



# ARIANNA

## DOCUMENT MANAGEMENT SYSTEM

**Trust in your documents.**  
**Control at every stage of the project.**

Arianna is a complete and intelligent solution for managing technical project documentation, designed for those who understand the daily challenges of designers, document controllers and project managers.



### ORGANIZE

An intuitive hierarchical structure that mirrors the reality of the project.



### REVISION

Complete revision management with total traceability.



### APPROVE

Configurable workflows for an efficient and transparent approval process.



### CONTROL

Full audits to always know who did what and when.



### FIND

Instant search by metadata and content, with precise results in seconds.



### SYNCHRONIZE

Work online or offline. Data always aligned.



### PROTECT

Security, permissions and roles to protect every document and information.



### REDUCE ERRORS

Accurate and up-to-date information always available.



### SAVE TIME

Automated processes, faster searches, less low-value activity.



### IMPROVE COLLABORATION

Everyone works on the same documents, in the right version, at the right time.



### INCREASE PRODUCTIVITY

More control, greater efficiency, better project results.

THE RIGHT DOCUMENT. AT THE RIGHT TIME. IN THE RIGHT HANDS.

# ARIANNA 2.0

---

DOCUMENT MANAGEMENT SYSTEM

## Complete Manual

Philosophy, concepts, operator's guide, and complete feature reference

*Arianna provides a fully localized user interface in both Italian and English. Users can switch the application language at any time through the application settings. Unless otherwise stated, the screenshots included in this manual show the Italian interface..*

*Client names, project names, and document contents shown in this manual are fictitious and generated for illustration purposes only.*

# CONTENTS

---

## PART I — PHILOSOPHY AND VISION

---

- 00 Manifesto
  - Why Arianna exists
  - The cost of document chaos
  - The philosophy of Arianna
- 01 What Arianna is

## PART II — CONCEPTUAL MODEL

---

- 02 The data hierarchy
- 03 Why check-out locks the file
- 04 Why revisions are never overwritten
- 05 Why a document has a workflow status
- 06 Relationships between document types

## PART III — OPERATOR'S GUIDE

---

- 07 Login
- 08 Creating a client
- 09 Creating a project
- 10 Importing a file and metadata
- 11 Document preview
- 12 Check-out
- 13 Check-in — same revision
- 14 New revision
- 15 Workflow status
- 16 Tags
- 17 Links between documents
- 18 Renaming a file
- 19 Removing a file
- 20 Plants and units (skids)
- 21 Project templates

## **PART IV — SEARCH AND NAVIGATION**

---

- 22 Search everywhere
- 23 Global search
- 24 Recent and favorite documents

## **PART V — TRACEABILITY AND CONTROL**

---

- 25 KPI dashboard
- 26 Audit trail
- 27 Project timeline
- 28 Project evolution report
- 29 Document history and Where used
- 30 Impact analysis by type
- 31 Document dependency map
- 32 Operational notifications

## **PART VI — ADMINISTRATION**

---

- 33 User management
- 34 Roles and permissions
- 35 Server and storage configuration
- 36 Connection profile and multi-site sync
- 37 Arianna 2.0 modules
- 38 Personal account

## **PART VII — CASE STUDY**

---

- 39 From an empty project to the first issued document

## **APPENDICES**

---

- A Icon legend
- B Glossary
- C Standard document dependency map
- D Frequently asked questions
- E Application messages and confirmations

PART I

# Philosophy and vision

Why Arianna exists, and what problem it actually solves

## 00 Manifesto

---

Every great industrial project begins with an idea.

*But it takes shape through thousands of documents.*

### Foreword

Arianna was born out of hands-on experience managing documentation for large industrial projects. It is not a simple document archive but a platform designed to organize, connect, and protect a project's information assets.

### Why Arianna exists

Every project generates thousands of documents. The problem isn't storing them — it's always knowing which revision is the correct one and which document should actually be used. A medium-sized plant engineering project juggles dozens of document types — P&IDs, process narratives, I/O lists, electrical diagrams, certificates — each with its own revision history, each subject to its own approval cycle. Without a system that imposes structure, that history scatters across network folders, emails, and local copies on whoever last touched the file.

### The cost of document chaos

Wasted time, errors, rework, and duplication carry a real cost. An operator working from a superseded revision of a P&ID can introduce an error that only surfaces on site, where fixing it costs orders of magnitude more than catching it at the drawing board. Arianna reduces these risks through traceability and control: every file has a certain provenance, every change has an author and a reason, every status is visible at a glance.

### The philosophy of Arianna

Documentation is a network of connected information, not a pile of independent files. Arianna manages relationships, workflow, revisions, and responsibility as part of the same model: a P&ID is never just a drawing — it's a node that feeds I/O lists, electrical diagrams, cause-and-effect charts — and knowing what depends on what is a precondition for managing change without surprises.

#### PRINCIPLE

Arianna never modifies original client documents: every imported file stays intact in the managed archive; changes always generate a new working copy or a new revision, never a silent overwrite.

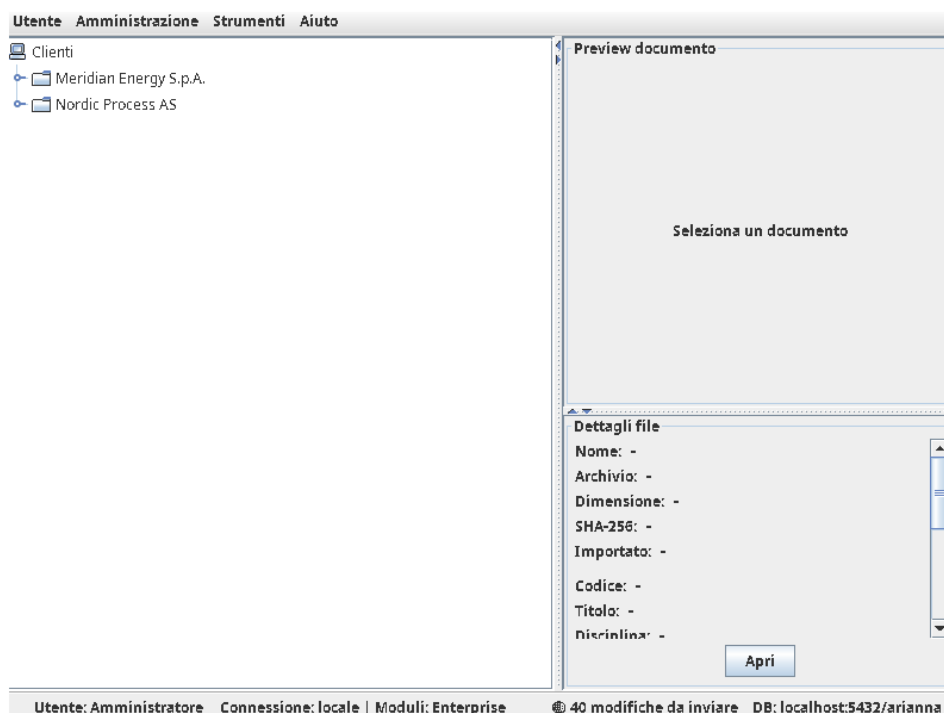
## 01 What Arianna is

Arianna 2.0 — a desktop Document Management System for projects and plant technical documentation: revisions, workflow, and traceability in a single shared repository.

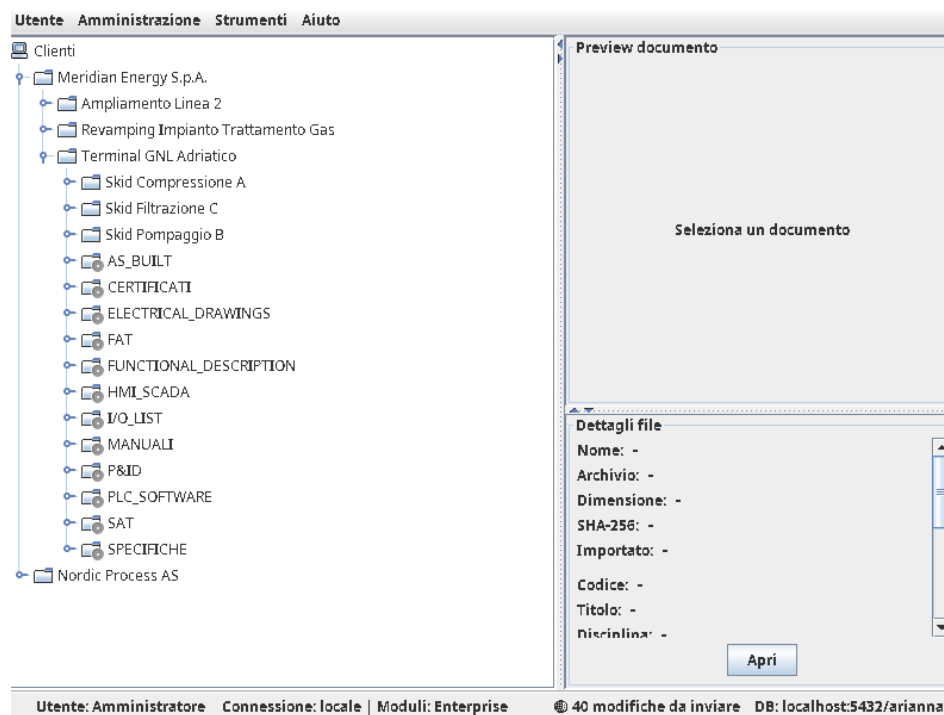
Arianna is a multi-user desktop DMS built for engineering firms and companies that manage projects with complex engineering documentation: drawings, P&IDs, process narratives, I/O lists, certificates. Each client has a single, shared repository, with full traceability of who did what and when.

Physical files stay outside the database, in managed, secure storage: the user always sees only the document's original name, never the internal technical path. Check-out generates a local working copy without ever touching the official file until a verified check-in is released.

- **Controlled revisions and workflow.** Each document type keeps its own revision numbering, with a full history and an advancement status (draft → internal/client review → approved → issued → archived).
- **Adaptive hierarchical structure.** Client → Project → (optional) Plant/Unit → Document type → Revision → File: the same scaffolding supports both a simple project and a multi-skid plant.
- **Multi-site synchronization.** A sync queue reliably propagates creations, renames, and deletions across connected sites.



**Fig. 1** Client tree



**Fig. 2** A project with multiple units/skids

PART II

# Conceptual model

How Arianna organizes information, and why

## 02 The data hierarchy

---

Before going through the procedures, it helps to understand how Arianna organizes information and why some operations — check-out, revisions, workflow status — exist in the form you find them. They aren't bureaucratic overhead: they mirror the way a plant engineering project actually produces, verifies, and approves its own technical documentation.

|                      |                                                                                                                                                                                         |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Client</b>        | The company you're working for — an Oil & Gas operator, an industrial client. It's the root of the tree.                                                                                |
| <b>Project</b>       | A specific undertaking for that client. It has its own document structure, optionally organized by plant/unit (skid) when the project comprises several distinct systems.               |
| <b>Plant / Unit</b>  | Optional level below a project. Useful when a plant consists of several independent skids (e.g. Compression Skid A, Filtration Skid C), each with its own documentation and revisions.  |
| <b>Document type</b> | A category of technical document defined by the project — P&ID, FUNCTIONAL_DESCRIPTION, SAT, etc. It's created with the project (or with a plant/unit) and always starts at Revision 0. |
| <b>Revision</b>      | The evolving state of a document type over time. It advances only when a change is substantial; earlier revisions remain available in the history, never overwritten.                   |
| <b>File</b>          | The actual document (PDF, DOCX, XLSX...) attached to a revision, with its own metadata, SHA-256 hash, check-out status, and workflow status.                                            |

**NOTE** A document type can hold more than one file when the project is organized by plant/unit: in that case Arianna automatically creates the container for each unit listed, keeping the same conceptual scaffolding whether the project is simple or a multi-skid plant.

### **03 Why check-out locks the file for others**

---

On a plant engineering project, several people work on the same documents. Without a lock, two people could edit the same revision in parallel and one would overwrite the other's work without knowing it. Check-out makes it explicit that "I'm working on this" until check-in arrives: no other operator can upload a new version of the same file in the meantime, though they can still view the latest released version read-only.

## **04 Why revisions are never overwritten**

---

Technical documents change because of comments, internal checks, client requests. Every substantial change must remain traceable: who made it, for what reason, who checked it, and who approved it (the Prepared by / Checked by / Approved by fields at check-in). Losing that history would mean losing the traceability plant engineering requires — and in the event of a dispute or non-conformance, being able to reconstruct with certainty which revision was in force at a given moment is not optional.

## **05** Why a document has a workflow status

---

A technical document is never simply "done": it moves through an approval chain — Draft, Internal review, Client review, Approved, Issued, Archived — that mirrors the real process by which a project validates its own documentation before it is used on site or in the workshop. The colored badge on the tree makes that status visible without opening the file.

## 06 Relationships between document types

---

A plant's documentation isn't a set of independent files: a P&ID feeds instrument lists, line lists, valve lists; a change to the process narrative can impact the PLC software. Arianna models these relationships on two levels.

### FEEDS

The upstream document feeds the downstream one in the project's natural lifecycle (e.g. PFD → P&ID).

### IMPACTS

A change to the upstream document may require checking or updating the downstream one (e.g. P&ID → Instrument List).

This map is defined once, at the document-type level — not per file — and is reused automatically in every project. It underlies both the Document dependency map (Section 31) and the Impact analysis by type (Section 30). The full standard map is in Appendix C.

PART III

# Operator's guide

From first login to day-to-day document management

## 07 Login

Arianna connects to a shared PostgreSQL database: the connection profile determines which server and which storage are used.

**1 Launch Arianna.**

The Arianna Login window opens.

**2 Choose the connection profile.**

Local for a single installation, Remote for a server shared over the network (Section 36).

**3 Enter username and password.**

Credentials are the ones created under Administration → Users... (Section 33).

**4 Confirm with Enter/Log in.**

If the credentials are correct, the main window opens, with the title bar showing the connected user.

Profilo: locale

Server: localhost:5432 / arianna

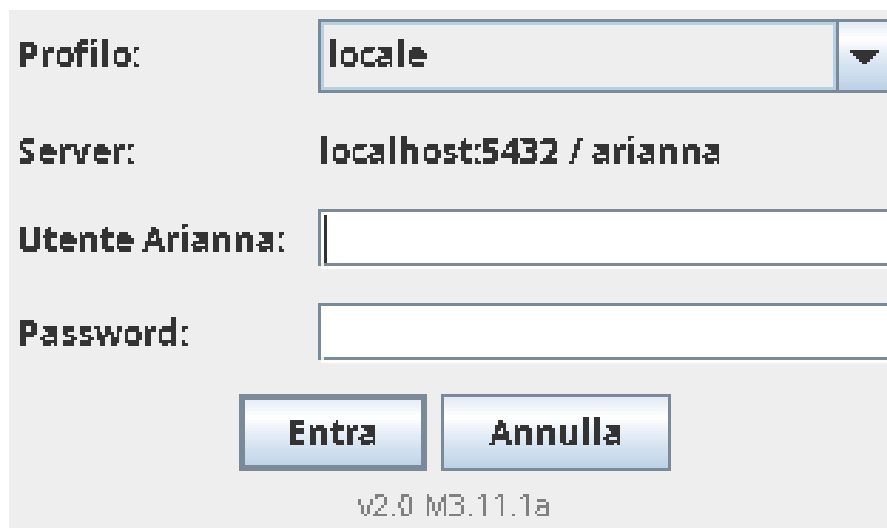
Utente Arianna: admin

Password: .....

Entra Annulla

v2.0 M3.11.1a

**Fig. 07.1** Login window — profile, server, and credentials



A login dialog box with a light gray background. It contains four labels on the left: 'Profilo:', 'Server:', 'Utente Arianna:', and 'Password:'. To the right of 'Profilo:' is a dropdown menu showing 'locale'. To the right of 'Server:' is a text field containing 'localhost:5432 / arianna'. To the right of 'Utente Arianna:' and 'Password:' are empty text input fields. Below these fields are two buttons: 'Entra' and 'Annulla'. At the bottom center, the text 'v2.0 MB.11.1a' is displayed.

|                               |                          |
|-------------------------------|--------------------------|
| Profilo:                      | locale                   |
| Server:                       | localhost:5432 / arianna |
| Utente Arianna:               |                          |
| Password:                     |                          |
| <div>Entra      Annulla</div> |                          |
| v2.0 MB.11.1a                 |                          |

**Fig. 07.2** Login — view from the Product Description

The Profile field determines which configuration file is read (Section 36): it's not just a convenience, it's the mechanism by which the same Arianna installation can point to different databases and storage without having to reconfigure them at every change of context.

## 08 Creating a client

Everything in Arianna starts from a client. It's the root of the tree and the first level of isolation between different companies' data.

**1 Open the context menu.**

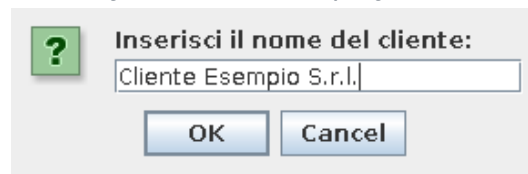
Right-click the Clients root.

**2 Add client.**

Type the company name in the dialog box, confirm with OK.

**3 The client appears in the tree.**

From here you can immediately create the first project (next section).



**Fig. 08.1** Add client — entering the company name

**NOTE**

The client's context menu also offers Rename client, Delete client, and Storage settings..., to assign that client a storage path different from the default one.

## 09 Creating a project

One or more projects are created under each client. A wizard generates the entire document container structure in one go, already sitting at Revision 0.

1

### Create the project.

Right-click the client → Create project . . . The wizard proposes every standard document container type, already checked.

2

### Choose the containers.

Use Deselect all and check only what you need, or start from a predefined Template (Section 21).

3

### Specify plants/units (optional).

If filled in, containers are created for each unit instead of directly on the project.

4

### Confirm with Create.

Arianna generates the full structure: every container is created with a Revision 0 ready to receive the first file.

Nome / codice commessa:

Commessa Demo 001

Impianti/unità (skid), uno per riga (opzionale): se compilato, i contenitori sotto vengono creati per ciascuna unità invece che direttamente sulla commessa:

Template:

Contenitori documentali da creare con Revisione 0:

|                                                            |                                       |
|------------------------------------------------------------|---------------------------------------|
| <input checked="" type="checkbox"/> P&ID                   | <input type="checkbox"/> SPECIFICHE   |
| <input checked="" type="checkbox"/> FUNCTIONAL_DESCRIPTION | <input type="checkbox"/> I/O_LIST     |
| <input type="checkbox"/> ELECTRICAL_DRAWINGS               | <input type="checkbox"/> PLC_SOFTWARE |
| <input type="checkbox"/> HMI_SCADA                         | <input type="checkbox"/> FAT          |
| <input type="checkbox"/> SAT                               | <input type="checkbox"/> MANUALI      |
| <input type="checkbox"/> CERTIFICATI                       | <input type="checkbox"/> AS_BUILT     |

Altri contenitori, uno per riga (opzionale):

Seleziona tutti Deseleziona tutti

Crea Annulla

**Fig. 09.1** Create project — name, units, template, and container selection

Nome / codice commessa:

Impianti/unità (skid), uno per riga (opzionale): se compilato, i contenitori sotto vengono creati per ciascuna unità invece che direttamente sulla commessa:

Template:

Contenitori documentali da creare con Revisione 0:

|                                                            |                                                  |
|------------------------------------------------------------|--------------------------------------------------|
| <input checked="" type="checkbox"/> P&ID                   | <input checked="" type="checkbox"/> SPECIFICHE   |
| <input checked="" type="checkbox"/> FUNCTIONAL_DESCRIPTION | <input checked="" type="checkbox"/> I/O_LIST     |
| <input checked="" type="checkbox"/> ELECTRICAL_DRAWINGS    | <input checked="" type="checkbox"/> PLC_SOFTWARE |
| <input checked="" type="checkbox"/> HMI_SCADA              | <input checked="" type="checkbox"/> FAT          |
| <input checked="" type="checkbox"/> SAT                    | <input checked="" type="checkbox"/> MANUALI      |
| <input checked="" type="checkbox"/> CERTIFICATI            | <input checked="" type="checkbox"/> AS_BUILT     |

Altri contenitori, uno per riga (opzionale):

Seleziona tutti   Deseleziona tutti

Crea   Annulla

**Fig. 09.2** "Create project" wizard — view from the Product Description

**QUICK  
ALTERNATIVE**

Add project creates an empty project, with no wizard: useful when document containers will be added one at a time later with Add document type.

## 10 Importing a file and metadata

Each container accepts one file per revision. On import, Arianna calculates the SHA-256 hash, copies the file into the managed archive, and immediately opens the metadata form.

1

### Add file.

Right-click the Revision 0 · current row of the container → Add file, choose the file from disk.

2

### Fill in the metadata.

The Document metadata dialog opens automatically: Code, Title, Discipline, Document type, Author, Approver, Notes.

3

### Save.

Metadata can be edited at any time from Document metadata . . . in the file's context menu.

|                                     |                                          |
|-------------------------------------|------------------------------------------|
| Codice:                             | FD-DEMO-001                              |
| Titolo:                             | Descrizione Funzionale Area Compressione |
| Disciplina:                         | Processo                                 |
| Tipo documento:                     | Descrizione funzionale                   |
| Autore:                             | Ufficio Tecnico                          |
| Approvatore:                        |                                          |
| Note:                               |                                          |
| <div>Salva</div> <div>Annulla</div> |                                          |

**Fig. 10.1** Document metadata — filled-in form

### TIP

Fill in at least Code and Title at import time: they're the fields Search everywhere (Section 22) relies on, and going back to fill them in later across dozens of files costs far more time than entering them up front.

## 11 Document preview

When you select a file, the right-hand panel renders the preview in the background — the window stays responsive even on a heavy PDF — and always returns full metadata: size, SHA-256 hash, import date, document code and title. Five format families are supported natively.

| Format      | Preview             |
|-------------|---------------------|
| .pdf        | first page rendered |
| .docx       | extracted text      |
| .xlsx       | extracted table     |
| .png / .jpg | scaled image        |
| .txt / .csv | text content        |

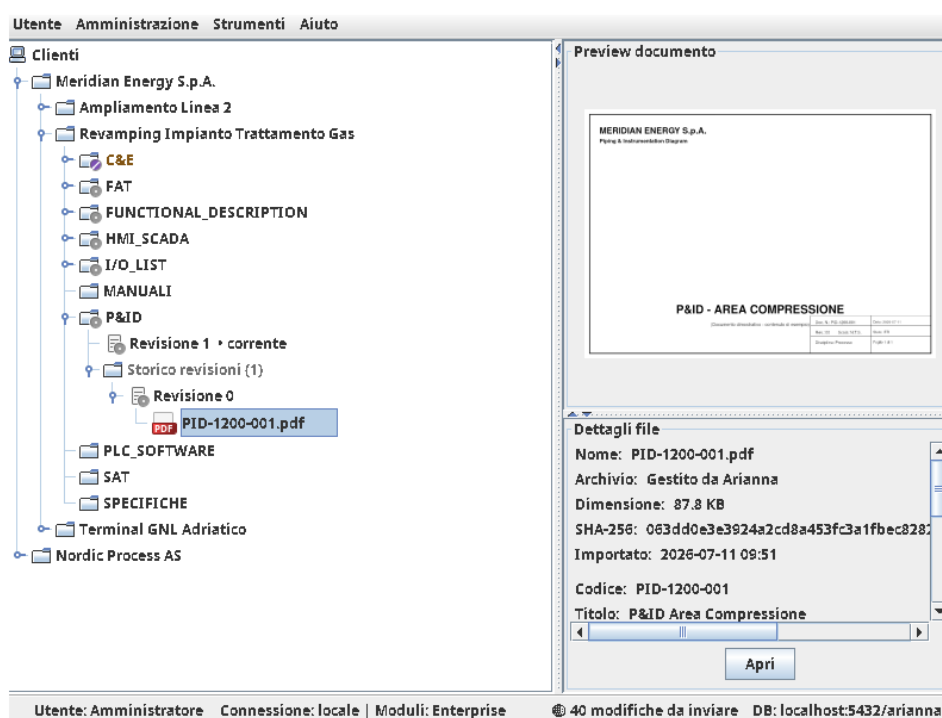


Fig. 11.1 PDF preview — P&ID

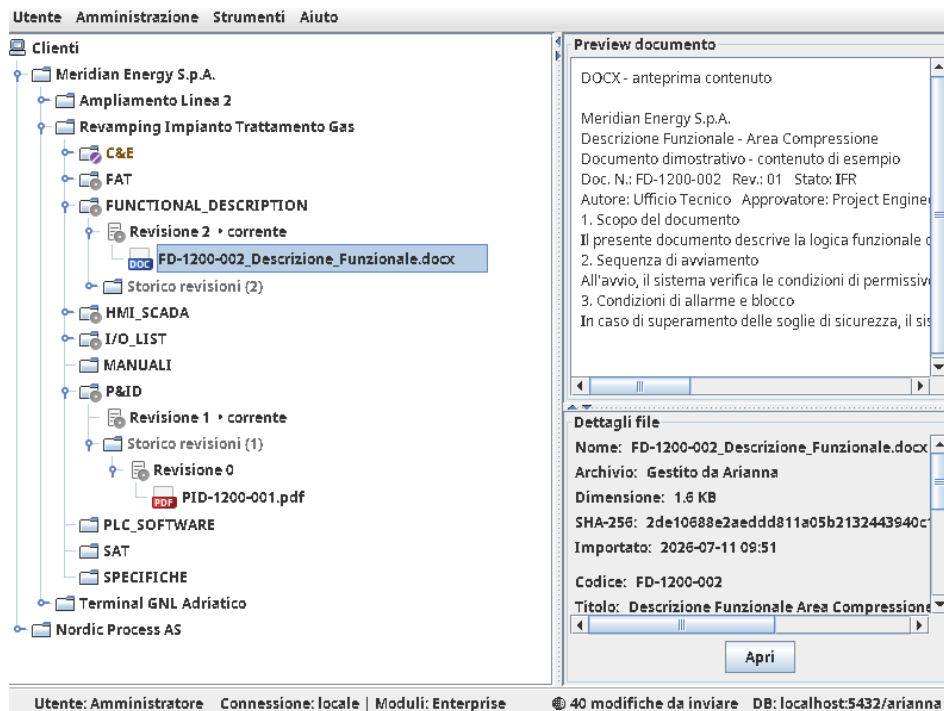


Fig. 11.2 DOCX preview — functional description

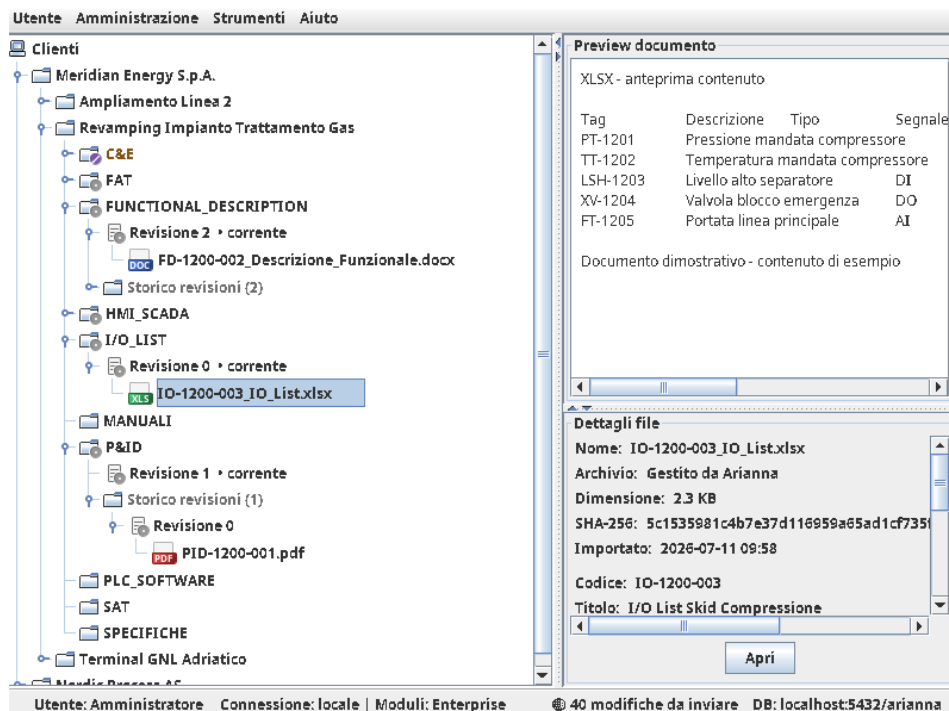
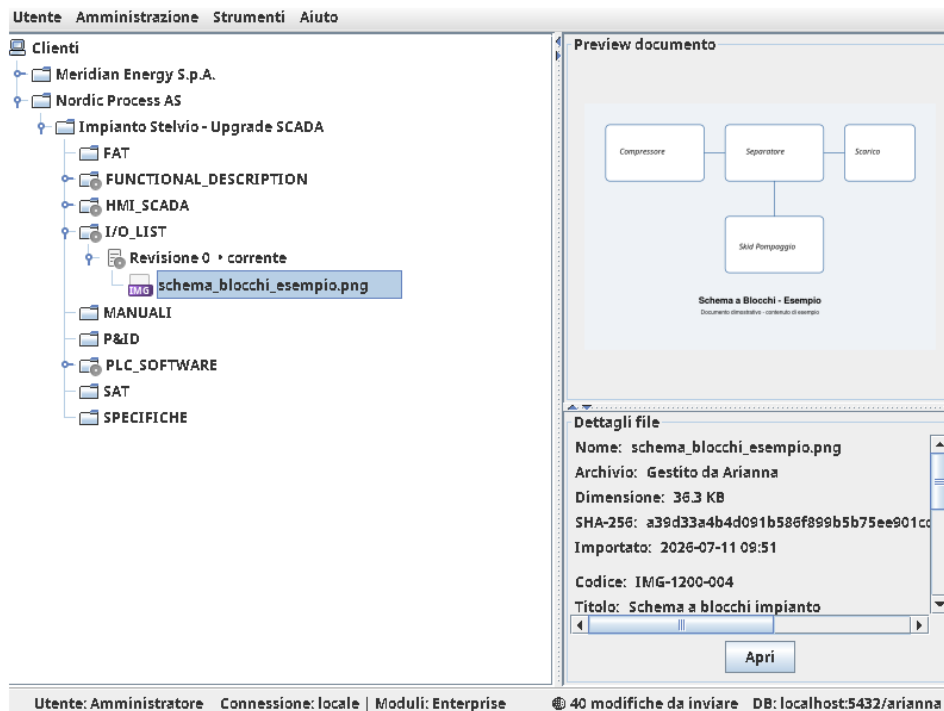
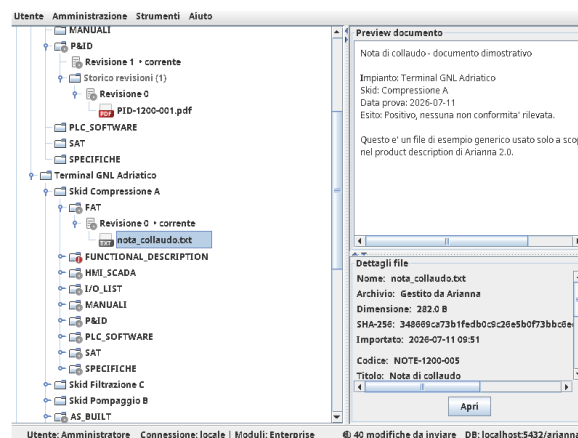


Fig. 11.3 XLSX preview — I/O list



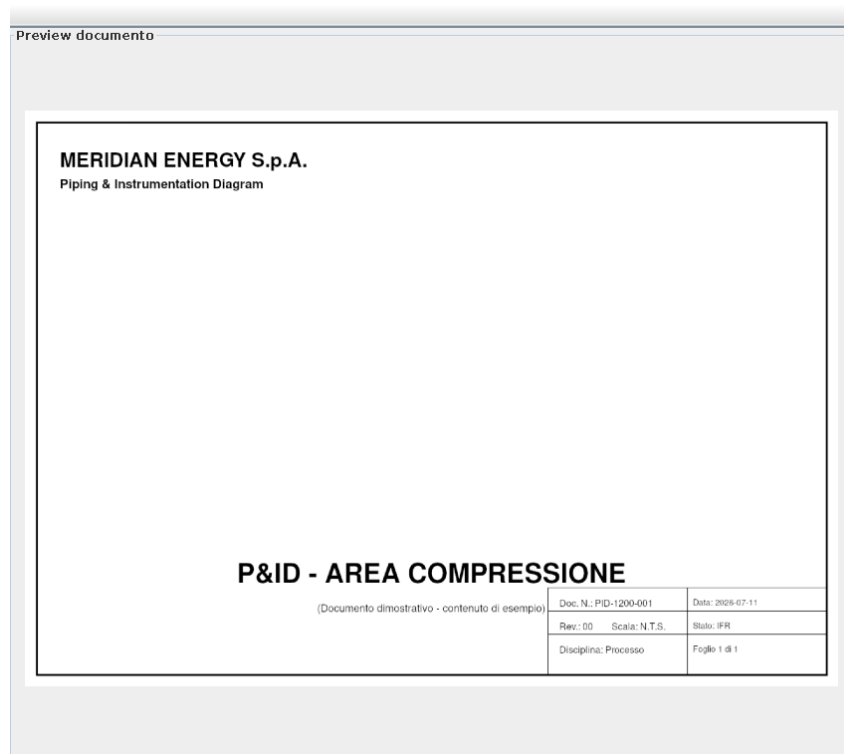
**Fig. 11.4** Image preview — block diagram



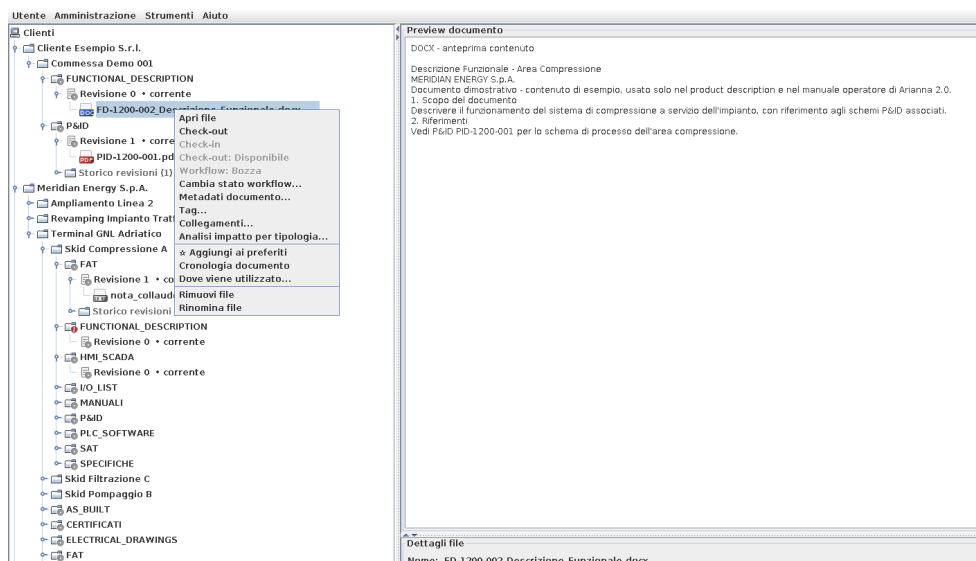
**Fig. 11.5** TXT preview — test note

## Preview and details side by side

The Details panel below the preview isn't a duplicate of the metadata: it also reports calculated information — actual size on disk, SHA-256 hash, import date — that isn't editable and exists to confirm the file being displayed is really the one expected.



**Fig. 11.6** PDF preview with the extended File details panel



**Fig. 11.7** DOCX preview with extracted text and the file's context menu

## 12 Check-out

---

To edit a file, you "borrow" it: check-out saves a working copy to the local disk and locks it for other users until it's returned via check-in.

1

### **Check-out.**

File's context menu → Check-out. Arianna asks where to save the working copy and marks the file as locked (orange padlock icon in the tree).

2

### **Edit the file**

with the appropriate external tool (CAD, Office, etc.), saving changes to that same working copy.

### **BENEFIT**

Only the user who performed the check-out can check the same file back in: no other operator can overwrite it in the meantime.

## 13 Check-in — same revision

Check-in returns the edited working copy. If the change doesn't warrant a new revision number (minor corrections, typos, formatting) it's deposited as Same revision.

### 1 Check-in.

Context menu → Check-in, select the edited file from disk.

### 2 Choose Same revision.

It's the default option in the dialog.

### 3 Confirm.

A "Check-in complete" message closes the operation; Details updates Check-out to Available and records the Last check-in date.

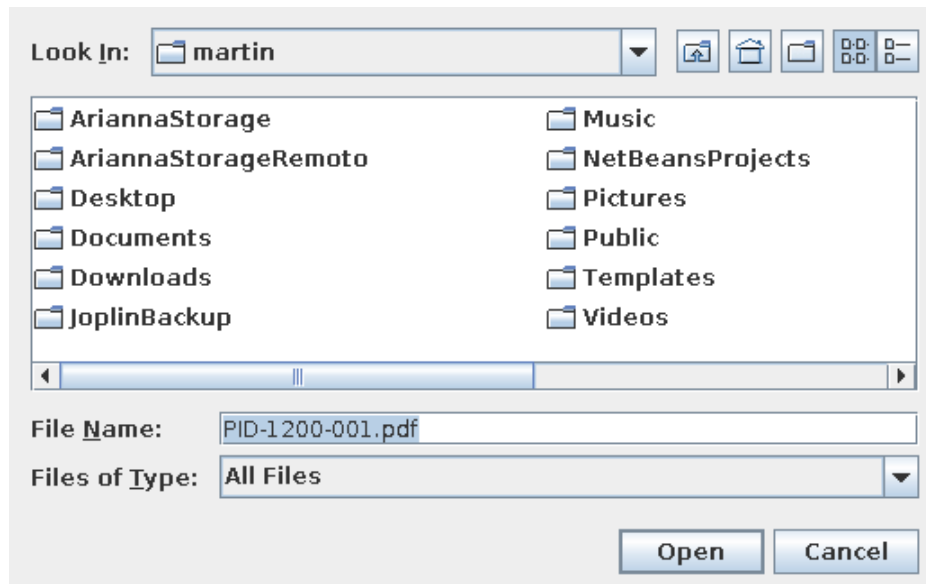


Fig. 13.1 Check-in — selecting the edited file

Revisione di destinazione:

☒ Stessa revisione (Rev. 0)

☐ Nuova revisione Numero:

Motivo revisione (obbligatorio per nuova revisione):

Redatto da:

Controllato da:

Approvato da:

Note check-in Arianna (facoltative):

OK Cancel

**Fig. 13.2** Check-in — choosing the target revision

## 14 New revision

When a change is substantial, check-in must be deposited as a new revision: Arianna increments the number, moves the previous revision into the history, and requires a reason, preparer, checker, and approver.

### 1 Select New revision

in the check-in dialog, giving the number (proposed automatically).

### 2

#### Fill in the reason and responsibility.

The Revision reason field is required; Prepared by, Checked by, and Approved by record the chain of responsibility.

### 3

#### Automatic duplicate check.

If the uploaded file is identical (same SHA-256) to the previous revision, Arianna warns and asks for confirmation before proceeding: this prevents pointless revisions created by mistake.

Revisione di destinazione:

☐ Stessa revisione {Rev. 0}

☒ Nuova revisione Numero:

Motivo revisione {obbligatorio per nuova revisione}:  
Aggiornamento layout dopo commenti revisione interna: modificata numerazione linee e aggiunta valvola di intercettazione VS-102.

Redatto da:

Controllato da:

Approvato da:

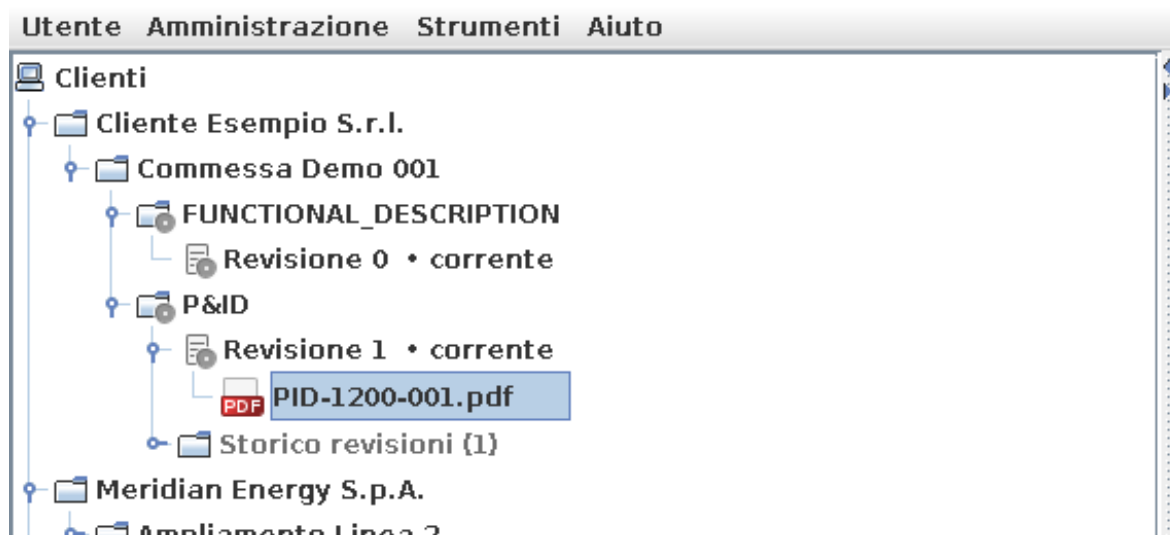
Note check-in Arianna {facoltative}:

Fig. 14.1 New revision — reason and responsibility filled in

 Il file selezionato è identico alla revisione precedente {Rev. 0}.  
SHA-256 uguale: 063dd0e3e3924a2cd8a453fc3a1fbec82827aa13de9923b919a4005e4e6ac822

Vuoi creare comunque una nuova revisione?

Fig. 14.2 Warning: file identical to the previous revision



**Fig. 14.3** The previous revision moves into "Revision history"; the new one becomes current

**CAUTION**

Metadata and tags don't carry over automatically from one revision to the next: after a new revision, check and re-enter Document metadata... and Tags... if they're needed on the new version too (Sections 10 and 16).

## 15 Workflow status

Every file follows an approval cycle: Draft → Internal review → Client review → Approved → Issued → Archived.

1

### Change workflow status.

Context menu → Change workflow status..., choose the new status from the dropdown.

2

### Add a note.

Optional, useful for explaining the status change.

3

### Save.

The badge on the tree and the Workflow field in Details update immediately.



The image shows a dialog box titled "Nuovo stato:". It contains a dropdown menu with the following options: "Bozza {DRAFT}", "Revisione interna {INTERNAL\_REVIEW}", "Revisione cliente {CUSTOMER\_REVIEW}", "Approvato {APPROVED}", "Emesso {ISSUED}", and "Archiviato {ARCHIVED}". The "Bozza {DRAFT}" option is currently selected. Below the dropdown menu are two buttons: "Salva" (Save) and "Annulla" (Cancel).

**Fig. 15.1** Change workflow status — available statuses

▲▼

Dettagli file

Nome: PID-1200-001.pdf

Archivio: Gestito da Arianna

Dimensione: 87.8 KB

SHA-256: 063dd0e3e3924a2cd8a453fc3a1fbec82827aa13de9923b919a4005e4e6ac822

Importato: 2026-07-11 11:30

Codice: PID-DEMO-001

Titolo: P&ID Area Esempio

Disciplina: Processo

Tipo: P&ID

Autore: Ufficio Tecnico

Approvatore: -

Tag: -

Workflow: Revisione interna (INTERNAL\_REVIEW)

Check-out: Disponibile

Bloccato da: -

Check-out dal: -

Ultimo check-in: 2026-07-12 16:27

Apri

42 modifiche da inviare

DB: localhost:5432/arianna

**Fig. 15.2** File details after the status change

## 16 Tags

A free-text field, one tag per line, useful for cross-cutting searches that the hierarchical model alone doesn't cover (e.g. "urgent", "client-review", "compression-area").

### 1 Open Tags...

from the file's context menu.

### 2 Type the tags, one per line.

No format constraint: they're free labels.

### 3

#### Save.

Tags appear in the File details panel and are searchable with Search everywhere (Section 22).

Tag, uno per riga:

urgente

revisione-cliente

area-compressione|

Salva

Annulla

**Fig. 16.1** Tags — one per line

#### RECOMMENDED CONVENTION

Agree on a list of recurring tags with the team (e.g. by discipline, by urgency, by phase) before using them at scale: Arianna doesn't enforce a controlled vocabulary, so "urgent" and "Urgent" remain two distinct tags.

## 17 Links between documents

Links... connects two documents with an explicit relationship type, independent of the standard map between document types (Section 06): useful for one-off relationships that don't apply to the whole document type.

### 1 Open Links...

from the file's context menu.

### 2 Add link...

Choose the linked document from the list of the entire installation and the relationship type: IMPACTS, DEPENDS\_ON, or RELATED.

### 3 Review or remove.

The list of links stays visible and editable at any time from the same dialog.

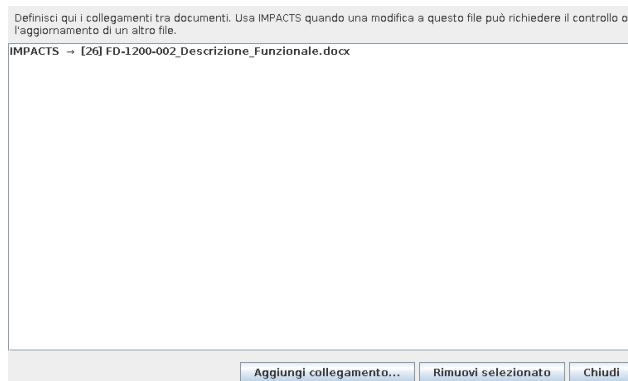


Fig. 17.1 An IMPACTS link to another document in the project

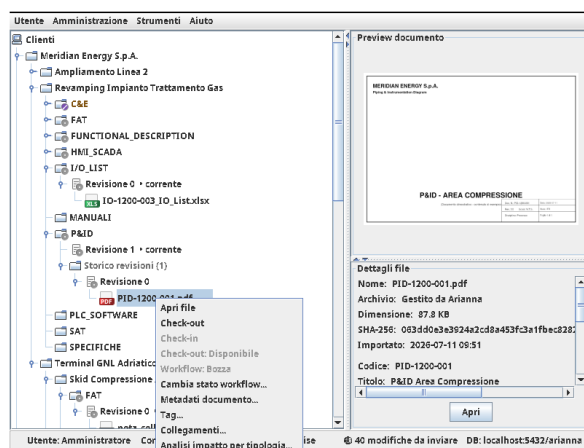


Fig. 17.2 File's context menu — Links... among the available commands

**DIFFERENCE  
FROM THE  
DEPENDENCY  
MAP**

A link (Section 17) is one-off: it connects two specific files. The dependency map (Section 31) is instead defined at the document-type level and applies automatically to every file of that type in every project. Use links for exceptions or relationships that don't follow the standard rule.

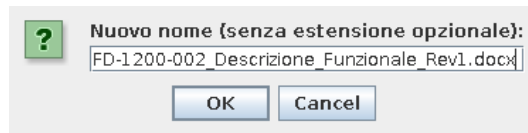
## 18 Renaming a file

---

A file's visible name can be changed without touching the physical file in the managed archive.

### 1 Rename file.

Context menu → Rename file, new name, no need to repeat the extension.



**Fig. 18.1** Rename file — new name

Renaming changes neither the SHA-256 hash nor the physical location in the managed archive: it's a safe operation that doesn't touch the file's integrity, useful for fixing a badly imported name or aligning it with a company convention adopted later on.

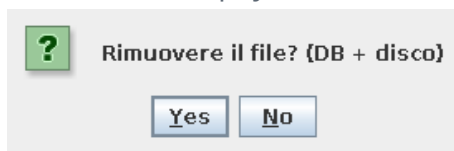
## 19 Removing a file

Destructive operations always require an explicit confirmation, to prevent accidental deletion.

1

### Remove file.

Context menu → Remove file: Arianna asks for confirmation because the operation deletes both the database record and the physical file in the archive.



**Fig. 19.1** Mandatory confirmation before removal (DB + disk)

### CAUTION

Unlike Rename file, Remove file is irreversible: there is no recycle bin and no restore from the interface. If the file is still needed as a historical reference, consider setting its workflow status to Archived (Section 15) instead of removing it.

## 20 Plants and units (skids)

Complex plant engineering projects are organized by plant/unit: each skid keeps its own documentation and revisions, under the same project.

1

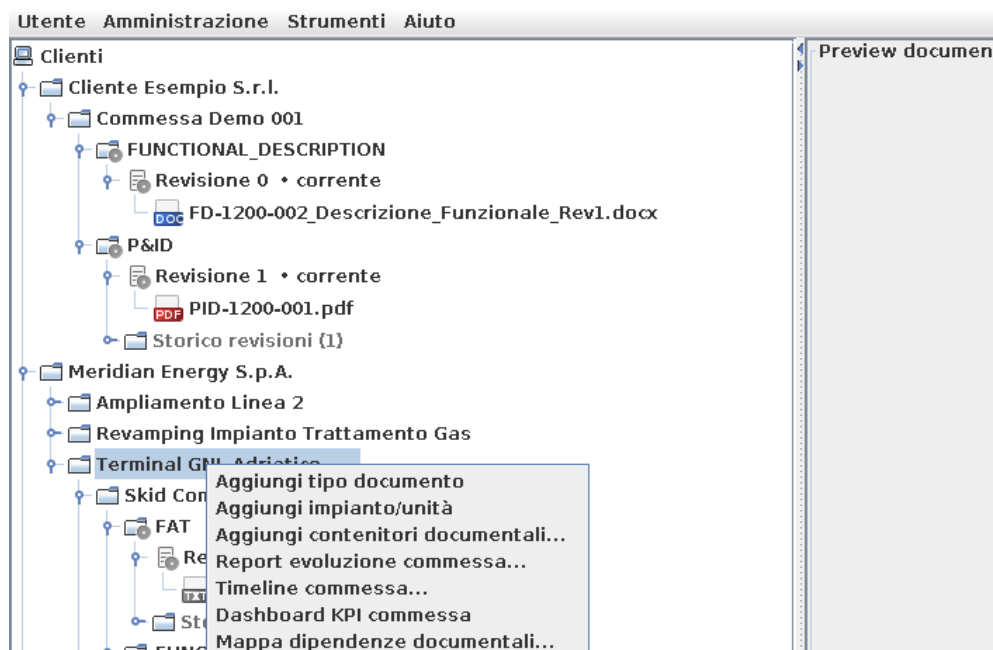
### Add plant/unit.

Right-click the project → Add plant/unit: from here on, document containers can be created per unit (e.g. "Compression Skid A", "Filtration Skid C").

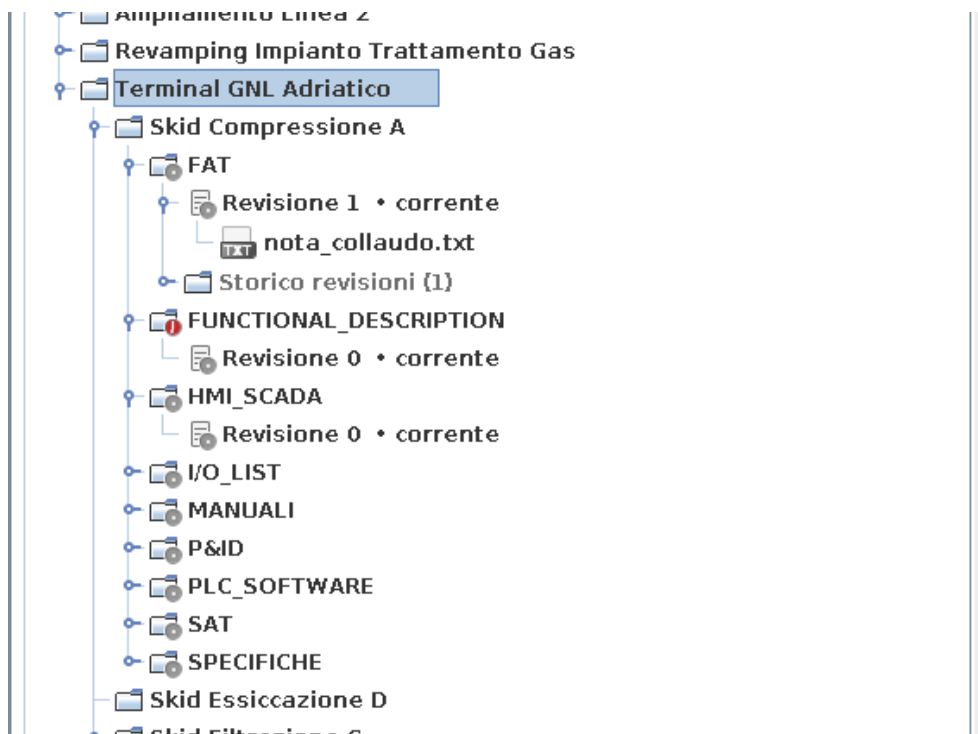
2

### Populate the unit.

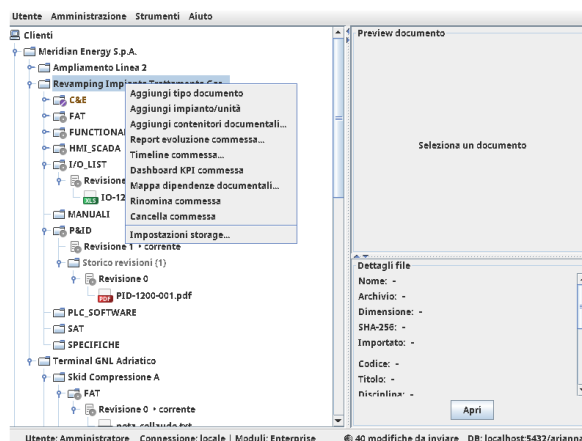
Right-click the unit → Add document containers... to assign it the document types it needs.



**Fig. 20.1** Project context menu: units, dashboard, timeline, dependency map



**Fig. 20.2** Plants/units in the project tree



**Fig. 20.3** Project context menu — view from the Product Description

## 21 Project templates

A template saves a predefined list of document containers, reusable in the project creation wizard so recurring projects don't have to start from scratch.

### 1 Open template management.

Menu Administration → Project templates....

### 2 Choose or create a template.

New to create one from scratch, or select an existing template to edit it.

### 3 Use it in the wizard.

The template appears in the Template dropdown of the Create project wizard (Section 09).

**Automazione standard**  
**Oil & Gas / Impiantistica**  
**Packaging**

**Codice template:**

OIL\_GAS\_STD

**Nome:**

Oil & Gas / Impiantistica

**Descrizione:**

VALVE\_LIST  
INSTRUMENT\_LIST  
CAUSE\_EFFECT  
I/O\_LIST  
FUNCTIONAL\_DESCRIPTION  
ELECTRICAL\_DRAWINGS  
PLC\_SOFTWARE  
FAT  
SAT  
AS\_BUILT

**Nuovo** **Salva** **Elimina** **Chiudi**

**Fig. 21.1** "Oil & Gas / Plant Engineering" template — list of predefined containers

#### TIP

Create a template for each recurring project type in your firm (e.g. "Oil & Gas / Plant Engineering", "Standard Automation", "Packaging"): the time invested once pays off on every new project of the same kind.

PART IV

# Search and navigation

Finding a document among thousands, in seconds

## 22 Search everywhere

A quick-open-style search: type, and results appear in real time, regardless of client or project.

|   |                                                                                                                                                               |
|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | <b>Open the search.</b><br>Keyboard shortcut <code>Ctrl+K</code> , available anywhere in the application.                                                     |
| 2 | <b>Type.</b><br>The filter applies to file name, code, and document title as you type, without pressing Enter.                                                |
| 3 | <b>Choose a result.</b><br>Each row shows the full path — client / project / document type / revision — to tell apart same-named files in different projects. |

PID-1200

PID-1200-001.pdf  
Cliente Esempio S.r.l. / Commessa Demo 001 / P&ID / Rev. 1

PID-1200-001.pdf  
Cliente Esempio S.r.l. / Commessa Demo 001 / P&ID / Rev. 0

PID-1200-001.pdf  
Meridian Energy S.p.A. / Revamping Impianto Trattamento Gas / P&ID / Rev. 0

**Fig. 22.1** Search everywhere — real-time results with full path

### TIP

Use Search everywhere even just to get your bearings in a project you don't know well: typing a document type's name (e.g. "P&ID") immediately gives you the list of every project that has one, without opening the tree.

## **23** Global search

---

Accessible from Tools → Global search..., it offers the same results as Search everywhere in a dedicated window, handy when you want to keep the search open next to the tree during an extended work session.

## 24 Recent and favorite documents

---

Tools → Recent documents keeps track of the last files opened in the current session, so you can get back to them without retracing the tree. Tools → Favorite documents instead shows files explicitly marked with Add to favorites from the file's context menu: a persistent bookmark for the documents you consult most often, regardless of when they were last opened.

### NOTE

Both lists are empty until you browse or explicitly add a file to favorites: they aren't pre-populated.

PART V

# Traceability and control

Always knowing where a project stands, and who did what

## 25 KPI dashboard

Summarizes a project's status at a glance: how many document types, how many revisions, how many files per workflow status, how many currently checked out.

### 1 Open the dashboard.

Project's context menu → Project KPI dashboard, or Administration → KPI dashboard . . . to pick the project from a dropdown.

### 2 Read the indicators.

Document types, Revisions, Document files, Files in check-out, plus a tile for each workflow status (Draft, In internal review, In client review, Approved, Issued, Archived), and the logged Audit events.

### 3 Refresh.

The Refresh button recalculates the counts without closing the window.

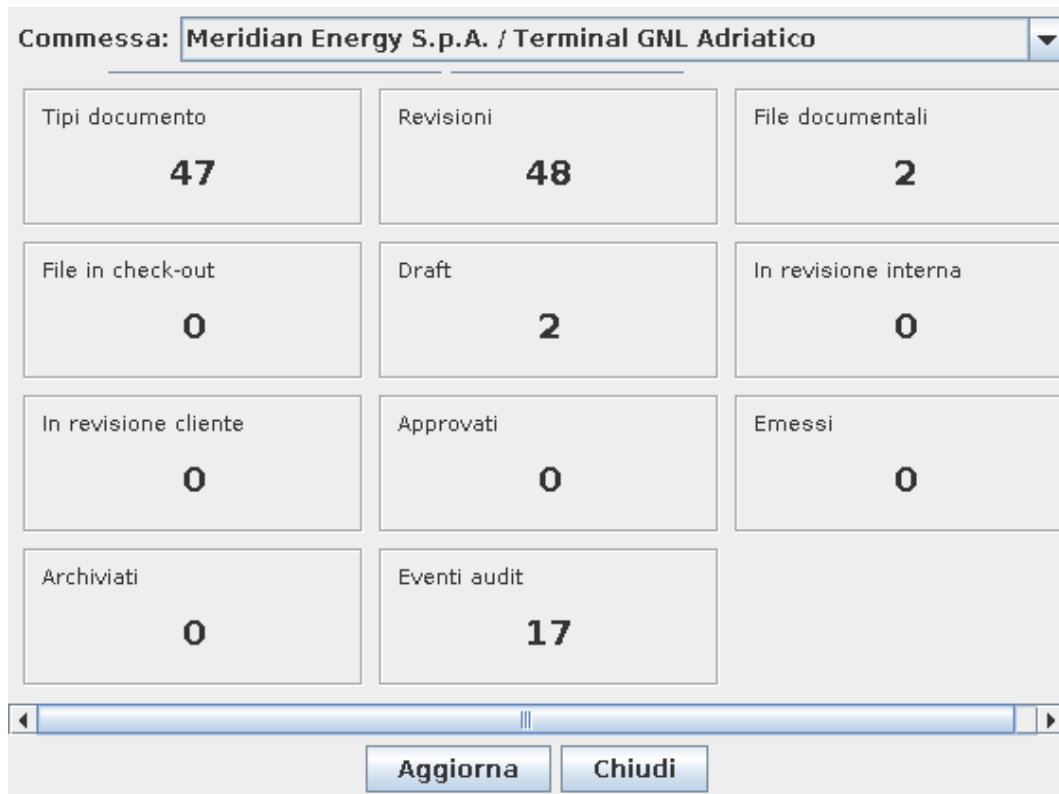


Fig. 25.1 KPI dashboard for a project

#### UNIT FILTER

When the project is organized by plant/unit, the dashboard can also be filtered by single unit/skid: useful for isolating the progress of a single plant inside a project that covers several.

## 26 Audit trail

The complete history of every action performed in Arianna: who, when, what. It's the control record of last resort, independent of any report.

### 1 Open the audit trail.

Menu Administration → Audit trail....

### 2 Filter.

By project and, where present, by single unit/skid, using the dropdowns at the top.

### 3 Read the table.

Date/Time, User, Action, Object, Description: every row is an immutable event.

| Commissa: Tutte le commesse |        | Unità: <div><div></div></div> |                | Applica filtro                   |  |
|-----------------------------|--------|-------------------------------|----------------|----------------------------------|--|
| Data/Ora                    | Utente | Azione                        | Oggetto        | Descrizione                      |  |
| 2026-07-12 16:59:27.211348  | admin  | CREATE UNITA                  | Commissa       | Creato impianto/unita'           |  |
| 2026-07-12 16:56:44.351619  | admin  | RENAME FILE                   | File           | Rinominato file                  |  |
| 2026-07-12 16:54:37.54047   | admin  | ADD DOCUMENT LINK             | File           | Aggiunto collegamento IMP...     |  |
| 2026-07-12 16:52:27.368636  | admin  | UPDATE METADATA               | File           | Aggiornati metadati docum...     |  |
| 2026-07-12 16:51:35.562112  | admin  | Import file                   | Revisione      | Importato file: FD-1200-002 ...  |  |
| 2026-07-12 16:42:51.335051  | admin  | UPDATE TAGS                   | File           | Aggiornati tag documento         |  |
| 2026-07-12 16:40:20.334048  | admin  | Import file                   | Revisione      | Nuova revisione da check-in...   |  |
| 2026-07-12 16:40:20.313324  | admin  | CHECKIN FILE                  | File           | Rev. 1 pronta per revisione ...  |  |
| 2026-07-12 16:31:57.886919  | admin  | CHECKOUT FILE                 | File           | Check-out file                   |  |
| 2026-07-12 16:29:58.447767  | admin  | Cambio workflow               | File           | Cambio stato workflow: Bozz...   |  |
| 2026-07-12 16:27:44.318363  | admin  | CHECKIN FILE                  | File           | Prima verifica dimensionale ...  |  |
| 2026-07-12 16:03:17.383525  | admin  | Import file                   | Revisione      | Nuova revisione da check-in...   |  |
| 2026-07-12 16:03:17.362816  | admin  | CHECKIN FILE                  | File           | test func                        |  |
| 2026-07-12 16:02:23.720011  | admin  | CHECKOUT FILE                 | File           | Check-out file                   |  |
| 2026-07-11 12:11:00.87981   | admin  | CHECKOUT FILE                 | File           | Check-out file                   |  |
| 2026-07-11 11:31:10.272812  | admin  | UPDATE METADATA               | File           | Aggiornati metadati docum...     |  |
| 2026-07-11 11:30:03.931543  | admin  | Import file                   | Revisione      | Importato file: PID-1200-001...  |  |
| 2026-07-11 11:25:37.964496  | admin  | Creazione commessa            | Commissa       | Creata commessa 'Commes...       |  |
| 2026-07-11 11:20:52.171542  | admin  | CREATE CUENTE                 | CUENTE         | Creato cliente                   |  |
| 2026-07-10 18:01:44.33734   | admin  | ADD CONTAINERS                | Impianto/unità | Aggiunti 17 contenitori in R...  |  |
| 2026-07-10 18:01:21.602569  | admin  | RENAME UNITA                  | Impianto/unità | Rinominato impianto/unita'       |  |
| 2026-07-10 18:01:13.697664  | admin  | CREATE UNITA                  | Commissa       | Creato impianto/unita'           |  |
| 2026-07-10 18:00:43.064248  | admin  | SET DUE DATE                  | Tipo documento | Impostata scadenza prevista      |  |
| 2026-07-10 18:00:20.629427  | admin  | SET DUE DATE                  | Tipo documento | Impostata scadenza prevista      |  |
| 2026-07-10 17:59:44.835745  | admin  | UPDATE METADATA               | File           | Aggiornati metadati docum...     |  |
| 2026-07-10 17:58:54.719393  | admin  | Import file                   | Revisione      | Importato file: Pasted Text.txt  |  |
| 2026-07-10 17:58:11.828337  | admin  | SET DUE DATE                  | Tipo documento | Impostata scadenza prevista      |  |
| 2026-07-10 17:46:55.217342  | admin  | ADD CONTAINERS                | Impianto/unità | Aggiunti 9 contenitori in Rev... |  |
| 2026-07-10 17:46:44.15268   | admin  | CREATE UNITA                  | Commissa       | Creato impianto/unita'           |  |
| Aggiorna                    |        | Chiudi                        |                |                                  |  |

Fig. 26.1 Audit trail — history of actions with user, action, and object

#### WHY IT'S DIFFERENT FROM THE TIMELINE AND THE REPORT

The audit trail logs every action at the system level, including administrative ones (user creation, role changes, configuration changes): it's the most granular and least filterable record. The Timeline and the Report (Sections 27-28) are editorial readouts instead, meant to be shared with people who don't have access to Arianna.

## 27 Project timeline

A text-based, chronologically ordered account of every document event in a project: creation, import, revisions, workflow changes. It's meant to be read or pasted into a report, not just viewed on screen.

### 1 Open the timeline.

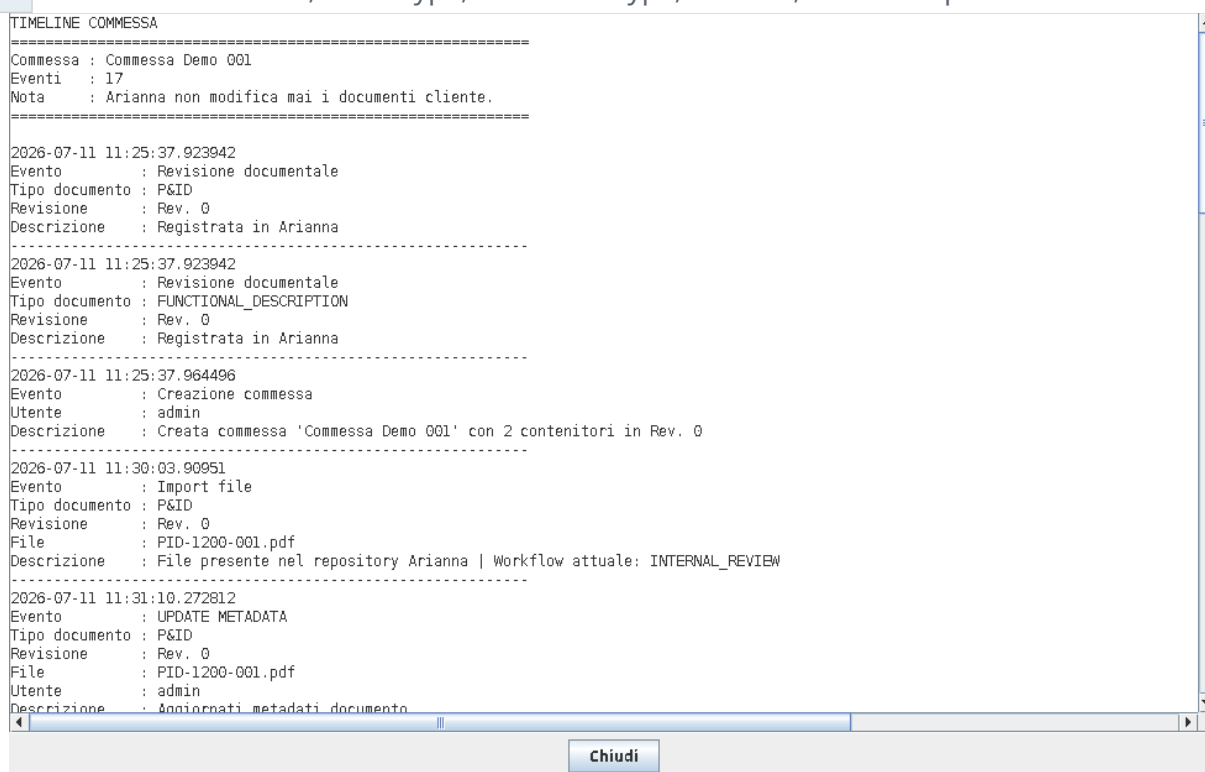
Project's context menu → Project timeline....

### 2 Read the header.

It reports the project, the total number of events, and a reminder of the core principle: "Arianna never modifies client documents".

### 3 Scroll through the events.

Each with date/time, event type, document type, revision, and description.



**Fig. 27.1** Project timeline — text-based chronological account

Every line of the timeline reports the event in a format that reads fine outside Arianna too: date and time down to the microsecond, event type, document type and revision involved, and a plain-language description. It's the most suitable format when the history needs to be attached to a communication with the client or with a regulatory body.

## 28 Project evolution report

Like the timeline, but with a quantitative summary up front and a filter by document type: meant to be generated on demand and exported as text.

### 1 Open the report.

Project's context menu → Project evolution report....

### 2 Filter by document type.

Dropdown Document type: "Entire project" or a single type.

### 3 Generate report.

The summary shows document types analyzed, revisions, files, workflow changes, first and last event; below, the detailed chronological evolution.

### 4 Copy text.

To paste the report into an email or an external document.

**Filtro report**

Tipo documento: Tutta la commessa Genera report

REPORT EVOLUZIONE COMMESSA

=====

Commessa : Commessa Demo 001  
Tipo documento : Tutta la commessa  
Criterio : storia completa della commessa  
Nota : Arianna non modifica mai i documenti cliente originali.  
I dati sotto riportati sono tracciati nel database Arianna.

=====

RIEPILOGO

-----

Tipi documento analizzati : 2  
Revisioni documentali : 3  
File documentali : 3  
Cambi workflow : 1  
Primo evento : 2026-07-11 11:25:37.923942  
Ultimo evento : 2026-07-12 16:51:35.539778

EVOLUZIONE CRONOLOGICA

-----

2026-07-11 11:25:37.923942  
Evento : Revisione documentale  
Tipo documento : P&ID  
Revisione : Rev. 0  
Descrizione : Registrata revisione in Arianna

-----

2026-07-11 11:25:37.923942  
Evento : Revisione documentale  
Tipo documento : FUNCTIONAL\_DESCRIPTION  
Revisione : Rev. 0  
Descrizione : Registrata revisione in Arianna

-----

2026-07-11 11:25:37.964496  
Evento : Creazione commessa  
Tipo documento : -  
Utente : admin  
Descrizione : Crea commessa 'Commessa Demo 001' con 2 contenitori in Rev. 0

-----

Copia testo Chiudi

**Fig. 28.1** Project evolution report — filter, summary, and chronological evolution

#### TIP

Generate the report filtered on a single document type before a progress meeting on that discipline: the quantitative summary up front saves the time of reconstructing the numbers by hand from the full timeline.

## 29 Document history and Where used

Two entries in the file's context menu that answer different but complementary questions.

## Document history

List of the points in the tree (File, Arianna path, Date, User/Type) where this specific file was opened or referenced over time.

## Where used

List of the documents that depend on this file via an explicit link (Section 17): answers "if I change this, what else do I need to check?"

**Fig. 29.1** Document history — view structure (empty table on a freshly imported file)

| File | Percorso Arianna | Data | Utente/Tipo |
|------|------------------|------|-------------|
|      |                  |      |             |

Vai nell'albero

Apri

Chiudi

**Fig. 29.2** Where used — empty until another document points to this file

**NOTE**

Both views are empty on a freshly imported file with no incoming links: that's not a bug, it's the correct initial state. They fill in as the file is opened or linked from other documents.

## 30 Impact analysis by type

---

Available from the file's context menu, it cross-references the standard dependency map (Section 06) with the files actually present in a project, to answer "if I change this document type, which other document types in this project could be impacted, and already have a file loaded?". It sits alongside the Document dependency map (Section 31), which shows the same network of relationships as a browsable tree.

### 1 Open the analysis.

File's context menu → Impact analysis by type....

### 2 Read the list of impacts.

Document types linked with an IMPACTS relationship to the current type, with an indication of which already have a file loaded in this project.

### 3 Plan the checks.

Use the list as a checklist of what to verify before releasing the change in progress.

Unlike the Dependency map (Section 31), which shows the project's entire FEEDS/IMPACTS network, Impact analysis starts from a single file and answers a specific operational question: useful right when you're about to check in a file and want to know immediately who else to notify.

## 31 Document dependency map

This map is based on document types, not individual files. In a project's lifecycle the root is usually the Narrative / Process Description; in operational impact analysis the central document is often the P&ID.

### 1 Open the map.

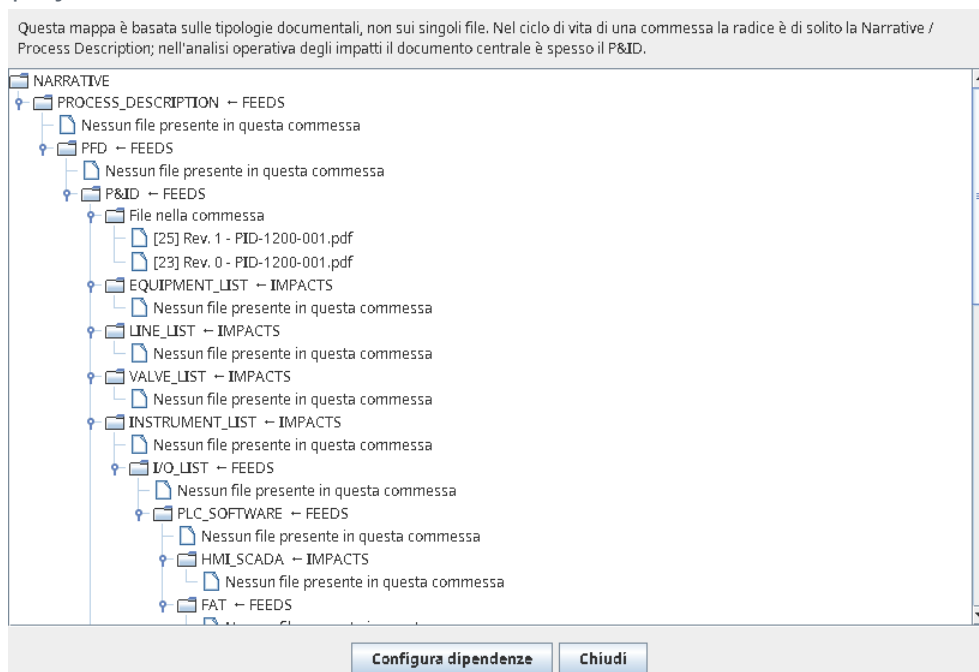
Project's context menu → Document dependency map . . .

### 2 Browse the tree.

Each node is a document type, labeled FEEDS or IMPACTS toward its parent node; under each type, the files actually present in this project.

### 3 Configure dependencies.

Opens the global editor for the standard map (Section 06 and Appendix C), shared by every project.

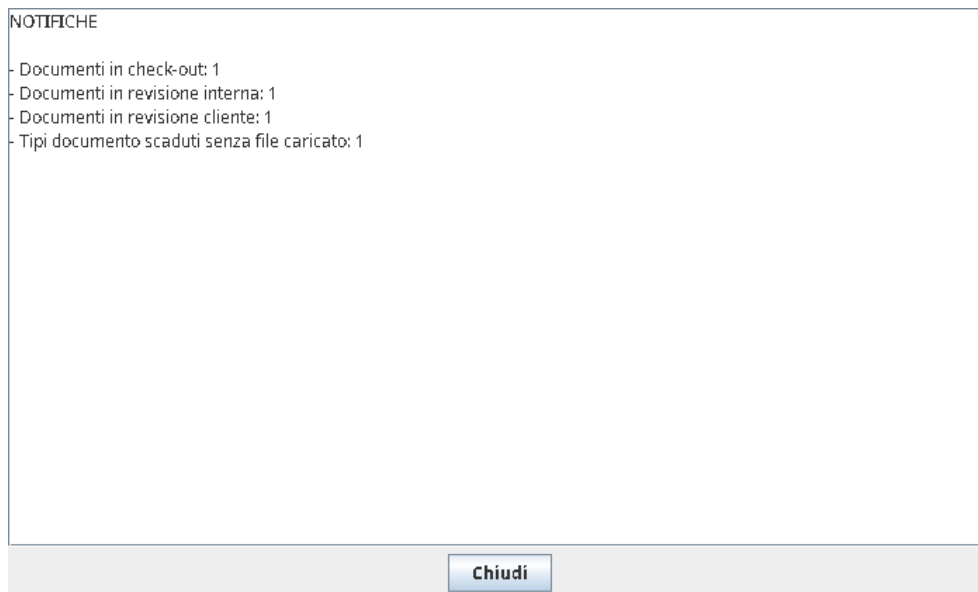


**Fig. 31.1** Document dependency map — FEEDS/IMPACTS tree with the project's actual files

## 32 Operational notifications

---

A concise digest of situations that need attention, updated in real time.



**Fig. 32.1** Operational notifications — checked-out, in-review, overdue documents

The digest reports the number of documents in check-out, in internal review, in client review, and the number of document types overdue with no file loaded — the same signal that drives the red exclamation-mark badge on the tree (Appendix A).

PART VI

# Administration

Users, permissions, configuration, and active modules

## 33 User management

The application users panel manages the accounts used to log into Arianna — independent of operating-system or database users.

- 1 Open user management.

Menu Administration → Users...

**2** Review the list.

Username, Name, Admin, Status, Password change, Created, Last access.

**Manage a user.**

New, Edit, Reset password, Roles... to assign roles (Section 34), Enable/Disable to suspend an account without deleting it.

[illegible]

**Fig. 33.1** Application user management

## 34 Roles and permissions

Each role bundles a subset of specific permissions on clients, projects, documents, revisions, and files. A user can hold more than one role at a time.

### 1 Open Roles and permissions.

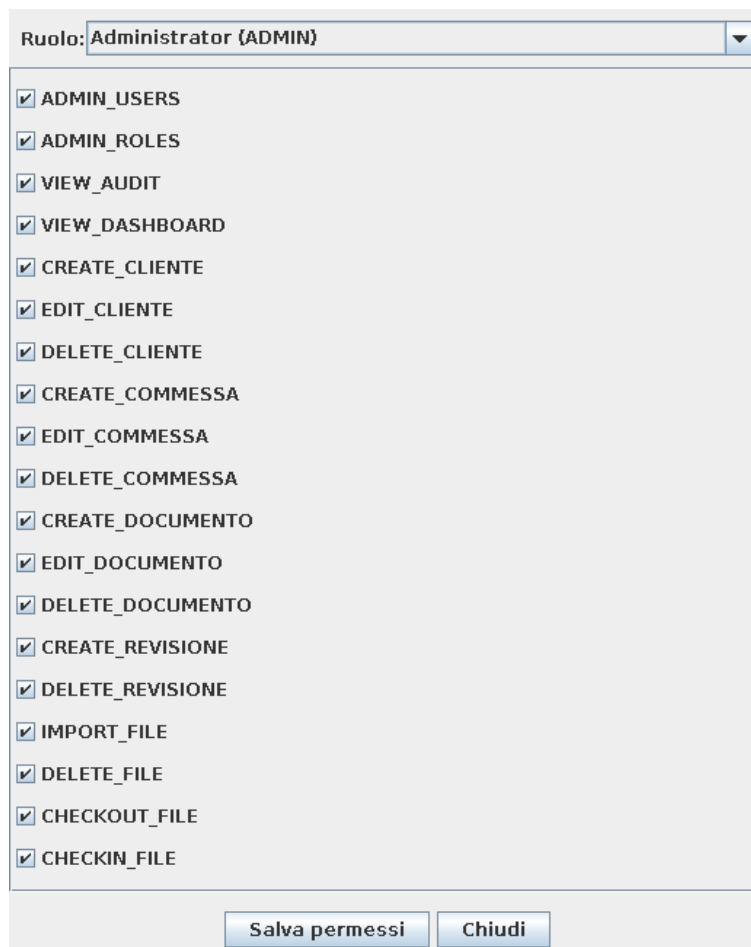
Menu Administration → Roles and permissions....

### 2 Choose the role.

Dropdown at the top: permissions update to reflect that role.

### 3 Check or clear permissions.

Each permission is an independent checkbox; Save permissions applies the changes.



The screenshot shows a web interface for managing roles and permissions. At the top, there is a dropdown menu labeled 'Ruolo:' with 'Administrator {ADMIN}' selected. Below this is a list of permissions, each with a checked checkbox and a label. The permissions are: ADMIN\_USERS, ADMIN\_ROLES, VIEW\_AUDIT, VIEW\_DASHBOARD, CREATE\_CLIENTE, EDIT\_CLIENTE, DELETE\_CLIENTE, CREATE\_COMMESA, EDIT\_COMMESA, DELETE\_COMMESA, CREATE\_DOCUMENTO, EDIT\_DOCUMENTO, DELETE\_DOCUMENTO, CREATE\_REVISIONE, DELETE\_REVISIONE, IMPORT\_FILE, DELETE\_FILE, CHECKOUT\_FILE, and CHECKIN\_FILE. At the bottom of the dialog, there are two buttons: 'Salva permessi' and 'Chiudi'.

| Ruolo:                |
|-----------------------|
| Administrator {ADMIN} |

- ☒ ADMIN\_USERS
- ☒ ADMIN\_ROLES
- ☒ VIEW\_AUDIT
- ☒ VIEW\_DASHBOARD
- ☒ CREATE\_CLIENTE
- ☒ EDIT\_CLIENTE
- ☒ DELETE\_CLIENTE
- ☒ CREATE\_COMMESA
- ☒ EDIT\_COMMESA
- ☒ DELETE\_COMMESA
- ☒ CREATE\_DOCUMENTO
- ☒ EDIT\_DOCUMENTO
- ☒ DELETE\_DOCUMENTO
- ☒ CREATE\_REVISIONE
- ☒ DELETE\_REVISIONE
- ☒ IMPORT\_FILE
- ☒ DELETE\_FILE
- ☒ CHECKOUT\_FILE
- ☒ CHECKIN\_FILE

Salva permessi Chiudi

**Fig. 34.1** Roles and permissions — the Administrator role

## The five predefined roles

| Role                | Typical use                                                                                |
|---------------------|--------------------------------------------------------------------------------------------|
| Administrator       | Full access: users, roles, configuration, every operation on documents.                    |
| Document Controller | Operational document management: import, revisions, workflow, without user administration. |
| Engineering         | Technical production: importing and revising the documents within their own discipline.    |
| Project Manager     | Visibility into dashboards, audit trail, and reports, with limited edit permissions.       |
| Viewer              | Read-only: opening and previewing documents, no changes.                                   |

## 35 Server and storage configuration

The active connection profile determines the database server and the shared storage path. This panel shows and lets you edit it.

### 1 Open the configuration.

Menu Administration → Server/storage configuration.

### 2 Check the parameters.

PostgreSQL server, Port, Database name, Admin DB, DB user, Password, Shared storage.

### 3 Test and save.

Test connection verifies the parameters before Save; Cancel discards the changes.

The screenshot shows a configuration window with the following fields and controls:

- Profilo connessione:** A dropdown menu currently set to "locale".
- Server PostgreSQL:** A text input field containing "localhost".
- Porta:** A text input field containing "5432".
- Nome database:** A text input field containing "arianna".
- DB amministrativo:** A text input field containing "postgres".
- Utente DB:** A text input field containing "arianna\_app".
- Password:** A password input field with masked characters (dots).
- Storage condiviso:** A text input field containing "/home/martin/AriannaStorage" and a button labeled "Sfoggia..." (Browse...) to the right.

Below the fields, there is a paragraph of Italian text: "Il profilo connessione separa locale e remoto. Cambiando profilo Arianna carica un file properties diverso e non sovrascrive piu' i parametri dell'altro profilo. Il campo Utente DB e' l'utente PostgreSQL tecnico normalmente arianna\_app non e' l'utente del login Arianna."

At the bottom right, there are three buttons: "Test connessione", "Annulla", and "Salva".

**Fig. 35.1** Server and storage configuration — connection parameters

#### CAUTION

The DB user field is the technical PostgreSQL user (normally arianna\_app), not the Arianna login user: they're two distinct credentials with different purposes.

## 36 Connection profile and multi-site synchronization

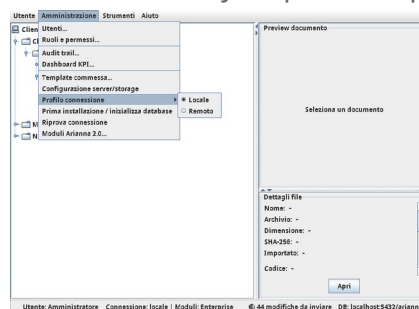
The profile separates the local configuration from the remote one: switching profile makes Arianna load a different properties file and doesn't overwrite the other profile's parameters.

### 1 Switch profile.

Menu Administration → Connection profile → Local or Remote.

### 2 Check the status.

Tools → Sync status shows whether the sync queue is up to date.



**Fig. 36.1** Connection profile — Local / Remote

When several sites share the same projects, a sync queue reliably propagates creations, renames, and deletions across the connected sites: every local operation is queued and replicated as soon as a connection is available, without blocking the operator's work while offline.

## 37 Arianna 2.0 modules

An installation can activate only the modules a given client or use case needs, starting from three predefined profiles — Base, Engineering, Enterprise — or composing a custom combination. The Core module is always active.

### 1 Open module management.

Menu Administration → Arianna 2.0 modules....

### 2 Choose a profile or customize.

Base, Engineering, or Enterprise pre-fill the selection; individual checkboxes remain editable.

### 3 Save.

The active combination determines which menu items and features are available throughout the application.

**Profilo installazione: Enterprise**  
Attiva solo le funzionalità necessarie per questo cliente/uso. Il Core resta sempre attivo.

| Attivo                              | Modulo                   | Descrizione                                                               |
|-------------------------------------|--------------------------|---------------------------------------------------------------------------|
| <input checked="" type="checkbox"/> | Ricerca documentale      | Ricerca globale su clienti, commesse e documenti.                         |
| <input checked="" type="checkbox"/> | Workflow documentale     | Stati documento, avanzamento e controllo emissioni.                       |
| <input checked="" type="checkbox"/> | Metadati tecnici         | Campi tecnici associati ai documenti.                                     |
| <input checked="" type="checkbox"/> | Tag library              | Classificazione tramite tag e parole chiave controllate.                  |
| <input checked="" type="checkbox"/> | Dipendenze tra tipologie | Relazioni standard tra tipologie documentali.                             |
| <input checked="" type="checkbox"/> | Analisi impatto          | Analisi dei documenti potenzialmente impattati da una modifica.           |
| <input checked="" type="checkbox"/> | Timeline commessa        | Vista cronologica degli eventi di commessa.                               |
| <input checked="" type="checkbox"/> | Dashboard KPI            | Indicatori sintetici su documenti, workflow e blocchi.                    |
| <input checked="" type="checkbox"/> | Audit trail              | Tracciamento storico delle operazioni.                                    |
| <input checked="" type="checkbox"/> | Notifiche operative      | Promemoria, scadenze e messaggi operativi.                                |
| <input checked="" type="checkbox"/> | Template commessa        | Creazione guidata di commesse da template.                                |
| <input checked="" type="checkbox"/> | Permessi avanzati        | Gestione granulare di ruoli e permessi.                                   |
| <input checked="" type="checkbox"/> | Multiutente / rete       | Funzioni pensate per uso condiviso in rete: lock, checkout e concorrenza. |

Base

Engineering

Enterprise

Salva

Chiudi

Fig. 37.1 Arianna 2.0 modules — full matrix of activatable modules

### The twelve modules

| Module | Description |
|--------|-------------|
|--------|-------------|

|                           |                                                                                |
|---------------------------|--------------------------------------------------------------------------------|
| Document search           | Global search across clients, projects, and documents.                         |
| Document workflow         | Document statuses, progress, and issue control.                                |
| Technical metadata        | Technical fields attached to documents.                                        |
| Tag library               | Classification via tags and controlled keywords.                               |
| Type dependencies         | Standard relationships between document types.                                 |
| Impact analysis           | Analysis of documents potentially impacted by a change.                        |
| Project timeline          | Chronological view of project events.                                          |
| KPI dashboard             | Summary indicators on documents, workflow, and blockers.                       |
| Audit trail               | Historical tracking of operations.                                             |
| Operational notifications | Reminders, deadlines, and operational messages.                                |
| Project templates         | Guided creation of projects from templates.                                    |
| Advanced permissions      | Granular management of roles and permissions.                                  |
| Multi-user / network      | Functions designed for shared network use: locking, checkout, and concurrency. |

## 38 Personal account

The User menu gathers operations on your own account, always accessible regardless of role.

**Change password...**

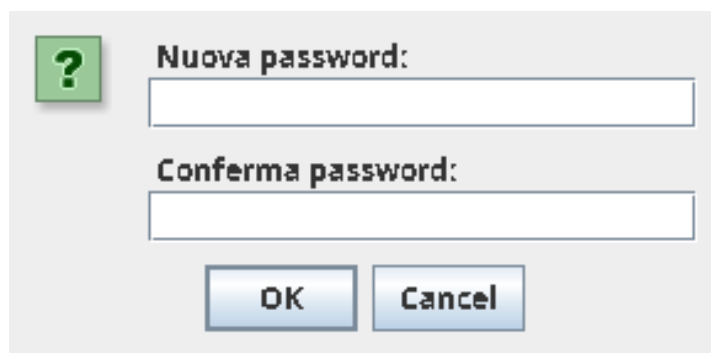
Sets a new password for the currently logged-in user, with double-entry confirmation.

**Switch user / Logout...**

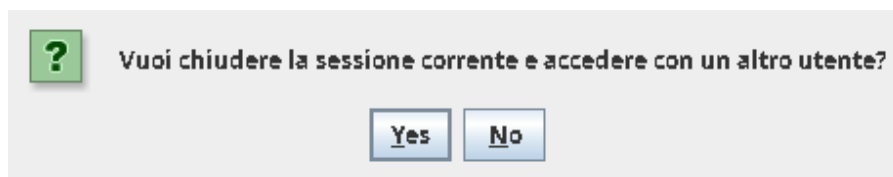
Closes the current session and returns to the login window, without closing the application.

**Exit**

Closes Arianna.



**Fig. 38.1** Change password



**Fig. 38.2** Switch user / Logout — confirmation

## PART VII

# Case study

A complete narrative walkthrough, from an empty project to an issued document

## 39 From an empty project to the first issued document

---

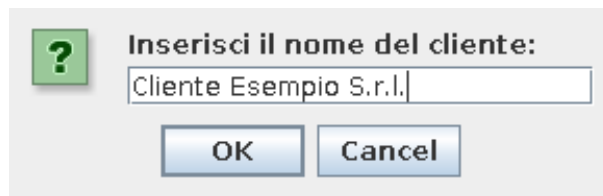
The previous sections describe each feature in isolation. This chapter puts them back together into a single narrative: the — realistic, fictitiously named — story of how a document is born, gets revised, and ends up issued, seen through Arianna from start to finish.

### The setting

A small engineering firm, Cliente Esempio S.r.l., is engaged to develop the process documentation for a new plant. The document controller opens Arianna and starts by setting up the workspace for the new project.

### Day one: the structure

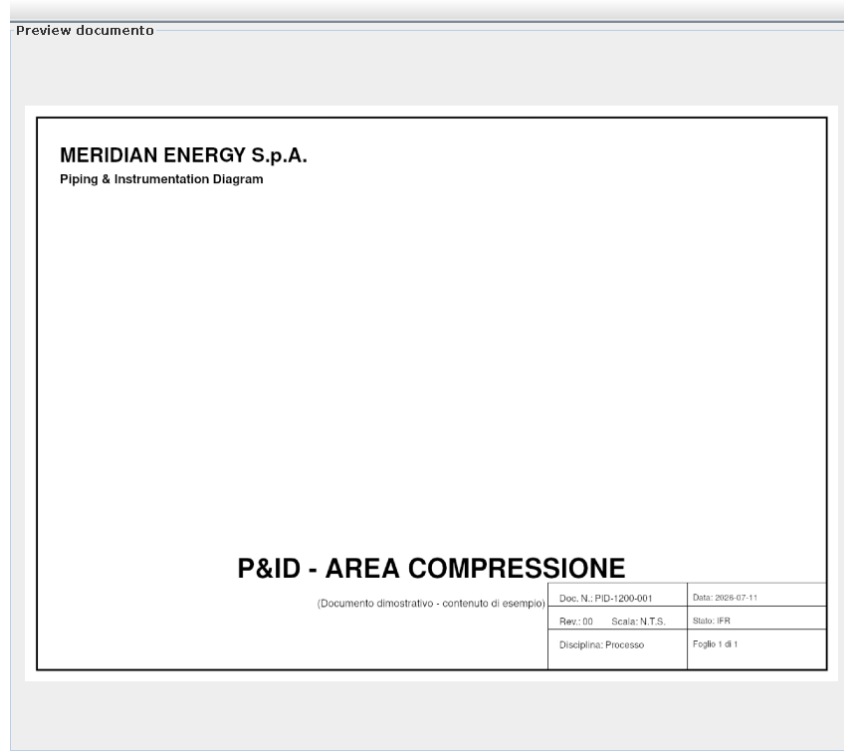
The first step isn't importing files, it's building the scaffolding that will hold them. Right-click the Clients root, Add client, and within seconds "Cliente Esempio S.r.l." appears in the tree (Section 08). From there, right-click the client and Create project...: the controller names the project "Commessa Demo 001" and, instead of checking every standard container, uses Deselect all and selects only P&ID and FUNCTIONAL\_DESCRIPTION — at this stage the assignment only covers process development, other containers will arrive later via Add document type (Section 09).



**Fig. 39.1** The client takes shape in the tree

### The first document

With the structure ready, the controller imports the first P&ID for the compression area: right-click Revision 0 · current under P&ID, Add file, and picks PID-1200-001.pdf from disk. The Document metadata dialog opens on its own; fills in Code (PID-DEMO-001), Title ("P&ID Compression Area"), Discipline ("Process"), Author ("Technical Office") — Section 10. Selecting the file, the preview panel shows the drawing without leaving Arianna: confirming the loaded file is the right one takes a glance, not a double-click to open it elsewhere.



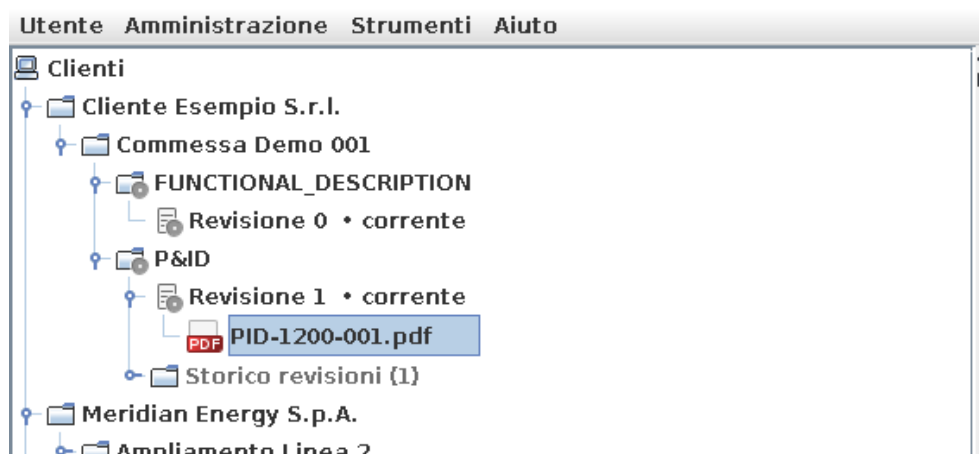
**Fig. 39.2** The imported P&ID, with preview and details

## A change on the fly

Two days later, an internal comment comes in: a detail on the bypass line needs correcting. The controller checks out the P&ID (Section 12): Arianna saves a local working copy and marks the file as locked — anyone else opening the same project immediately sees that document is "in progress". The correction is minor, it doesn't change the substance of the drawing: on return, the check-in is deposited as Same revision (Section 13). No revision number changes, but the "Last check-in" date in File details updates, and anyone can verify when the last intervention happened.

## The substantial change

A week later, the client asks for an isolation valve not shown in the original drawing to be added: this is no longer a detail, it's a change that alters the document's technical meaning. This time the check-in must be deposited as New revision (Section 14). The controller fills in the required Revision reason field ("Layout update after internal review comments: renumbered lines and added an isolation valve"), and the Prepared by, Checked by, Approved by fields with the names of whoever actually worked on the change. Arianna bumps the revision to Rev. 1; Rev. 0 doesn't disappear, it moves into Revision history, always available.



**Fig. 39.3** Rev. 0 in the history, Rev. 1 current

## The approval cycle

With Rev. 1 deposited, the document isn't ready for the client yet: it must first go through internal review. Change workflow status... moves the P&ID from Draft to Internal review (Section 15); a colleague at the same firm checks it, finds nothing else to correct, and the status advances to Client review. The client approves it without comments: status Approved. At this point the document is ready to be formally released: status Issued — the badge on the tree, green and unambiguous, tells this whole story to anyone looking at it, without opening the file.

## Tracking the connections

Meanwhile, the controller also imports the related Functional Description (FUNCTIONAL\_DESCRIPTION), and uses Links... on the P&ID to tag it with an IMPACTS relationship: a future change to the P&ID will need to trigger a check on that document (Section 17). A few free tags are added too — "compression-area", "client-review" — to be able to find it later with Search everywhere, even by someone who doesn't remember which project it's in (Sections 16 and 22).

## Looking back

At month end, the controller opens the project's KPI dashboard for the internal report: two document types, three revisions overall, one file with Issued status (Section 25). For the client, a Project evolution report filtered on P&ID is prepared instead, with Copy text ready to paste into an update email (Section 28). If it were ever necessary to reconstruct with certainty who approved what and when, the Audit trail stays there, immutable, available (Section 26).

**THE POINT OF  
ALL THIS**

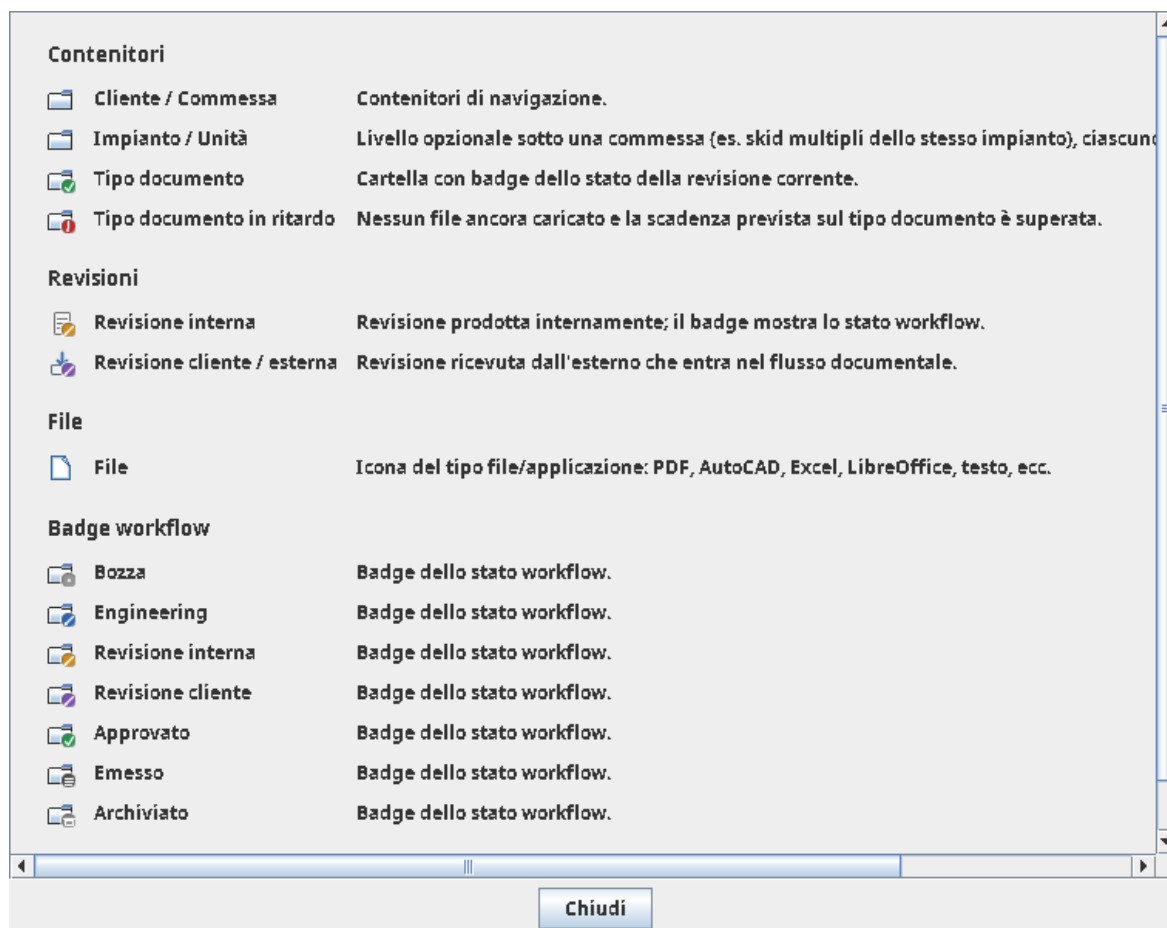
No step in this story requires leaving Arianna, and no step allows a previous version to be lost or who approved what to be forgotten. It's exactly the problem described at the opening of this manual (Part I): always knowing which revision is the correct one, and who is accountable for each.

# Appendices

Quick reference: icons, glossary, dependencies, frequently asked questions

## A Icon legend

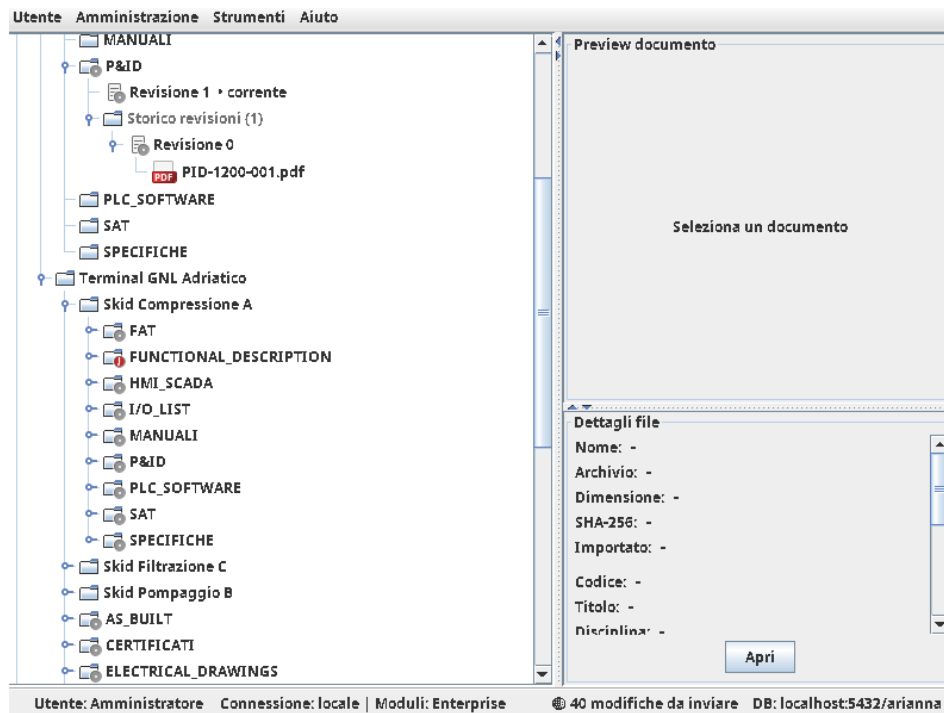
Tree icons carry information, not decoration: a colored badge on a document type's folder reflects the current revision's workflow status; a red badge with an exclamation mark flags a document type still missing a file past its expected upload date. Available at any time from Help → Icon legend...



**Fig. A.1** Document icon legend — containers, revisions, files, workflow badges

### A concrete example: an overdue document type

When a document type has been assigned an expected upload date (settable from the document type's context menu) and that date has passed with no file imported, the container's icon changes to flag it — the same signal that drives the "Document types overdue with no file loaded" count in Operational notifications (Section 32).



**Fig. A.2** Overdue document type — deadline badge on the tree

| Workflow status                   | Meaning                                                     |
|-----------------------------------|-------------------------------------------------------------|
| Draft (DRAFT)                     | First draft, not yet submitted for review.                  |
| Internal review (INTERNAL_REVIEW) | Under review by the internal team.                          |
| Client review (CUSTOMER_REVIEW)   | Under review by the client or a third party.                |
| Approved (APPROVED)               | Checked and approved, not yet formally issued.              |
| Issued (ISSUED)                   | Formally released for use on site/in the workshop.          |
| Archived (ARCHIVED)               | Superseded by a later revision, or closed at project end.   |
| Deadline overdue                  | Document type past its expected date, still without a file. |

## **B** Glossary

---

|                           |                                                                                                                                       |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <b>Project</b>            | A specific undertaking for a client; the top-level container for technical documentation.                                             |
| <b>Document container</b> | Operational synonym for "document type": the folder that collects the revisions of a document category.                               |
| <b>DEPENDS_ON</b>         | Link type: the current document depends on the selected one.                                                                          |
| <b>FEEDS</b>              | Standard relationship between types: the upstream document feeds the downstream one.                                                  |
| <b>IMPACTS</b>            | Standard relationship between types, or one-off link type: a change to the upstream document may require checking the downstream one. |
| <b>RELATED</b>            | Link type: a generic relationship between two documents, with no implied direction.                                                   |
| <b>Revision</b>           | The evolving state of a document type; earlier revisions stay in the history.                                                         |
| <b>SHA-256</b>            | A unique digital fingerprint calculated from a file's contents, used to detect duplicates and verify integrity.                       |
| <b>Skid</b>               | A self-contained plant/unit within a multi-plant project, with its own documentation.                                                 |
| <b>Revision history</b>   | The list of revisions prior to the current one, always available.                                                                     |
| <b>Document type</b>      | A category of technical document defined by the project (P&ID, SAT, etc.).                                                            |
| <b>Workflow</b>           | The cycle of statuses a document moves through from draft to archiving.                                                               |

## **c** Standard document dependency map

The map Arianna applies by default to every new project (Administration → Modules, "Type dependencies" module; editable from Tools → Document type dependencies...).

Definisci una sola volta la mappa standard degli impatti, a livello di tipologia documentale. Esempio: P&ID → Instrument List. La mappa viene poi riutilizzata in ogni commessa.

```
NARRATIVE → PROCESS_DESCRIPTION {FEEDS}
PROCESS_DESCRIPTION → PFD {FEEDS}
PFD → P&ID {FEEDS}
P&ID → EQUIPMENT_LIST {IMPACTS}
P&ID → LINE_LIST {IMPACTS}
P&ID → VALVE_LIST {IMPACTS}
P&ID → INSTRUMENT_LIST {IMPACTS}
P&ID → CAUSE_EFFECT {IMPACTS}
P&ID → FUNCTIONAL_DESCRIPTION {IMPACTS}
INSTRUMENT_LIST → I/O_LIST {FEEDS}
CAUSE_EFFECT → I/O_LIST {IMPACTS}
FUNCTIONAL_DESCRIPTION → PLC_SOFTWARE {IMPACTS}
I/O_LIST → PLC_SOFTWARE {FEEDS}
PLC_SOFTWARE → HMI_SCADA {IMPACTS}
PLC_SOFTWARE → FAT {FEEDS}
FAT → SAT {FEEDS}
SAT → AS_BUILT {FEEDS}
```

Aggiungi dipendenza... Rimuovi selezionata Chiudi

**Fig. C.1** Document type dependencies — standard, editable map

| From                   | To                     | Relationship |
|------------------------|------------------------|--------------|
| NARRATIVE              | PROCESS_DESCRIPTION    | FEEDS        |
| PROCESS_DESCRIPTION    | PFD                    | FEEDS        |
| PFD                    | P&ID                   | FEEDS        |
| P&ID                   | EQUIPMENT_LIST         | IMPACTS      |
| P&ID                   | LINE_LIST              | IMPACTS      |
| P&ID                   | VALVE_LIST             | IMPACTS      |
| P&ID                   | INSTRUMENT_LIST        | IMPACTS      |
| P&ID                   | CAUSE_EFFECT           | IMPACTS      |
| P&ID                   | FUNCTIONAL_DESCRIPTION | IMPACTS      |
| INSTRUMENT_LIST        | I/O_LIST               | FEEDS        |
| CAUSE_EFFECT           | I/O_LIST               | IMPACTS      |
| FUNCTIONAL_DESCRIPTION | PLC_SOFTWARE           | IMPACTS      |
| I/O_LIST               | PLC_SOFTWARE           | FEEDS        |
| PLC_SOFTWARE           | HMI_SCADA              | IMPACTS      |
| PLC_SOFTWARE           | FAT                    | FEEDS        |
| FAT                    | SAT                    | FEEDS        |
| SAT                    | AS_BUILT               | FEEDS        |

## **D** Frequently asked questions

---

### **Why can't I find a freshly imported file in Search everywhere?**

The search index considers file name, code, and title: if the metadata hasn't been filled in yet, try searching by file name alone.

### **Why is Check-in disabled in the context menu?**

Check-in is only active if the file is checked out, and only for the user who checked it out: check the status under File details → Check-out.

### **Can I undo a new revision created by mistake?**

The revision stays in the history and can't be hidden, but the most recent file can be replaced with a new check-in; for a formal correction, use Delete revision from the revision's context menu (a destructive operation that requires confirmation).

### **Why do Document history and Where used come up empty?**

These are views that fill in with use: the first logs openings over time, the second logs incoming links created by other documents. On a freshly imported file, they are correctly empty.

### **What's the difference between Project timeline and Project evolution report?**

The Timeline is a continuous chronological log of the whole project. The Report adds a quantitative summary up front and can be filtered by single document type: use it when you need a number, not just a list of events.

### **How do I limit what a user can see or change?**

Assign a role via Administration → Users... → Roles..., then fine-tune that role's specific permissions in Administration → Roles and permissions... (Section 34).

### **What's the difference between Add project and Create project...?**

Add project creates an empty project with no document containers at all: useful when the structure will be built by hand, type by type. Create project... opens the wizard that generates every chosen container in one go, with or without a template (Sections 09 and 21).

### **Can a file belong to more than one unit/skid at the same time?**

No: a file always belongs to a single combination of project → (unit) → document type → revision. To conceptually share information across units, use links (Section 17), not file duplication.

### **What happens if I try to check in a file someone else has already released in the meantime?**

It can't happen: only whoever performed the check-out can check the same file back in (Section 12). If another user tries anyway, Arianna rejects the operation with an explicit message.

**Do I need a continuous network connection to work?**

No: check-out creates a local working copy that can be edited offline; only the final check-in requires a database connection. In multi-site setups, the sync queue (Section 36) automatically recovers queued operations as soon as the network is back.

## E Application messages and confirmations

Arianna never silently performs an operation that touches data or files: every relevant action is accompanied by an explicit message. This table collects the most common ones, with the context in which they appear and what they mean.

| Message                                                                                              | When it appears                                               | Meaning / what to do                                                                                               |
|------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Check-in not allowed. Only the user who performed the check-out can upload the modified file.        | Check-in of a file checked out by another user                | Operation blocked by design (Section 3). Ask whoever holds the check-out to release it, or wait.                   |
| Modified file not found.                                                                             | Check-in with a file path that's no longer valid              | The file chosen for check-in has been moved or renamed on disk: reselect the correct path.                         |
| The selected file is identical to the previous revision. Do you still want to create a new revision? | New revision with a SHA-256 identical to the previous one     | Automatic anti-duplicate check (Section 14). Answer No if the change wasn't actually saved; Yes to proceed anyway. |
| To create a new revision you must state the revision reason.                                         | Check-in as a new revision without the Reason field filled in | Required field: the traceability of why a revision was created isn't optional (Section 4).                         |
| The file already exists. Do you want to overwrite it?                                                | Check-out to a local path that's already occupied             | Confirmation before overwriting a previous working copy at the same path.                                          |
| Remove the file? (DB + disk)                                                                         | Remove file from the context menu                             | Destructive and irreversible operation: deletes both the record and the physical file in the managed archive.      |
| Do you want to close the current session and log in as another user?                                 | Switch user / Logout...                                       | Confirmation before ending the session: any files checked out remain locked for the outgoing user.                 |
| Check-in complete.                                                                                   | End of a successful check-in, same revision or new revision   | Positive confirmation: the file has been deposited and the check-out released.                                     |
| Invalid revision number.                                                                             | New revision with a non-numeric or already-used number        | Correct the proposed number in the dedicated field of the check-in dialog.                                         |

### COMMON PRINCIPLE

Every confirmation that appears before a destructive or irreversible operation (removing a file, deleting a revision, deleting a project or client) requires an explicit response: there is no mode that skips these warnings, not even for the Administrator.

