NOMAD SMALL BUSINESS SUITE

Nomad Support Desk Free - Implementation Guide

A deployment guide for installing, configuring, securing, testing and launching Nomad Support Desk Free, including customer and resource portals, inbound email, time billing, CRM and Accounting integration, customer sign-off and the embedded Ticket AI assistant.

Edition: 0.2.3
Audience: WordPress administrators, implementers and support managers

Contents

This guide is based on the verified product package and is written as an operational reference. Use the online version for navigation and this PDF for offline access.

1. Product architecture and scope

Understand what the Free edition installs and which capabilities are available before changing the site.

Edition scope

Nomad Support Desk Free is a complete standalone WordPress ticketing foundation. It stores tickets, replies, customers, resources, time entries, billing batches, signatures, email events and AI resources in WordPress database tables created by the plugin.

The embedded Ticket AI engine belongs to Service Desk Free. It does not call, license or require the separate AI Support Agent plugin.

- Admin ticket management and customer records
- Logged-in customer support portal
- Assigned-resource mobile portal
- Inbound email webhook
- Time capture and billing batches
- Customer/job-card signature capture
- CRM status tagging and Accounting Pro invoice integration when those products are detected
- Knowledge-only, web-only or combined Ticket AI analysis

Edition boundaries

Free does not provide Pro attachment management, customer email event notifications, the Outlook-friendly structured email template, quote/onsite workflows or customer analytics PDF downloads. Those are documented in the Pro guides.

Important Install only features that are actually present in the selected edition. Do not promise Pro workflows to Free users.

Core status and priority model

Type	Available values	Implementation use
Status	Open, Pending, Waiting on Customer, Resolved, Closed	Drives queues, analytics, CRM tags and lifecycle timestamps
Priority	Low, Normal, High, Urgent	Supports sorting, filtering and triage
Ticket number	NSD-1001 by default	Prefix and next number are configurable

Runtime and data ownership

The Free plugin owns the active runtime when Pro is not active. It creates and reads its own `nsd_` tables, registers the customer and resource shortcodes, schedules the billing task, and loads the embedded AI engine. All operational data remains in the site's WordPress database; the portal is a view over that data rather than a separate hosted service.

When Pro is later installed, Free must remain active as the required foundation, but Pro supplies the combined runtime. This prevents two copies of the same hooks and shortcodes from running at once. Implementers should therefore treat Free as the data foundation and Pro as an extension, not as two independent desks.

Component	Owner in Free-only installation	Operational consequence
Tickets, customers and replies	Service Desk Free	Back up the WordPress database
Customer and resource portals	Service Desk Free shortcodes	Protect the WordPress pages and user accounts
Embedded Ticket AI	Service Desk Free	Configure under Support Desk, not in AI Support Agent
CRM and Accounting links	Optional detected integrations	The desk continues when those plugins are absent

Lifecycle timestamps and reporting

The status is more than a label. The first public staff response populates the first-response time; moving to Resolved records the resolution time; moving to Closed records the closure time. Analytics use those timestamps, so testing should include deliberate status changes and a genuine public reply rather than editing records directly in the database.

Reporting rule	Agree the meaning of Pending, Waiting on Customer, Resolved and Closed before launch. Inconsistent use produces misleading response and resolution measures.

2. Requirements and pre-deployment planning

Confirm the WordPress environment, user model and operating workflow before activation.

Technical requirements

Requirement	Minimum or expectation	Verification
WordPress	6.0 or later	Check Dashboard > Updates
PHP	7.4 or later	Check Tools > Site Health > Info
HTTPS	Strongly recommended	Confirm portal and sign-off URLs use HTTPS
Permalinks	Any normal pretty-permalink structure	Save Settings > Permalinks once after launch
Outbound email	Working WordPress mail delivery	Send a controlled test message
Database backups	Current and restorable	Verify the backup before activation

Decisions to make

- Who may administer the desk and change settings
- Which WordPress users will become support resources
- Whether customers will self-register or be created by staff
- Which page will contain each shortcode
- Whether inbound email will create unknown customers
- Whether exact email or company-domain matching is allowed
- Whether CRM sync and Accounting time billing will be enabled
- Whether Ticket AI will use local knowledge only or an external API and live web research

Data and privacy planning

Ticket descriptions may contain customer, site or technical information. Decide what may be submitted to an external AI provider before enabling natural-language answers or internet research. Service Desk excludes stored licence-key fields and customer identity fields from prepared AI context and redacts common secrets, but administrators must still set an appropriate internal policy.

Security	Never place production passwords, private API keys or full payment details in tickets or AI resources.

Deployment worksheet

- Production URL, staging URL and responsible site administrator
- WordPress and PHP versions, database prefix and backup location
- Customer portal, resource portal and sign-off page URLs
- Support mailbox, sender identity and mail-delivery provider
- Inbound bridge owner, endpoint secret location and retry policy

- Support roles, named resources and escalation owner
- CRM status-tag owner and Accounting billing approver
- AI data policy, API owner, approved models and monthly cost owner
- Acceptance-test date, go-live date and rollback decision maker

Hosting and caching checks

Authenticated portal pages must display live, user-specific information. Exclude the Support Desk, Support Agent Desk and sign-off pages from full-page caching where the cache system does not already bypass logged-in users. On LiteSpeed and QUIC.cloud sites, purge all layers after changing pages or front-end assets and test with two different accounts.

Confirm that Wordfence or another firewall does not block authenticated admin-post actions or the Service Desk REST routes. PHP upload limits do not affect Free ticket creation, but normal request-size, execution-time and database limits still affect large descriptions, AI resource uploads and imports.

Mail readiness

1. Verify the From email belongs to a domain authorised by the site's mail provider.
2. Send a WordPress test message before testing Service Desk.
3. Confirm SPF, DKIM and DMARC alignment with the chosen sender.
4. Create a controlled mailbox for inbound tests and keep its forwarding rules documented.
5. Decide who will monitor failed outbound messages and rejected inbound webhook requests.

3. Installation and activation

Install the exact Free package and confirm the generated foundation before exposing the portal.

Safe installation

1. Take a current database and files backup.
2. In WordPress, open Plugins > Add New > Upload Plugin.
3. Upload `nomad-support-desk-v0.2.3-role-access-control.zip`.
4. Select Install Now and then Activate.
5. Confirm the Support Desk menu appears and no PHP error is recorded.
6. Open Support Desk > Dashboard and confirm the dashboard loads.

What activation creates

- Support Agent WordPress role and support capabilities
- Support Desk database tables
- Default settings including NSD numbering and ZAR currency
- Draft Support Desk, Support Agent Desk and ticket sign-off pages when equivalents are not found
- A daily billing event
- AI resource storage and embedded AI defaults

Activation validation

Do not publish the generated pages immediately. First confirm access rules, test accounts, menu placement and the correct shortcode on each page.

Check	Expected result
Support Desk menu	Dashboard, Tickets, Add Ticket, Customers, Resources, Analytics, Billing and Settings are available to authorised users
AI menus	AI Resources and AI Settings appear for settings managers
Generated pages	Support Desk pages exist as drafts or existing compatible pages are reused
Role	Support Agent exists with view/manage support capabilities

Database and file backup procedure

1. Take a database backup that can be restored independently of the plugin ZIP.
2. Back up `wp-content/plugins` or retain the exact currently installed Service Desk ZIP.
3. Record the current plugin version and active integration versions.
4. Verify that the backup completed successfully and is stored outside the live account.
5. For an update, test the restore process in staging before changing production.

Generated pages and their publication state

Activation looks for compatible existing pages before creating new ones. It can reuse `support` or `support-desk`, creates the customer and agent pages as drafts when needed, and ensures the sign-off page is published because secure token links must resolve. Review the content and template of every page; do not assume a generated page is correctly placed in menus or protected by the surrounding theme.

Generated page	Default state	Implementation decision
Support Desk	Draft when newly created	Publish only after customer isolation tests
Support Agent Desk	Draft when newly created	Keep out of public menus; link from staff navigation
Support Ticket Sign-off	Published	Verify HTTPS and token access; do not add to main navigation

Post-activation smoke test

1. Open every Support Desk admin submenu as an administrator.
2. Create a temporary support-agent user and verify the role exists.
3. Open the generated pages while logged out and confirm no ticket data appears.
4. Create a temporary customer and resource, then create and assign one ticket.
5. Add a public reply, internal note and time entry; resolve and sign the ticket.
6. Delete only the disposable test ticket after confirming the required records and totals.

4. Roles, permissions and user accounts

Implement least-privilege access for administrators, resources and customers.

Permission model

User type	Typical access	Implementation action
Administrator	All Service Desk screens and AI settings	Retain only for trusted administrators
Support Agent	View and manage support tickets	Use for operational support staff
Resource linked to WP user	Assigned tickets through the front-end portal	Create/link a resource record
Customer WP user	Tickets associated with the user's email	Use a normal logged-in customer account

Create a resource login

1. Create or identify the WordPress user.
2. Open Support Desk > Resources.
3. Add or edit the resource.
4. Select the Linked WordPress user.
5. Enter display name, email, phone, role/title, hourly rate, currency and weekly capacity as required.
6. Keep Active selected and save.
7. Assign a test ticket and confirm the user sees only that assigned ticket in the front-end resource portal.

HR integration

If the suite HR module is active, use Sync active HR employees to resources, then review every created or updated resource. HR linkage can supply the employee relationship, while Service Desk still controls support assignment, rate, capacity and portal access.

Access check A resource record without a correctly linked WordPress user cannot use the assigned-resource portal.

Capabilities and least privilege

Service Desk adds `nsd_view_support` and `nsd_manage_support` to the Support Agent role and administrator. Only administrators receive `nsd_manage_settings` by default. The front-end resource portal also requires an active resource record linked to the current WordPress user; a capability alone does not grant access to every ticket.

Administrators and users who can manage support may use administrative ticket screens. A linked resource is restricted to tickets assigned through the WordPress owner, primary resource or additional-resource relationship. Customer access is determined from the signed-in user's email and the matched Service Desk customer record.

Control	What it permits	What it does not replace
`nsd_view_support`	View support areas and resource portal	A valid resource-to-user link

Control	What it permits	What it does not replace
`nsd_manage_support`	Create and update support work	Administrator-only settings capability
`nsd_manage_settings`	Change Service Desk and AI settings	Good change-control and credential policy
Customer email match	View that customer's portal tickets	A secure WordPress login

Account provisioning procedure

1. Create named WordPress accounts; avoid shared technician logins.
2. Assign the Support Agent role only to operational support staff.
3. Create or sync the resource and link the correct WordPress user.
4. Verify display name, email, phone, title, rate, currency and capacity.
5. Assign one test ticket and test permitted and non-permitted access.
6. Document the offboarding owner and the process for reassigning open tickets.

Free 0.2.3 role model

Free provides one internal role only: Service Desk Agent. The role receives every Service Desk capability, including all-ticket access, customer and resource management, analytics, billing, settings, AI resources/settings, Service Desk user management and ticket deletion. WordPress Administrators inherit the same Service Desk capabilities while retaining normal site administration.

A Service Desk Agent is automatically linked to an active Service Desk resource profile. The account can work in the complete Support Desk administration area and can also use the assigned-resource portal where that workflow is preferred.

Account	Service Desk scope	WordPress scope
WordPress Administrator	All Service Desk functions	Normal unrestricted WordPress administration
Service Desk Agent	All Service Desk functions	Restricted to Support Desk pages and actions
Customer/requester	Own or authorised organisation tickets through the portal	No internal Service Desk role

Create a Free Service Desk user

1. Sign in as a WordPress Administrator or authorised Service Desk Agent.
2. Open Support Desk > Users.
3. Enter a unique username and valid email address.
4. Enter the optional first and last name.
5. Enter a controlled initial password, or leave Password blank to generate a secure password.
6. Leave the login/set-password notification selected when WordPress mail delivery has been tested.
7. Create the user and confirm it appears as Service Desk Agent with an active support resource.
8. Test the account in a private browser: it must land on Support Desk and must not open unrelated WordPress administration pages.

Free access restriction and migration

Non-administrator Service Desk Agents are redirected to the Support Desk dashboard after login. Unrelated top-level WordPress menus are removed, direct requests for unapproved administration pages redirect to Support Desk, and the front-end WordPress admin bar is hidden. Form handlers still enforce capabilities and nonces; menu removal is not the security boundary.

During upgrade, legacy users with direct broad Support Desk capabilities are moved to Service Desk Agent and direct per-user overrides are removed. If Pro is no longer active, existing Pro roles are consolidated into Service Desk Agent. Review every converted account because a former narrow Pro role becomes all-access in Free.

Downgrade review	Deactivating Pro removes its scoped role separation. Confirm that every remaining internal user should receive full Free Service Desk access.

5. Customer and resource portal pages

Publish clear, protected pages for each audience.

Recommended page map

Page	Shortcode	Audience
Support Desk	`[nomad_support_portal]`	Logged-in customers; full portal, ticket form, analytics and history
Open a Ticket	`[nomad_support_ticket_form]`	Logged-in customers; creation form only
Support Analytics	`[nomad_support_customer_analytics]`	Logged-in customers; on-screen analytics only
Support Agent Desk	`[nomad_support_agent_portal]`	Linked active resources; assigned tickets only
Ticket Sign-off	`[nomad_support_ticket_signoff]`	Secure token links and authorised customer access

Publishing procedure

1. Open the generated draft page or create a new page.
2. Insert only the intended shortcode in the content area.
3. Apply the marketplace site's normal full-width page template if used by other portal pages.
4. Exclude the agent page from public navigation.
5. Require login for customer and agent pages through the site's normal account flow.
6. Publish, clear LiteSpeed/QUIC.cloud caches and test in a private browser window.

Portal acceptance

- Logged-out users receive a login message rather than customer data
- Customer A cannot see Customer B's tickets
- Resources see only tickets assigned as primary, assigned user or additional resource
- Forms remain usable at 320px mobile width
- Signature drawing survives screen rotation

Page-by-page configuration

Use one primary shortcode per page. A full customer portal already contains analytics, ticket creation and ticket history, so placing the form and analytics shortcodes on the same page again creates duplicate interfaces. The dedicated shortcodes are intended for sites that want those functions on separate pages.

The resource portal is operational rather than promotional. Apply the same authenticated header and account-navigation pattern used by other staff pages, prevent search indexing where appropriate, and do not expose the URL as proof of access—the plugin still verifies the logged-in user, role and resource assignment.

Customer identity test

1. Create Customer A and Customer B with distinct email domains.
2. Create a WordPress login for each customer using the matching primary email.

3. Create at least one ticket for each customer.
4. Sign in as Customer A and verify only Customer A's records and analytics appear.
5. Repeat as Customer B, then test an unrelated logged-in account.
6. If domain association is enabled, test both an approved secondary address and an unrelated address on a public mailbox domain.

Isolation	Do not launch until cross-customer tests have passed in both normal and private browser sessions.

Resource portal test

1. Assign a ticket to Resource A as primary and Resource B as an additional resource.
2. Verify both can see the ticket and an unrelated Resource C cannot.
3. Remove Resource B and confirm access is removed after refresh.
4. Test search, filters, sort order, pagination, quick status, notes, time and sign-off at desktop and phone widths.

6. Core settings

Configure numbering, email intake, CRM behaviour and billing before creating production tickets.

Email support settings

- Support email address
- From name and From email
- Inbound endpoint and private inbound token
- Allow unknown email customers
- Associate inbound messages by company domain

CRM settings

Enable CRM sync only when the compatible Nomad CRM tables are present. Configure a tag slug for each Service Desk status. When a ticket changes status, the plugin can apply the mapped support tag to the linked CRM contact.

Ticket status	Default tag slug
Open	support-open
Pending	support-pending
Waiting on Customer	support-waiting-on-customer
Resolved	support-resolved
Closed	support-closed

Accounting billing settings

- Enable or disable Accounting time billing
- Choose whether resolved/closed tickets may auto-invoice
- Select the Accounting service product code, normally SUPPORT-TIME
- Set a fallback hourly rate and currency
- Choose the billing increment in minutes

Numbering

Set the ticket prefix and next ticket number before live tickets are created. Do not move the next number backwards after launch because duplicate human-readable ticket numbers may confuse email threading and staff.

Launch control	Save settings and create one disposable test ticket before importing or opening the desk.

Complete settings reference

Setting	Purpose	Recommended first configuration
Support email	Receives new-ticket staff notices and represents the service address	A monitored team mailbox

Setting	Purpose	Recommended first configuration
From name / email	Sender identity for customer replies	A verified domain identity
Inbound token	Authenticates public webhook requests	Long random value stored as a secret
Unknown email customers	Allows a new sender to become a customer	Disable unless the intake policy permits it
Associate by domain	Matches secondary senders to a company	Use only for controlled business domains
CRM sync	Creates or links CRM records and status tags	Enable after mapping tests
Accounting billing	Creates batches and optional invoices	Enable after rate and customer tests
Auto invoice	Invoices eligible unbilled work automatically	Leave off during implementation
Billing increment	Rounds charge calculations	Match the service agreement
Prefix / next number	Controls human-readable ticket numbers	Set once before production

Safe numbering practice

The plugin checks for an existing number and advances until it finds an unused value, but administrators should still treat the next-number field as controlled configuration. Changing the prefix changes future ticket references only; it does not rename old tickets. Preserve the exact ticket number in email subjects because inbound threading searches for that value.

Settings change-control

1. Export or record the current settings before a major change.
2. Make the change in staging or during an approved maintenance window.
3. Clear relevant caches and repeat the workflow affected by the setting.
4. Record the setting, date, reason, tester and result.
5. Roll back immediately if customer isolation, mail, billing or AI privacy tests fail.

7. Customer and resource data setup

Prepare customer matching and assignment records before connecting email intake.

Customer records

1. Open Support Desk > Customers.
2. Add the company name, contact name, primary email and phone.
3. Add each legitimate additional email on its own line.
4. Add company domains only when domain-level association is appropriate.
5. Save and confirm the record appears with expected CRM/Accounting references.

Domain matching caution

Domain matching can associate a new sender with a customer when an exact address is not present. Use it for controlled company domains, not public domains such as gmail.com or outlook.com.

Caution	Do not add shared public email-provider domains to customer records.

Resource records

- Linked WordPress user
- Optional linked HR employee
- Display name and contact details
- Role/title
- Hourly rate and currency
- Capacity hours per week
- Active status and internal notes

Customer field reference

Field	Operational use	Quality rule
Company name	Customer display and integration matching	Use the recognised business name
Contact name	Primary person for the support relationship	Keep it current
Primary email	Portal and exact inbound matching	Must belong to the correct account
Additional emails	Other authorised addresses for the same customer	One controlled address per line
Domains	Optional company-wide inbound matching	Never enter shared public mail domains
Phone	Resource contact action	Include a usable country/area code
CRM / Accounting IDs	Integration references	Do not edit directly in the database

Resource field reference

Field	Operational use	Quality rule
Linked WordPress user	Authenticates the resource portal	One named user per person
HR employee	Connects the central employee record	Review after HR sync
Display name and title	Ticket and portal identification	Use customer-appropriate wording
Hourly rate and currency	Default time-value calculation	Approve commercially before use
Capacity	Planning reference	Keep consistent with working arrangements
Active	Controls new assignment and portal linkage	Deactivate during offboarding

Initial data-loading sequence

1. Create or sync resources before assigning production tickets.
2. Create key customers with verified primary emails.
3. Add only controlled additional addresses and domains.
4. Test CRM and Accounting links with disposable records.
5. Create a ticket for each main customer type and complete the lifecycle.
6. Review duplicates before importing any legacy ticket history.

8. Inbound email implementation

Connect a mailbox bridge without exposing the inbound token.

Endpoint contract

Field	Requirement	Purpose
token	Required	Authenticates the inbound bridge
from or from_email	Required	Finds or creates the customer
from_name	Optional	Sender display name
to	Optional	Records destination context
subject	Required	Creates a subject or matches `[NSD-1001]`
text/body/message	Required	Message content
html	Optional fallback	Used when plain text is missing
message_id	Recommended	Duplicate protection

Implementation sequence

1. Create the mailbox or forwarding rule.
2. Choose a mailbox parser, Cloudflare Email Worker, Zapier/Make flow or controlled custom bridge.
3. Copy the endpoint and token from Support Desk > Settings.
4. Store the token as a secret in the bridge.
5. Map the required fields and include message_id.
6. Send a new-ticket test from a known customer.
7. Reply with the ticket number in the subject and confirm the reply attaches to the same ticket.
8. Repeat from an unknown address and confirm behaviour matches the configured policy.

Operational controls

- Rotate the inbound token if it is exposed
- Retain bridge delivery logs without retaining unnecessary message content
- Rate-limit the public bridge
- Reject oversized requests before forwarding to WordPress
- Monitor duplicate and failed deliveries

Bridge behaviour and response handling

The endpoint accepts the token as a posted field or `X-NSD-Token` header. It requires a valid sender, subject and text body; HTML is stripped only as a fallback. If `message_id` has already been logged, the endpoint returns a successful duplicate response without creating another reply or ticket. A subject containing the existing ticket number adds a reply; otherwise a new ticket is created.

A 403 response normally means the secret is wrong. A 400 response means required fields are absent. A 200 response can mean a ticket or reply was created, or that a duplicate was safely ignored. Configure the bridge to log the HTTP status and response body without permanently storing unnecessary message content.

Response condition	Meaning	Bridge action
403 invalid token	Authentication failed	Stop retries and correct/rotate the secret
400 missing fields	Sender, subject or body is absent	Correct mapping before retrying
200 ticket_created	New request accepted	Store returned ticket ID for audit
200 reply_added	Existing ticket matched	Preserve the subject on future replies
200 duplicate	Message ID already processed	Treat as delivered; do not resend

Inbound security controls

- Use HTTPS and keep the token outside browser-side code
- Limit the bridge to the endpoint and expected method
- Reject oversized messages before sending them to WordPress
- Sanitise and normalise sender fields in the bridge
- Rate-limit public intake and alert on repeated authentication failures
- Rotate the token immediately after suspected exposure
- Retain message IDs and delivery outcomes, not unnecessary full mailbox content

End-to-end email acceptance test

1. Send a new plain-text message from a known customer and verify customer matching.
2. Reply to the support response without removing `[NSD-####]` and verify one reply is added.
3. Resend the same `message_id` and confirm no duplicate appears.
4. Test an unknown sender with the allow/deny setting in both states.
5. Test a secondary address and a domain match independently.
6. Confirm internal notes are never emitted as customer email replies.

9. Embedded Ticket AI implementation

Configure knowledge-assisted analysis independently inside Service Desk.

AI modes

Mode	Data source	API requirement
Provided resource files	Indexed Service Desk AI resources	No key required for direct resource matching; generated synthesis requires AI enabled
Internet only	Approved live web research	AI answers, API key and web search enabled
Resource files + internet	Both sources	AI answers, API key and web search enabled

Configure AI Settings

1. Open Support Desk > AI Settings and enable Support Desk AI features.
2. Choose whether the API should generate natural-language answers.
3. Confirm the API base URL and enter the API key; leave the key field blank on later saves to retain it.
4. Set the supported model, temperature and maximum matched resources.
5. Choose whether to show Ask Support AI before the customer form and enable live internet research only after privacy approval.
6. Select Save and Test API and confirm the connection result.

Build the resource library

1. Open Support Desk > AI Resources > Add New.
2. Choose manual content, approved URL or supported file upload.
3. Use a clear title and status, then upload or paste only reviewed material.
4. Save and confirm the resource is indexed; use Re-index after replacing source material.
5. Run a known-answer ticket test and inspect returned sources.

Supported knowledge content

- Manual troubleshooting notes
- Approved public URLs
- PDF manuals
- Text and code files
- Plugin ZIP packages

Verification	AI output is assistance, not an automatic repair. Staff must verify commands, versions, citations and risk before using a suggestion.

AI settings field reference

Setting	Meaning	Implementation guidance
Enable Support Desk AI	Turns the embedded assistant panels on	Enable after access and privacy approval
Generate natural-language answers	Uses the configured API for synthesis	Requires a valid API key
API base URL	Provider endpoint	Use the approved HTTPS endpoint
API key	Provider credential	Enter once; later blank saves retain it
Model	Model sent to the API	Use a model supported by the account and endpoint
Temperature	Variation in generated output	Keep low for support analysis
Maximum matched resources	Limits retrieved local context to 1-12 items	Start with 6-8 and test relevance
Show before ticket form	Adds Ask Support AI to the customer workflow	Enable only after customer guidance is ready
Live internet research	Permits current public-web searching	Approve privacy and cost before enabling

Resource ingestion and maintenance

AI Resources can contain manual notes, a website URL, or an uploaded file/ZIP. Use descriptive titles and set obsolete material inactive rather than leaving contradictory instructions searchable. URL resources depend on the current retrieved content. Supported text-oriented files and code inside ZIP packages are extracted for indexing; image resources retain metadata and need written notes to explain the support knowledge they represent.

1. Classify the source as approved, internal or public before adding it.
2. Remove credentials, customer data and obsolete instructions.
3. Add a title that includes product and version where relevant.
4. Save and confirm the indexed word count or extracted content is plausible.
5. Ask a known question and inspect both the answer and cited source.
6. After replacing material, re-index and repeat the known-answer test.

AI access, rate limits and audit

Pre-ticket queries require a logged-in user. Ticket analysis requires an administrator, support manager or assigned active resource with access to that specific ticket. Requests are rate-limited, and completed ticket analyses are recorded as Service Desk events. Generated text is inserted only when the user selects an action; it remains editable before the reply or work note is saved.

Human verification	Citations show where information came from; they do not prove that an instruction is safe for the customer's exact system. Verify versions, backups, commands and likely impact.

10. CRM and Accounting integration

Connect optional suite modules defensively and test both directions.

CRM integration test

1. Enable CRM sync.
2. Create or update a Service Desk customer.
3. Confirm the related CRM contact/company mapping is populated when supported.
4. Create a ticket and change its status.
5. Confirm the mapped CRM support-status tag is applied.
6. Disable the test tag mapping if it conflicts with an existing CRM taxonomy.

Accounting time billing test

1. Confirm Accounting Pro is active and the configured service product exists.
2. Set the service product code or fallback hourly rate.
3. Log billable time on a test ticket.
4. Open Support Desk > Billing and generate a batch for the customer and period.
5. Review minutes, amount and currency.
6. Create the Accounting invoice and confirm the invoice/customer/item linkage.

Auto-invoice caution

Leave automatic invoice on close disabled until manual billing batches have passed acceptance testing. A resolved or closed ticket can trigger invoicing only when unbilled billable time exists and the Accounting integration is ready.

Financial control	Use manual review during initial rollout so rates, rounding increments and customer mappings can be confirmed.

CRM record matching and tags

The integration checks for compatible CRM tables and columns before writing. A Service Desk customer can link to an existing CRM contact by email or create the appropriate contact record when supported. Company information may also be synchronised. Each status can map to a CRM tag slug, allowing CRM lists and automation to reflect the support state.

Use distinctive support tag slugs and test them in a non-production contact first. If the CRM already uses similar tags for another workflow, agree ownership before enabling Service Desk mapping. A status change should add the current support tag without corrupting unrelated CRM classification.

Accounting billing flow

A time entry stores minutes, rate, amount, currency, billable choice, user/resource and note. The configured billing increment influences the calculated amount. Billing groups eligible unbilled entries into a draft customer batch; creating an Accounting invoice links the batch and entries to the resulting invoice so they are not selected again.

Stage	Control	Evidence
Time capture	Correct customer, resource, minutes, rate and billable flag	Ticket time table
Batch creation	Customer and optional date period	Batch number and line items
Review	Minutes, amount, currency and scope	Draft batch totals
Invoice creation	Compatible Accounting customer and product	Accounting invoice ID on batch and entries

Integration failure policy

- Keep Service Desk usable when CRM or Accounting is unavailable
- Do not retry invoice creation blindly after an uncertain response
- Check whether a customer or invoice was created before repeating an action
- Correct mapping in the application screens rather than editing IDs directly
- Record the affected customer, batch, ticket and timestamp for escalation

11. Customer sign-off implementation

Configure secure customer or job-card approval for completed work.

Sign-off flow

1. Ensure the sign-off page contains `[nomad_support_ticket_signoff]`.
2. Open a test ticket as authorised staff or resource.
3. Generate or use the secure sign-off link.
4. Open the link in a separate browser context.
5. Enter signer name/email, draw the signature and confirm consent.
6. Submit and verify the signature record appears on the ticket.
7. Confirm the internal timeline note and optional ticket closure behaviour.

Evidence retained

- Signer name and email
- Signature image data and integrity hash
- Consent text
- Source, date and time
- IP address and browser user-agent
- Internal timeline event

Implementation policy

Define what sign-off means for the business, who may sign, whether closure is automatic and how long the evidence must be retained. The plugin captures evidence but does not replace the organisation's legal or record-retention policy.

Sign-off access and evidence

A sign-off link contains the ticket ID and an HMAC token derived from site secrets and ticket data. The page can also be used by the authorised logged-in customer, assigned resource or support manager. The record stores the signer name and email, signature image, integrity hash, consent text, source, time, IP address and browser user-agent. Submitting again creates another record; it does not overwrite the earlier signature.

Business process design

- Define which ticket states make sign-off appropriate
- Decide whether the default Mark ticket as closed option should be used operationally
- Write an approved consent statement for the business context
- Tell technicians what the customer must review before signing
- Define how corrections, disputed work and repeat signatures are handled
- Set retention, export and access rules for signature evidence

Sign-off acceptance test

1. Open the secure link while logged out and confirm it resolves only with the correct token.

2. Alter the token and confirm access is denied.
3. Draw with a mouse and then on a phone or tablet.
4. Rotate the mobile screen and verify the drawing is preserved.
5. Submit with and without closure selected and verify the resulting status.
6. Confirm the internal timeline entry and stored signature are attached to the correct ticket.

12. Testing and acceptance

Use role-based tests before publishing production links.

Minimum functional test matrix

Area	Test	Pass condition
Customer isolation	Two customers view portals	Each sees only own tickets
Resource isolation	Two resources view assigned desk	Each sees only assigned work
Lifecycle	Open to Pending to Waiting to Resolved to Closed	Statuses and timestamps are correct
Time	Billable and non-billable entries	Totals and amounts recalculate
Email-in	New email and threaded reply	Correct customer and ticket
AI	Knowledge and permitted web modes	Sources shown; no hidden identity fields
Sign-off	Token link and logged-in flow	Signature stored on correct ticket
Mobile	320px, tablet and desktop	No horizontal workflow loss

Performance and cache checks

- Exclude authenticated support portal pages from full-page caching
- Purge LiteSpeed and QUIC.cloud after page or CSS changes
- Confirm REST requests are not blocked by Wordfence or another firewall
- Check Site Health and PHP logs after representative portal use

Acceptance record

Record the tested WordPress/PHP versions, test accounts, test cases, results, open defects, responsible owner and go-live approval. Keep this record outside the plugin so it survives updates.

Role-based acceptance accounts

Create separate test accounts for administrator, support manager, Resource A, Resource B, Customer A and Customer B. Do not perform the entire test as an administrator because administrator access can hide permission defects. Keep a test matrix that records the account, action, expected result, actual result, tester and evidence.

Detailed acceptance scenarios

Scenario	Steps	Pass condition
New portal ticket	Customer creates and staff opens it	Correct customer, number, priority and staff notice
Public and internal reply	Staff sends each type	Customer sees only the public reply
Resource isolation	Assign/remove additional resources	Access follows current assignment
Inbound threading	New mail, reply, duplicate retry	One ticket, correct reply, no duplicate
Time and batch	Billable and non-billable entries	Only eligible work enters the batch

Scenario	Steps	Pass condition
AI knowledge	Known-answer question	Relevant sources, editable output and audit event
Sign-off	Valid and invalid token	Correct access and evidence
CRM/Accounting	Status and invoice tests	Correct linked records without duplicates

Non-functional acceptance

- No PHP warnings or fatal errors in the tested workflows
- No cross-customer or unassigned-resource data exposure
- Usable controls at 320px, tablet and desktop widths
- Keyboard-visible focus and labelled controls
- Reasonable response time with representative ticket volume
- Working outbound mail and documented failed-mail monitoring
- Current backup and tested rollback procedure

Role and user acceptance tests

Test	Expected result
Agent login	Redirects to Support Desk dashboard
Agent menu	Only Support Desk top-level administration is visible
Direct URL	Unrelated wp-admin request redirects to Support Desk
User creation	New account is Service Desk Agent and active resource
Agent ticket access	All tickets and all Service Desk functions are available
Customer account	Only authorised portal tickets and public content
WordPress Administrator	Full WordPress and Service Desk access remains

13. Go-live, maintenance and rollback

Launch in a controlled sequence and maintain a recoverable service.

Go-live checklist

- Backups verified
- Portal URLs published and correctly linked
- Agent portal kept out of public menus
- Support email and outbound delivery tested
- Inbound token protected
- Customer and resource access tested
- AI privacy and cost settings approved
- CRM and billing settings confirmed
- First-line staff trained
- Rollback owner identified

Routine maintenance

- Apply WordPress and plugin updates in staging first
- Review inactive resources and customer domains
- Rotate exposed inbound or API credentials
- Re-index changed AI resources
- Review failed mail and bridge events
- Audit unbilled time and old billing batches
- Export or back up before major configuration changes

Rollback

1. Pause inbound forwarding and customer announcements.
2. Take a diagnostic backup of the affected state.
3. Deactivate the changed build without deleting data.
4. Restore the last verified plugin files or site backup.
5. Save permalinks and clear caches.
6. Re-run the minimum acceptance tests before reopening the service.

Data protection	Do not delete Service Desk tables as part of a normal rollback.

Controlled launch sequence

1. Freeze configuration and approve the final acceptance record.
2. Take a verified database and files backup.
3. Publish the customer and resource pages in their intended navigation contexts.

4. Enable inbound forwarding and verify the first message.
5. Invite a small pilot group before opening access broadly.
6. Monitor the first tickets, mail events, time entries and AI requests in real time.
7. Enable optional CRM, billing automation and web research one at a time after the core desk is stable.

Maintenance calendar

Frequency	Checks
Daily	Failed inbound/outbound mail, urgent queue, unassigned tickets and unbilled anomalies
Weekly	Backups, inactive resources, stale Waiting tickets, AI failures and integration health
Monthly	Customer isolation test, resource access test, token exposure review, AI resource freshness and billing reconciliation
Before updates	Staging update, release notes, backup, role-based smoke test
After updates	Permalinks/caches, portals, inbound mail, AI, sign-off and billing

Rollback decision and procedure

Rollback is appropriate when the new build causes a reproducible security, access, data-integrity or critical workflow failure that cannot be corrected safely within the maintenance window. Deactivate or replace plugin files only after capturing diagnostic evidence. Do not delete Service Desk tables as part of a normal rollback; the data must remain available to the restored version.

14. Troubleshooting reference

Resolve common deployment faults methodically.

Common issues

Symptom	Likely check	Action
Portal says sign in	Current user/session	Use the intended account and clear stale login cookies
Resource sees no tickets	Resource linkage and assignment	Link the WordPress user, activate resource and assign ticket
Inbound email rejected	Token and field mapping	Compare bridge payload with Settings endpoint contract
Reply creates a new ticket	Ticket number in subject	Preserve `[NSD-####]` in replies
AI disabled message	AI Settings	Enable AI and required mode; test API when applicable
REST request blocked	Firewall/cache	Allow the Service Desk REST route for authenticated use
Billing amount unexpected	Rate, increment and billable flag	Correct settings and re-test with a disposable entry
Mobile save difficult	Old cached CSS/JS	Purge all cache layers and retest

Escalation evidence

- Exact page and role
- Ticket number without private secrets
- WordPress and PHP versions
- Service Desk version
- Browser/device
- Relevant Site Health or PHP log excerpt
- Steps to reproduce
- Whether the issue occurs with caching/security plugins temporarily isolated in staging

Diagnostic method

1. Reproduce the issue with the exact affected role in a private browser session.
2. Record the page, ticket, action, time and full on-screen message.
3. Check Site Health, PHP logs, browser console/network and mail or bridge logs as applicable.
4. Confirm WordPress, PHP, theme and plugin versions.
5. Repeat in staging with caching/security interference isolated one control at a time.
6. Compare the result with a second account or known-good ticket.
7. Apply the smallest reversible correction and repeat the acceptance test.

Failure-to-owner map

Failure area	First owner	Evidence to collect
Login or portal access	WordPress/site administrator	User, role, email match, resource link and page URL
Inbound mail	Mailbox bridge owner	Message ID, HTTP status, mapped fields and timestamp
Outbound mail	Mail-delivery owner	Recipient, sender, mail log and provider result
CRM sync	CRM administrator	Customer/contact IDs, status and expected tag
Billing/invoice	Accounting administrator	Ticket, time entry, batch and invoice IDs
Ticket AI	AI/settings administrator	Mode, model, source, error and audit time
Mobile layout	Website administrator	Device, viewport, page, cache state and screenshot

Evidence privacy

Remove or mask passwords, tokens, full licence keys, customer personal data and confidential file content before sharing diagnostic evidence. Ticket numbers, timestamps and non-sensitive record IDs usually provide enough correlation without exposing the underlying content.

15. Implementation handover and technical reference

Hand the configured service to its operational owners with a complete, maintainable record.

Handover pack

- Installed Free version and source ZIP checksum
- Production, staging and portal URLs
- Named business owner, technical owner, queue manager and billing approver
- Role and resource assignment list
- Settings export or controlled configuration record
- Inbound bridge diagram, secret owner and rotation procedure
- Mail sender configuration and failed-delivery monitoring
- CRM tag map and Accounting product/rate map
- AI policy, model, API owner and approved resources
- Acceptance record, known limitations, open defects and rollback steps

Database table reference

Table suffix	Purpose
customers	Support customer identity and integration references
tickets	Ticket lifecycle, assignment, totals and metadata
replies	Public replies, internal notes and status movements
time_entries	Minutes, rates, values and billing links
resources / ticket_resources	People and multi-resource assignments
signatures	Customer/job-card sign-off evidence
email_logs	Inbound and outbound message audit
events	Operational and AI audit events
billing_batches / billing_items	Grouped unbilled work and invoice preparation
ai_resources	Indexed Service Desk knowledge sources

Shortcode and route reference

Interface	Reference	Access
Full customer portal	`[nomad_support_portal]`	Logged-in customer matched by email/domain
Ticket form only	`[nomad_support_ticket_form]`	Logged-in customer
Customer analytics	`[nomad_support_customer_analytics]`	Logged-in customer

Interface	Reference	Access
Resource portal	`[nomad_support_agent_portal]`	Support capability plus linked active resource
Sign-off	`[nomad_support_ticket_signoff]`	Valid token or authorised logged-in user
Inbound email	`/wp-json/nomad-support-desk/v1/inbound-email`	Private token
Pre-ticket AI	`/wp-json/nomad-support-desk/v1/support-ai/query`	Logged-in user and enabled AI
Ticket AI	`/wp-json/nomad-support-desk/v1/ticket-ai/{id}`	Administrator/manager or assigned resource

Final ownership checklist

1. Obtain written operational acceptance from the service owner.
2. Confirm administrators know where settings and AI resources are managed.
3. Confirm queue staff understand status, reply visibility and time rules.
4. Confirm customers and resources have the correct portal links.
5. Schedule the first weekly and monthly reviews.
6. Store the handover pack and backup locations where authorised owners can retrieve them.