

Installation & Lifecycle

Managed documentation for Installation & Lifecycle.

- [Installation Procedure](#)
- [Upgrade and Rollback](#)

Installation Procedure

Installation Procedure

Field	Value
Distribution	BSS — OpenC5ISR
Product	OpenC5ISR
Release	0.15.1
Deployment	<code>openc5isr.bss.dev</code>
Source	<code>/srv/bss/releases/OpenC5ISR-v0.15.0-20app</code>
Evidence	Release manifests plus controlled deployment pattern
Source fingerprint	<code>93926e8c834f1ed79c9017a4096f70c76d73ce4f870f104fbd0cf54b2bed50ee</code>
Status	Generated baseline — human review required

“ **Verification boundary:** This page combines platform design guidance with static evidence from the release. It does not prove that every detected interface is enabled, reachable, secure, or operational in the deployed environment.

Candidate build/install entry points

Evidence	Invocation	Declared action
<code>package.json</code>	<code>npm run start</code>	<code>npm run verify:startup && node server.js</code>

Evidence	Invocation	Declared action
package.json	npm run test	python3 tests/integrated_smoke_test.py && python3 tests/openknowledge_opentask_ontology_smoke_test.py && python3 tests/opendaa_opendata_smoke_test.py && python3 tests/opendaa_openanalytics_smoke_test.py && python3 tests/opencop_smoke_test.py && python3 tests/smoke_test.py && python3 tests/openeyes_smoke_test.py && python3 tests/raspbots_agent_smoke_test.py && python3 tests/openpnt_smoke_test.py && python3 tests/openpnt_linuxptp_wrapper_smoke_test.py && python3 tests/openpnt_agent_smoke_test.py && python3 tests/external_integrations_smoke_test.py && python3 tests/compiled_deployment_smoke_test.py && node tests/openssa/test_openssa.js && python3 tests/openssa/test_sensor.py && python3 tests/openssa_opensof_smoke_test.py && python3 tests/openrf_opensof_smoke_test.py && python3 tests/openlvc_edge_gateway_test.py && python3 tests/openlvc_smoke_test.py && python3 tests/opencybersec_smoke_test.py && python3 tests/opencybersec_agent_smoke_test.py && python3 tests/openiia_smoke_test.py && python3 tests/openaar_mr_smoke_test.py
package.json	npm run verify:startup	node -e "require('./lib/openpnt-timing'); require('./lib/openpnt'); console.log('startup dependencies OK')"
edge/openssa-sensor/package.json	npm run start	node local_server.js 3000
edge/openssa-sensor/package.json	npm run check	node --check lib/openssa.js && node --check public/open-ssa/app.js && node --check local_server.js && python3 -m py_compile drivers/ssa_sensor.py tools/*.py demo/run_streaming_demo.py tests/*.py
edge/openssa-sensor/package.json	npm run test	npm run check && node tests/test_openssa.js && python3 -m unittest discover -s tests -v
third_party/ontowiki/Makefile	make clean	Make target
third_party/ontowiki/Makefile	make codebeautifier	Make target
third_party/ontowiki/Makefile	make codesniffer	Make target

Evidence	Invocation	Declared action
third_party/ontowiki/Makefile	make composer-install	Make target
third_party/ontowiki/Makefile	make debianize	Make target
third_party/ontowiki/Makefile	make default	Make target
third_party/ontowiki/Makefile	make deploy	Make target
third_party/ontowiki/Makefile	make devenv	Make target
third_party/ontowiki/Makefile	make directories	Make target
third_party/ontowiki/Makefile	make getcomposer	Make target
third_party/ontowiki/Makefile	make help	Make target
third_party/ontowiki/Makefile	make help-cs	Make target
third_party/ontowiki/Makefile	make help-test	Make target
third_party/ontowiki/Makefile	make install	Make target
third_party/ontowiki/Makefile	make list-events	Make target
third_party/ontowiki/Makefile	make odbctest	Make target
third_party/ontowiki/Makefile	make test	Make target
third_party/ontowiki/Makefile	make test-directories	Make target
third_party/ontowiki/Makefile	make test-extensions	Make target
third_party/ontowiki/Makefile	make test-integration-mysql	Make target
third_party/ontowiki/Makefile	make test-integration-virtuoso	Make target
third_party/ontowiki/Makefile	make test-unit	Make target
third_party/ontowiki/debian/Makefile/Makefile	make clean	Make target
third_party/ontowiki/debian/Makefile/Makefile	make default	Make target
third_party/ontowiki/debian/Makefile/Makefile	make package	Make target
third_party/ontowiki/debian/Makefile/Makefile	make prepare	Make target

Controlled installation procedure

1. Record the release artifact name, version, checksum, source and approval.
2. Back up configuration, persistent data, database state and the active proxy/service definitions.

3. Extract the release into a new immutable version directory; do not overwrite the active release.
4. Install dependencies using the release manifests and pinned lock files. Review install scripts before executing them as a privileged user.
5. Create/update edition-specific configuration and secrets outside the release directory.
6. Run unit/static tests and a local health check before switching production traffic.
7. Activate atomically through the current symlink/container tag/service definition, then restart only affected services.
8. Validate process state, local endpoints, Nginx routing, WebSockets, login, primary workflow and logs.
9. Record actual results and keep the previous release until the acceptance window completes.

Do not assume

The generator deliberately does not invent an install command when the release does not provide enough evidence. Promote a command into this page only after it has been executed successfully on a clean or controlled target.

Maintainer Notes

Add human-reviewed deployment notes, corrections, decisions, screenshots, and links here. Content outside the generated block is preserved on future runs.

Upgrade and Rollback

Upgrade and Rollback

Field	Value
Distribution	BSS — OpenC5ISR
Product	OpenC5ISR
Release	0.15.1
Deployment	<code>openc5isr.bss.dev</code>
Source	<code>/srv/bss/releases/OpenC5ISR-v0.15.0-20app</code>
Evidence	Lifecycle pattern and optional release comparison
Source fingerprint	<code>93926e8c834f1ed79c9017a4096f70c76d73ce4f870f104fbd0cf54b2bed50ee</code>
Status	Generated baseline — human review required

“ **Verification boundary:** This page combines platform design guidance with static evidence from the release. It does not prove that every detected interface is enabled, reachable, secure, or operational in the deployed environment.

Pre-upgrade gate

- Previous release path and rollback command are known and tested.
- Database/configuration migrations are identified, backed up and reversible or explicitly irreversible.
- External interfaces and dependent modules have compatibility expectations recorded.
- A maintenance/communications plan exists for user-visible interruption.
- Acceptance tests and responsible reviewer are assigned.

Upgrade sequence

```
backup → stage new release → install/validate dependencies → migrate if required  
→ switch active release → restart affected services → smoke/acceptance tests  
→ observe → approve or roll back
```

Rollback sequence

```
stop traffic/affected service → restore compatible data/config if required  
→ repoint active release → restart → validate local/public endpoints  
→ record incident and preserve failed-release evidence
```

Release-specific comparison

No previous release was supplied. Run with `--compare /path/to/previous-release` to generate a static change inventory.

Maintainer Notes

Add human-reviewed deployment notes, corrections, decisions, screenshots, and links here. Content outside the generated block is preserved on future runs.