
    });
  });
}

function escHtml(s) {
  return s.replace(/&/g, '&amp;').replace(/"/g, '&quot;').replace(/</g, '&lt;').replace(/>/g, '&gt;');
}

document.getElementById('add-rule').addEventListener('click', () => {
  rules.push({ pattern: '', replacement: '' });
  renderRules();
});

document.getElementById('save').addEventListener('click', async () => {
  const inputs = document.querySelectorAll('.rule');
  const newRules = [];
  inputs.forEach(div => {
    const pattern = div.querySelector('.pattern').value.trim();
    const replacement = div.querySelector('.replacement').value;
    if (pattern) newRules.push({ pattern, replacement });
  });
  await chrome.storage.local.set({ customRules: newRules });
  document.getElementById('status').textContent = 'Saved! Rules will apply on next copy.';
}

```

```

    setTimeout(() => { document.getElementById('status').textContent = ''; }, 2000);
  });

loadRules();

```

6.3 Register the Options Page in manifest.json

```
"options_page": "options.html"
```

Now users can right-click your extension icon → "Options" to open the settings page.

Chapter 7: Icons and Branding

Your extension needs icons. At minimum: 16×16, 48×48, and 128×128 pixels. Chrome uses 16px for the toolbar, 48px for the extensions page, and 128px for the Web Store listing.

7.1 Creating Simple Icons

You can create simple icons with an HTML canvas. Create generate-icons.html and open it in a browser:

```

<!DOCTYPE html>
<html>
<body>
<script>
const sizes = [16, 48, 128];
sizes.forEach(size => {
  const canvas = document.createElement('canvas');
  canvas.width = size;
  canvas.height = size;
  const ctx = canvas.getContext('2d');

  // Background circle
  ctx.fillStyle = '#1a73e8';
  ctx.beginPath();
  ctx.arc(size/2, size/2, size/2 - 1, 0, Math.PI * 2);
  ctx.fill();

  // Letter C (simplified)
  ctx.fillStyle = 'white';
  ctx.font = `bold ${size * 0.6}px sans-serif`;
  ctx.textAlign = 'center';
  ctx.textBaseline = 'middle';
  ctx.fillText('C', size/2, size/2 + 1);

  // Download
  canvas.toBlob(blob => {
    const a = document.createElement('a');
    a.href = URL.createObjectURL(blob);
    a.download = `icon-${size}.png`;
    a.click();
  });
});
</script>
</body>
</html>

```

For production, use a proper icon (SVG converted to PNG, or a designer's asset). The key requirement: icons should be recognizable at 16px in the toolbar.

7.2 Extension Name and Description

Your extension name and description in manifest.json are shown in the Chrome Web Store and in chrome://extensions. Make them clear and descriptive:

- Name: "Clean Copy Lite -- HTML to Markdown" (descriptive)
- Description: "Copy selected text as clean, formatted Markdown. Right-click, copy, paste anywhere." (tells the user exactly what it does and how to use it)

Avoid keyword stuffing, but do include relevant terms users search for: "Markdown," "copy," "format," "clipboard."

Chapter 8: Publishing to the Chrome Web Store

8.1 Prepare Your Package

Chrome Web Store accepts a .zip file containing your extension. The zip should include:

- manifest.json
- background.js
- popup.html + popup.js
- offscreen.html
- options.html + options.js (optional)
- icon-16.png, icon-48.png, icon-128.png

Important: Remove any development-only files (test files, build scripts, .git folder). The zip must be under 500 MB.

Create the zip:

```
zip -r clean-copy-lite.zip * -x "**.git*" -x "**.md" -x "generate-icons.html"
```

8.2 Create a Developer Account

- Go to chrome.google.com/webstore/devconsole
- Sign in with your Google account
- Pay the one-time \$5 registration fee
- Accept the developer agreement

8.3 Upload Your Extension

- In the Developer Console, click "New item"
- Upload your .zip file
- Fill in the store listing:
 - Description: Write 2-3 paragraphs explaining what your extension does, why it's useful, and how to use it. Include the key features as bullet points.
 - Screenshots: At least one 1280×800 screenshot showing the popup in action. You can take this by opening the popup and using Chrome's screenshot tool.
 - Promotional images: Optional but recommended (a small tile for the Chrome Web Store homepage).
 - Category: "Productivity" or "Developer Tools"
 - Language: English
- Review the permission justification -- Chrome will ask you to explain each permission you request
- Submit for review

8.4 The Review Process

Chrome Web Store reviews typically take 1-3 business days. Common reasons for rejection:

- Insufficient permissions justification: If you request activeTab, explain "Only activated when user right-clicks or opens the popup." For clipboardWrite, say "Required to write formatted text to the

clipboard."

- Minimal functionality: The extension must provide real value. A simple "Hello World" will be rejected.
- Broken functionality: Make sure everything works before submitting
- Privacy concerns: Any data collection must be disclosed. If you don't collect data, say so explicitly.

For Clean Copy Lite, the permission justification could be:

"activeTab -- Only accesses the current tab's selected text when the user right-clicks and chooses 'Copy as Markdown' or opens the popup. No background data collection. clipboardWrite -- Required to write converted Markdown text to the system clipboard. scripting -- Required to inject a selection-reader script into the active tab at the user's request. storage -- Stores user's custom cleanup rules locally; no data is sent externally."

8.5 Handling Updates

When you update your extension:

- Increment the version field in manifest.json
- Build a new .zip
- Upload to the Developer Console
- Submit for review again

Minor updates (bug fixes) are usually reviewed within 24 hours. Major UI changes may take 1-3 days.

8.6 Post-Publish Checklist

After your extension is live:

- [] Verify the listing looks correct (visit the Chrome Web Store URL)
- [] Install from the store and test all features
- [] Check that screenshots are correct
- [] Respond to any user reviews within 48 hours
- [] Monitor the Developer Console for policy violation warnings

Chapter 9: Testing and Debugging

9.1 Inspecting the Service Worker

- Go to chrome://extensions
- Find your extension
- Click "Service Worker" (under "Inspect views")
- A DevTools window opens showing the background service worker's console

This is where you'll see console.log output from background.js.

9.2 Debugging the Popup

- Right-click your extension's toolbar icon
- Select "Inspect popup"
- DevTools opens for the popup HTML page

Note: The popup closes when it loses focus. To keep it open, click inside the DevTools window immediately after it opens.

9.3 Common Pitfalls

Problem	Cause	Solution
-----	-----	-----
Context menu doesn't appear	`contextMenus` permission missing	Add to manifest
Clipboard doesn't work	MV3 service worker can't use `navigator.clipboard`	Use offscreen document
Extension stops working after idle	Service worker was terminated	Use `chrome.storage` for persistent state
Popup shows nothing	Missing permissions or JS error	Check DevTools console
`executeScript` returns error	Missing `scripting` permission	Add to manifest
Keyboard shortcut doesn't work	Shortcut conflicts with browser	Check `chrome://extensions/shortcuts`

9.4 Simple Unit Tests

You can test your core logic (HTML-to-Markdown conversion) without Chrome. Create test.js:

```
// Copy htmlToMarkdown and stripTagsSafe here (or import them)
const assert = require('assert');

// Test cases
assert.strictEqual(
  htmlToMarkdown('<h1>Title</h1>').trim(),
  '# Title'
);

assert.strictEqual(
  htmlToMarkdown('<p>Hello <strong>world</strong></p>').trim(),
  'Hello **world**'
);

assert.strictEqual(
  htmlToMarkdown('<a href="https://x.com">link</a>').trim(),
  '[link](https://x.com)'
);

console.log('All tests passed!');
```

Run with Node.js: node test.js

These tests run in Node, not Chrome, so they're fast for development iteration. Keep a file of 5-10 core tests and run them after every change.

Chapter 10: From Lite to Production -- What's Next

You've built a working Chrome extension. Here's how to take it further:

10.1 Firefox Port

Chrome extensions are almost compatible with Firefox (which uses the WebExtensions API). Key differences:

- Manifest: Firefox needs browser_specific_settings.gecko.id
- Clipboard: Firefox lets background pages use navigator.clipboard.writeText() directly (no offscreen document needed)
- API prefix: Chrome uses chrome., Firefox also supports browser. (promise-based)
- MV3: Firefox MV3 uses event pages, not service workers

Tools like web-ext help with cross-browser development.

10.2 Premium Features

If you want to monetize your extension:

- Free + Pro model: Basic features are free, advanced features (batch conversion, custom rules, cloud sync) require a license
- License keys: Generate license keys via a service like Lemon Squeezy, validate against a remote API
- In-app purchase: Chrome Web Store supports one-time purchases and subscriptions

Example Pro features for Clean Copy:

- Custom cleanup rule sets (save and load)
- Batch convert multiple pages
- Cloud sync of settings across devices
- PDF export

10.3 Analytics (Privacy-First)

Know your users without compromising privacy:

- chrome.storage counters: Increment a counter on each copy
- Optional opt-in analytics: Ask users if they want to share anonymous usage data
- No tracking scripts: Never inject Google Analytics or similar into web pages

A simple counter:

```
chrome.storage.local.get('copyCount', (data) => {
  const count = (data.copyCount || 0) + 1;
  chrome.storage.local.set({ copyCount: count });
});
```

10.4 Your Extension's GitHub Repository

Publish your code on GitHub. A public repo:

- Builds trust (users can inspect the code)
- Enables community contributions
- Serves as a portfolio piece

Include:

- README with install instructions and screenshots
- MIT license
- .github/workflows/ for CI testing

Appendix A: Complete manifest.json Reference

```
{
  // Required
  "manifest_version": 3,
  "name": "Your Extension Name",
  "version": "1.0.0",

  // Recommended
  "description": "What your extension does in one sentence.",
  "icons": {
    "16": "icon-16.png",
    "48": "icon-48.png",
    "128": "icon-128.png"
  }
}
```

```

},
// Optional but common
"action": {
  "default_popup": "popup.html",
  "default_title": "Tooltip text"
},
"background": {
  "service_worker": "background.js"
},
"options_page": "options.html",
"permissions": ["activeTab", "storage", "contextMenus", "clipboardWrite", "scripting", "offscreen"],
"commands": {
  "my-command": {
    "suggested_key": { "default": "Alt+Shift+1" },
    "description": "Run my command"
  }
},
// Rarely needed
"host_permissions": ["https://*/*"],
"content_scripts": [{
  "matches": ["https://*/*"],
  "js": ["content.js"]
}]
}

```

Appendix B: Permissions Quick Reference

Permission	What It Allows	User Perception
-----	-----	-----
`activeTab`	Access the current tab on user action	Low concern — "only when I click"
`storage`	Read/write chrome.storage.local	Low — local only, no data sent
`contextMenus`	Add right-click menu items	Low — expected behavior
`clipboardWrite`	Write to system clipboard	Low — required for copy features
`clipboardRead`	Read from system clipboard	Medium — only if truly needed
`scripting`	Inject scripts into pages	Medium — explain clearly
`offscreen`	Create hidden documents	Low — MV3 technical requirement
`tabs`	Full tab info (URLs, etc.)	High — avoid when possible
``	Access all websites	Very high — avoid unless essential

Appendix C: Final Checklist

Before publishing, verify:

- ☐ Extension works without errors in chrome://extensions
- ☐ Context menu appears on text selection
- ☐ Popup opens and converts text
- ☐ Keyboard shortcut triggers the copy
- ☐ Options page saves and loads settings
- ☐ Icons are crisp at all 3 sizes
- ☐ All permissions are justified in the store listing
- ☐ Version number is updated

- [] Zip contains only necessary files
- [] Privacy policy is mentioned (even if "no data collected")

About the Author

This book was written by the team behind Clean Copy, an open-source browser extension that converts HTML to clean Markdown. Clean Copy is available for Chrome and Firefox, and has been downloaded and tested by developers worldwide.

The techniques in this book are battle-tested in a production extension with thousands of lines of code, multiple release versions, and real user feedback.

© 2026. This book is provided for educational purposes. Chrome is a trademark of Google LLC. All code examples are MIT-licensed and may be used freely.

More from the publisher

- Clean Copy -- a free browser extension that copies any web page as clean Markdown. Built with the exact techniques this book teaches: <https://hermes-passiv.pages.dev/clean-copy>

About This Bundle

The Complete EU Compliance Bundle brings together six practical guides written for small web agencies that serve EU clients. Each guide was researched and written to be immediately actionable — no legal background required.

Published August 2026 by Mahope / Clean Copy Publications.

Free edition: Creative Commons BY-NC 4.0.

Individual EPUB editions available at hermes-passiv.pages.dev/books