

IntelliTrack™

User's Guide

Version 4.0.0

2026-08-20

Factory Data Systems, LLC

© 2026 Factory Data Systems, LLC. All rights reserved.

IntelliTrack User's Guide

Welcome to the IntelliTrack™ AI User's Guide. IntelliTrack is a web-based manufacturing track-and-trace and MES application from Factory Data Systems. This guide is task-oriented: each chapter explains what an area is for, how to get to it, and the exact steps to complete the work there.

How to use this guide

- Chapters are ordered by a typical user's journey — start with **Getting Started**, then the data model, then the daily production workflows, then periodic and administrative tasks. The last four chapters are the reporting data dictionary, security and sign-in, electronic signatures and the audit trail, and validation documentation.
- Each task lists the **role** it requires. The eight roles are **General** (any signed-in user), **Production** (line operators), **Production Supervisor** (everything Production does, plus releasing, correcting and authorizing), **Recipe, Engineering** (the plant and equipment model, OEE setup and alarm definitions), **Quality** (closing process audit records), **Audit**, and **Administrator** (everything Engineering and Quality do, plus administering the system); your administrator decides which directory groups or identity-provider roles give each one ([Chapter 19](#)). The menus show almost every page to everyone (**Check-Out** on the **Production** menu, and the **Inventory** menu with **Receiving** and **Shipping Verification**, only to Production users and above): if you open a page your roles do not allow, IntelliTrack names the roles it needs, and **My access** (under your name in the menu bar) shows the roles you hold and what grants them. Ask your administrator for a role you need.
- Where your site runs in **GxP Mode**, some tasks ask for an **electronic signature**; each such task says so, and [Chapter 20](#) explains how to sign.
- UI labels, buttons, and menu paths appear in **bold**; navigation paths read **Menu** → **Item**. Your screens may carry your company's logo and colours.

Table of contents

1. [Getting Started](#) — signing in and out, roles and **My access**, the menu bar and site, the home page and product search, the top menus, the in-app Help center, and your date and language preferences.
2. [Plant & Equipment Model](#) — the ISA-95 hierarchy (Enterprise → Site → Area → Workcell → Equipment), equipment settings and fault codes, processes, and process attributes.
3. [Products](#) — looking up products, status/history/genealogy, creating, editing, and rejecting, receiving inventory, and split/merge/consume.
4. [Materials & Suppliers](#) — materials and the shared approval lifecycle, suppliers, and units of measure.

5. [Recipes](#) — from recipe to machine; site namespaces; authoring and approving recipes; values, control instances, deviations and overrides; GTINs, ECO keys, and attribute maps.
6. [Bills of Material](#) — static and dynamic BOMs, their approval lifecycle, and how a BOM is checked against genealogy.
7. [Work Orders](#) — the work-order lifecycle, creating and editing orders, assigning and moving products, quantity enforcement, retention, and the work-order API.
8. [Production Execution](#) — the OK-to-Work gate, Check-Out, how machines run production, the queue, shipping verification, routing, and carriers.
9. [Product Audits](#) — the audit loop, the audit station, completing audits, skipping or postponing a due audit, the QA backlog, results, and configuration.
10. [Quarantine & Disposition](#) — raising and working quarantine incidents, product status (disposition) codes, and shipping verification.
11. [OEE](#) — OEE reports, the state timeline, alarms, event review, modes and states, and OEE configuration.
12. [Reports](#) — the shared report mechanics and the product, where-used (recall impact), performance and SPC, production, quality, inventory, supplier, carrier, material, curve, and WIP reports.
13. [Dashboards](#) — building, playing, archiving, and restoring report slideshows.
14. [AI Insights](#) — AI settings, connections and routing, running insights, retention, and the AI memory review gate.
15. [Notifications](#) — profiles and addresses, publications, subscriptions, notification channels, and the delivery log.
16. [ERP Integration](#) — connections, field mappings, transaction enablement, production confirmations, and diagnostics.
17. [System Configuration](#) — system parameters, the site date and time format, connection/SMTP setup, the three database synchronization jobs, units of measure, locations, approval statuses, branding, API clients, licensing, and Log Search (the error and warning logs).
18. [Reporting Database Data Dictionary](#) — reference: the read-only reporting views, including the carrier views, and the carrier-activity function (columns, data types, and descriptions) for building your own custom reports.
19. [Security and Sign-In](#) — the ways to sign in, sessions, roles and the role mapping, and the administrators' **Security** pages (settings, sign-in review, users and provider links, the security audit trail, and the connection test).
20. [Electronic Signatures & Audit Trail](#) — GxP Mode, signing a change, reading who signed a record, the inactivity lock, searching and reviewing the audit trail, tamper-evidence verification, and administering signature policies, reason codes, and readiness.
21. [Validation Documentation](#) — the validation library: finding, previewing and downloading tests (with values or a parameter set), the coverage register, generating a validation package from a URS (IQ, OQ, PQ, FAT and SAT protocols, the traceability matrix, and the optional FRS), document templates, library updates and their Change Notices, and the basis for the document content.

Reference

- [Glossary](#) — terms as they appear in the application.
- [Chapter outline](#) — the chapter plan, the map of every menu item and where it leads, and the role model.

This guide was generated from the IntelliTrack source code. Working/meta files for documentation maintainers (for example, `_feature-inventory.md`, `_open-questions.md`, `_build-report.md`) live in the sibling `UserGuideAuthoring` folder so they do not appear in the in-app Help viewer.

Chapter 1: Getting Started

Overview

IntelliTrack™ AI is a web-based manufacturing track-and-trace and MES application from Factory Data Systems. You use it from a browser to look up product status, run production, configure recipes and the plant model, and review reports. This chapter orients you: how you sign in and out, why what you see depends on your **role** and **site**, how to navigate, and where the in-app Help lives.

Why start here: two things shape your entire experience of IntelliTrack — the **site** you are connected to (which determines *whose* data you see) and your **role** (which determines *what you can do*). Understanding them up front explains why a colleague can open pages and use buttons you can't, and why a screen shows one plant's product and not another's.

Before You Begin

- On a computer on your company network, IntelliTrack usually **signs you in automatically** with your Windows sign-in — there is no separate IntelliTrack password. Your plant may also let you sign in with your Windows user name and password (from a computer that is not on the network), or through your organization's sign-in service; see [Sign in and sign out](#) below and [Chapter 19](#). Everything you do is recorded under your name, for traceability.
- What you can see and do depends on your **role**. IntelliTrack recognizes eight; your administrator decides which Windows groups or identity-provider roles give each one (on the **Security** → **Role mapping** page, Chapter 19), and you may hold several. Two of them include others: **Production Supervisor** includes everything **Production** may do, and **Administrator** passes every Production, Production Supervisor, Engineering and Quality check (with the one exception under *Release decisions* below). That inclusion is built into IntelliTrack, so a supervisor's group does not also have to be mapped to Production. The other roles do not include one another — **Engineering** and **Quality** include no production role, and no production role includes them: each opens only its own pages, which is why everyone who uses IntelliTrack normally also holds **General**.

Role	Who it is for	What it unlocks
General	Any signed-in user	Look-ups, reports, viewing — the read-only backbone — and marking a product as a reject.
Production	Line operators	Running the process as defined and recording what happened: Check-Out and check-in, shipping verification, receiving, downtime and alarm response, a product's description, location or carrier, placing product on Hold, containing suspect product, starting and stopping work orders.

Role	Who it is for	What it unlocks
Production Supervisor	Shift leads & supervisors	Everything Production does, plus releasing, correcting, overriding and authorizing: releasing product from Hold, changing a pass/fail result, renaming or correcting a product, scrapping, splitting, merging and consuming, managing work orders, skipping or postponing audits, dispositioning or rolling back quarantine, requesting recipe overrides.
Recipe	Process/quality engineers	Authoring and approving recipes, materials, and BOMs.
Engineering	Controls, process & manufacturing engineers	Maintaining the plant and equipment model: enterprises, sites, areas, work cells, equipment and locations; processes and routes; attributes and attribute maps; OEE modes, states and user-defined fields; equipment enforcement; each machine's alarm definitions.
Quality	Quality technicians	Recording the quality department's verification of production: closing a process audit record with its measurements and verdict (Chapter 9).
Audit	Quality reviewers	Reading the audit trail and deciding about its findings. Changes nothing.
Administrator	System administrators	Everything Engineering does, plus administration: security and role mapping, system parameters, licensing, branding, ERP and API connections, AI and notification setup, signature policy and the compliance pages, approval statuses, master data such as units of measure, suppliers, carriers and dispositions, and restoring deleted items — and every production, supervisor and quality task, with the one exception below.

Engineering is separate from Administrator so that the engineer who builds the plant model, commissions a machine's OEE settings and loads its alarm list does not also have to be handed security, licensing and signature policy. Quality is separate from the production roles because a process audit checks the line: the people who run it do not also close the audit of it.

Each task in this guide names the role it needs. The menus show almost every page to everyone; the exceptions are **Check-Out** on the **Production** menu and the whole **Inventory** menu (**Receiving** and **Shipping Verification**), shown only to people who hold Production, Production Supervisor or Administrator, since nobody else can use them. If you open a page your roles do not allow, IntelliTrack shows **Unauthorized** and names the roles that page needs (by their keys: *Admins* is the Administrator role), with a link to **My access**. Many pages also show their editing buttons only to the roles that may use them. If you need a role you do not hold, ask your administrator.

Release decisions. A release decision is an act that lets stopped or suspect product continue, or waives a quality check. There are exactly seven: changing a reject to a pass, or any other change of a pass/fail result except to reject; releasing product from Hold or another restrictive disposition; dispositioning quarantined product; rolling back a quarantine incident; and skipping a due process audit, or postponing it by time or by units. Production supervisors and administrators make them. When your site has **GxP Mode** on, each must be signed by someone who holds **Production Supervisor**: an administrator who does not also hold that role reaches the page but cannot sign the act. With GxP Mode off no signature is taken, and administrators make release decisions like any other supervisor task. Your site can change this in its signature policies (see [Who may sign: roles](#)).

Rejecting is open to everyone. Any signed-in user may mark a product as a reject, and no role ever restricts it: whoever spots a bad unit should be able to pull it out at once. It is not signed as shipped. The control sits on the way back — changing that reject to a pass is a release decision.

- Every change to configuration and quality records is kept in a permanent **audit trail**, and where your site has switched on **GxP Mode**, some changes ask for your password in an **Electronic signature** panel before they are saved. See [Electronic Signatures & Audit Trail](#).

How to Get Here

Open the application URL in your browser; once you are signed in you arrive on the **Home** page. The brand at the left of the menu bar — **IntelliTrack™ AI**, or your company's logo or name — always brings you back Home, with the navigation across the top.

Note: Your screens may not look exactly like this guide. An administrator can apply your company's branding on **Configuration** → **Additional Configuration** → **Branding**: the logo or company name in the menu bar and on the sign-in page, the browser-tab icon, the colours of the header, footer, buttons and links, and the contact details shown on **Home** → **Contact** and in the page footer. The *Powered by IntelliTrackAI* mark and the IntelliTrack copyright always remain.

Sign in and sign out

Signing in is how IntelliTrack knows who you are: the records you create or change carry your name, and the roles your sign-in brings decide what you may do.

- **At a site that signs people in with Windows**, a company computer is normally signed in with your Windows account the moment you open IntelliTrack — there is nothing to choose.
- **Otherwise** — after you have signed out, or at a site that also offers its organization's sign-in service — the **Sign in to IntelliTrack** page lists the ways your site offers: **Sign in with Windows**;

Sign in with a user name and password (your Windows user name and password, for a computer that is not on the company network — it works only at IntelliTrack's secure `https://` address); and **Sign in with your organization's sign-in service** (an identity provider, which may ask for extra verification). If that service is the only way in, you go straight to it.

- **If a sign-in is refused**, the page says why and who can fix it — for example *You do not have access to IntelliTrack* when none of your groups is mapped to an IntelliTrack role. See [Security and Sign-In](#).

To **sign out**, select **Sign out** under your name at the right of the menu bar. The **Signed out** page appears, and IntelliTrack does not sign anyone back in on that browser until someone chooses to sign in again (**Sign in again**) — so a shared station can change hands safely (close the browser when you leave a shared computer). If you signed in through your organization's sign-in service, you are signed out of it too, when the service supports that.

Your session also ends on its own — after a period without activity (15 minutes unless your administrator set another) and after a maximum length (12 hours unless set otherwise) — and you are asked to sign in again; on a company computer this usually happens without you noticing. Separately, if your administrator has turned on the **inactivity lock**, a browser left untouched for the set number of minutes shows **This browser is locked** until you enter your password again (see [The inactivity lock](#)). If you signed in through your organization's sign-in service, IntelliTrack cannot check a password for you, so the lock cannot be opened: close the browser, then open IntelliTrack and sign in again.

Read the menu bar — who you are and whose data you're in

The right side of the menu bar shows who is signed in and where:

- **Current User**: your name as IntelliTrack recorded it when you signed in (point at **Current User**: to see your account name). It is the name your records carry. Your name, underlined with dots, is a link: it opens **My Access**, what your roles allow; see below.
- **Inbox** — the bell button under your name: the notifications sent to you inside IntelliTrack. A red bubble on its corner shows how many you have not read yet, from 1 to 9, and **9+** for more than nine; with nothing unread there is no bubble. See [Check your notifications \(Inbox\)](#).
- **Sign out** — see [Sign in and sign out](#). When nobody is signed in, a **Sign in** link appears instead.
- **Site Name**: the configured site, e.g. *Site Name: Plant 1*. If it reads **Site Not Set**, the site has not been configured — an administrator binds this installation to its site with **Is Current Site** (see [Set the current site](#)).
- The language flags, beside **Sign out** — see [Dates, times, and your own preferences](#).

Why it matters: production, recipes, and reports are scoped to the current site, so the **Site Name** is your confirmation that you are working in the right plant's data before you act — and **Current User** confirms that the right person is signed in at a shared station.

Check what you can do — My Access

Your name in the menu bar opens **My Access**, which answers "why can't I open that page?". It shows how you are signed in (**Signed in as**, **User name**, **Signed in with**, and when your sign-in was **Checked**) and, for each IntelliTrack role — **Administrators**, **Recipe**, **Engineering**, **Production**, **Production Supervisor**, **Quality**, **Audit**, **General** — whether you hold it (**Allowed** or **Not allowed**) and which of your groups or identity-provider roles grants it (**Granted by**). The roles you do not hold are listed too, so you can tell your administrator exactly what is missing. The page opens for anyone, even someone who holds no role at all. It shows the roles your groups are mapped to: a production supervisor can see **Not allowed** beside Production and still do everything Production may, and an administrator can see **Not allowed** beside Engineering and still do every Engineering task, because those inclusions are built in rather than mapped. If you have just been added to a group in the directory (or given a role at your sign-in service), IntelliTrack sees it only when you sign in again: select **Sign in again** on this page.

Messages and notices at the top of a page

- **Status messages.** When you save, delete, or run something, the result appears in a coloured bar at the top of the page you land on: red when it failed, yellow for a warning (something was not done as asked, or needs your attention), green when it completed, and blue for information. Read it before moving on — a yellow message usually means part of what you asked for did not happen.
- **License notice.** When the license needs attention, a bar across the top of every page says so: a trial and its days remaining, a license that expires within 30 days, a license in its reactivation grace period, the licensed number of equipment modules reached, or no valid license at all (then only licensing administration is available). Tell your administrator (see [System Configuration](#)).
- **Unexpected errors.** If something goes wrong that IntelliTrack did not expect, the error page shows a **Reference ID**. Give it to your administrator: it ties your report to the error IntelliTrack logged.

Find your way around — the seven menus

The whole application is organized into seven top menus. **Click** a menu title to open it, then click an item to go there; pointing at a title does not open it, and the title itself goes nowhere. The pages the titles used to open are items in their menus — for example **Home** → **Home**, **Reports** → **Reports**, and **Configuration** → **Additional Configuration**, the hub for every setting that has no menu item of its own. In a window narrower than about 1,200 pixels (a tablet or a small laptop screen, say) the menus fold behind a menu button (≡) next to the brand.

The menus group the pages by the activity areas of the ISA-95 standard for manufacturing systems, which most manufacturing execution systems follow: **Production**, **Quality**, **Product Definition** (what is made, and from what) and **Inventory**, after **Home** and **Reports** and before **Configuration**. Within a menu the items run in the order the work happens — on **Production**, for example, from the work order to running the process, then tracking the product and reviewing the equipment events.

This map shows what each menu is for and the role that drives it:

The top menu — what each area is for

Home <i>orient & look up</i>	Reports <i>see what's happening</i>	Production <i>make & track product</i>	Quality <i>inspect & hold</i>	Product Definition <i>define what is made</i>	Inventory <i>goods in & out</i>	Configuration <i>set up the system</i>
Home Help About Contact	Reports · Product Report Performance · Quality Audit Results · OEE Inventory · Supplier Carrier Reports AI Insights · Dashboards	Work Orders Production · Check-Out Products OEE Events	Product Audits Quarantine	Materials Recipes Bills of Material	Receiving Shipping Verification	Equipment Model Processes Notifications → 6 notification pages API Clients · Licensing System Errors & Warnings Additional Configuration → Compliance (8 pages)
All users	General	Production	Quality · Production Supervisor	Recipe	Production	Engineering · Administrator

Tip: The role shown at the bottom of each column is the role that *drives* that area. Most pages are still **viewable** by any signed-in user — the role gates the actions, not always the looking. Everyone can open the **Equipment Model**, **Processes**, **Attributes** and **Locations** pages, but only Engineering and Administrators can change them. A few pages need a role just to open, for example **AI Insights** (Administrator), **Check-Out**, **Receiving** and **Shipping Verification** (Production, Production Supervisor or Administrator), the OEE configuration pages and **Equipment Enforcement** (Engineering or Administrator), **API Clients**, and the **Compliance** pages (Administrator).

Process **Attributes** and the other settings without a menu item of their own (units of measure, locations, carriers, suppliers, system parameters, security, branding, and more) are tiles on **Configuration** → **Additional Configuration**, sorted into groups: **Equipment & Processes**, **Materials & Logistics**, **Quality & Change Control**, **Connections & Notifications** and **Security & Administration**. Select a group to see its pages; only the pages your role can open are shown. Its **Compliance** tile opens a page with a tile for each of **Audit trail**, **Audit trail reviews**, **Tamper-evidence verification**, **Signature policies**, **GxP Mode**, **Reason codes**, **Signature attempts**, and **Signature readiness** — where administrators search and review the audit trail and set up electronic signatures (see [Electronic Signatures & Audit Trail](#)). Those eight pages used to be listed at the foot of the Configuration menu.

Configuration → **Notifications** works the same way: it opens a page with a tile for each of **Inbox**, **Subscriptions**, **Publications** and **User Profile**, and for administrators **Delivery Log** and **Notification Channels** (see [Notifications](#)).

Search for a product by serial number

The fastest traceability lookup is the home-page search box.

1. Select **Home** → **Home** (or the brand) to return to the home page.

2. In **Product Search**, type into **Enter the serial number here...** whatever identifies the unit: its serial number, an alias, its shipping code, or a serial number it used to have.
3. Click **Search Now** (or press Enter).

IntelliTrack tries the entry as a serial number or alias, then as a shipping code, then as a former serial number (so a renamed product still answers to its old serial), and opens that product's status, history, genealogy, and performance under its **current** serial number — the same destination as **Production** → **Products** (Chapter 3). Deleted products are not found; if nothing matches, a not-found page appears. Keep this in mind as your one-step answer to "what's the story on this unit?"

Use the Quick Reference links

Below the search box, the home page's **Quick Reference Links** offer shortcut tiles — **Product Status**, **Production**, **Reporting**, **Diagnostics** (the system error log), **Recipe Configuration**, and **System Configuration** (the Additional Configuration hub) — that jump to the most common areas. They are shortcuts only; everything is also on the top menu.

Open the Help center

This User's Guide ships inside the application.

1. Open **Home** → **Help**.
2. The **Help Resources** page lists **AOI Help** (PLC Add-On Instructions, for controls engineers), the **Security Matrix** (which kinds of task each role can do), **User Guides** — this guide — the **Reporting Database Data Dictionary** (for people who build their own reports on the Reporting database), and the **End User License Agreement (EULA)**.
3. In the **User Guides** section, the left **User Guide Navigation** tree lists the chapters; click one to read it in the right panel. Links from one chapter to another stay inside the Help center.
4. To keep or hand out a copy, select **Print / Save as PDF**. A clean, printable copy opens in a new tab together with your browser's print dialog; print it, or choose *Save as PDF* as the destination to make a PDF. (A PDF document is shown in place instead, and a Word or Excel document offers **Download Document**.)

Note: The guide is built from the files in the application's Help folder, so it travels with the installed product — there is no separate site to visit or version to keep in sync.

Help beside a field. Some fields that take a formula or a special format have a **Help** button beside them — for example **Serial Number Formula** on a material; **Shipping Verification Code Formula**, the minimum and maximum time since the last operation, and **Maximum Number of Reruns Allowed** on a process; and a system parameter's value. It opens a small help window about that field, so you can read the rules without leaving the form.

Dates, times, and your own preferences

- **Dates and times** on screens, reports and export headers look the same for everyone at the site. An administrator chooses the format from a list of presets (**Date and Time Format** on **Configuration** → **Additional Configuration**); it is not a personal setting and does not change with your language, so every operator records and prints dates the same way. A few places keep a fixed form — signature and audit-trail records always read year-month-day with the time zone. See [Date and time format](#).
- **Items per Page**, also on **Configuration** → **Additional Configuration**, is yours alone: how many rows most lists and reports show per page.
- **Language**. The flag at the right of the menu bar sets your display language, and IntelliTrack remembers your choice; only English is offered there today. Until you choose, IntelliTrack follows your browser's language where it has a translation — at present only the electronic-signature texts (the signature panel and the audit-trail and signature pages) are also available, in Spanish, Japanese, and Chinese; everything else is in English.

View product and contact information

- **Home** → **About** shows the **Application Version** that is running and, when the build carries one, its **Build Date** — quote both when you report an issue.
- **Home** → **Contact** shows your company's contact details when your administrator has entered them with the branding (see the note under [How to Get Here](#)); otherwise it shows Factory Data Systems'.

Field Reference

Home page **Product Search**:

Field	Required?	What it does / valid values
Serial number	Yes, to search	A serial number, alias, shipping code, or former serial number of the product to open.

Related Topics

- [Plant & Equipment Model](#)
- [Products](#)
- [Reports](#)
- [System Configuration](#)
- [Security and Sign-In](#)
- [Electronic Signatures & Audit Trail](#)
- [Glossary](#)

Chapter 2: Plant & Equipment Model

Overview

The equipment model is the structural backbone of IntelliTrack. It describes your operation as an ISA-95 (S95) hierarchy — **Enterprise** → **Site** → **Area** → **Workcell** → **Equipment** — and almost everything else attaches to it: products are processed *at equipment*, recipes are instantiated *to equipment*, routing runs *between equipment*, OEE is measured *per equipment*, and reports roll up *by the hierarchy*. This chapter covers building and maintaining that hierarchy, plus the two related building blocks: **Processes** and **Process Attributes**.

Why this matters: get the model right and every later feature has somewhere to hang. Equipment isn't just a name in a tree — each piece carries settings that **directly drive the gates and reports** elsewhere in the system. And because machines on the floor identify themselves to IntelliTrack **by their equipment name**, the model is also the contract between the software and the PLCs: a name mismatch means a machine can't find itself.

Before You Begin

- **Viewing** the model is available to any signed-in user.
- **Creating, editing, and deleting** model items, processes, and attributes requires the **Engineering** or **Administrator** role, and so does the rest of a machine's engineering setup: its routing, attribute map, OEE settings, enforcement and alarm definitions. Engineering exists so that the controls and process engineers who build the model can do it without being made administrators, which would also hand them security, role mapping, system parameters, licensing and signature policy. Which directory groups or identity-provider roles grant each role is set by an administrator — see [Security and Sign-In](#).
- Build top-down: an Enterprise before a Site, a Site before an Area, and so on.
- Equipment links to a **Process**, so define your processes alongside (or before) the equipment that performs them.
- With **GxP Mode** on, creating, editing, or deleting a **process attribute** is electronically signed by default (see [Manage Process Attributes](#)). Nothing else in this chapter asks for a signature.

How to Get Here

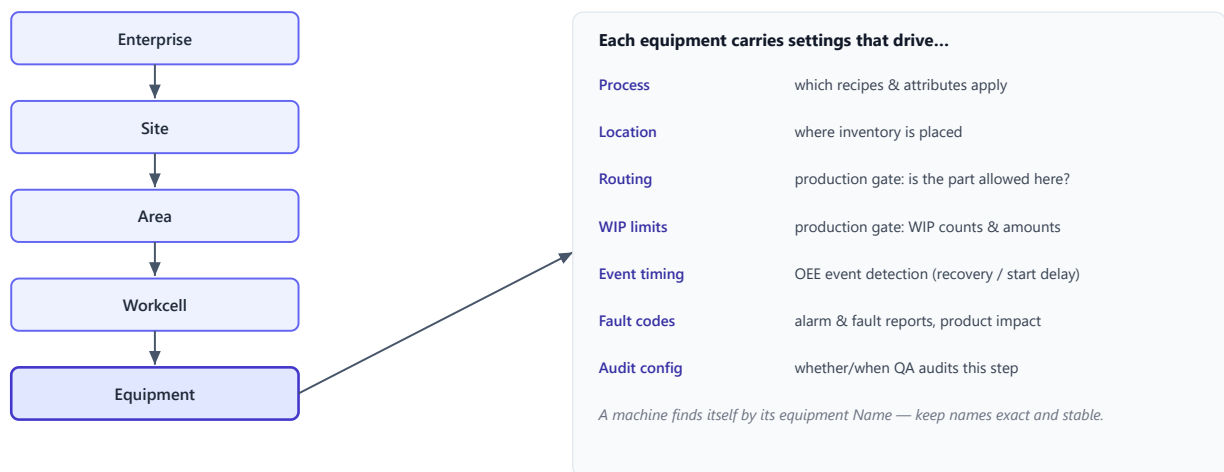
Configuration → **Equipment Model** opens the equipment-model page (heading **Equipment Configuration**). Processes are at **Configuration** → **Processes**. Process attributes have no menu item of their own: open a process's **Attributes** link on the Processes list, or go to **Configuration** → **Additional Configuration** → **Equipment & Processes** → **Attributes** (page **Processes and Attributes**).

The model — and what each equipment setting drives

The page has two panels: the **Equipment Model Definition** tree (expand Enterprise → Site → Area → Workcell → Equipment) and a panel describing the node you select — its parent (for equipment, its workcell and process), who last changed it and when, and links to **Details** and **Children**. Engineering users and administrators also see **Edit**, **Delete**, and the link that creates the next level down. A **Navigation Path** above the tree shows where you are. Selecting a Site shows **Current Site: Yes** for the site this installation runs — see [Set the current site](#).

The diagram shows the hierarchy and — the part that matters operationally — what each equipment setting feeds:

Equipment model — and what each equipment setting drives



Add a level to the hierarchy

The five levels are managed the same way; each lower level is created under its parent.

1. Select the parent node in the tree (for a new Site, its Enterprise; for a new Area, its Site; and so on).
2. Select **Create Site**, **Create Area**, **Create Workcell**, or **Create Equipment** in the panel (shown only to Engineering users and administrators). A new Enterprise needs no parent and has no link in the tree: open `/Enterprises/Create` directly.
3. Fill in at least the **Name**, plus a **Description** and the level's code if helpful. The codes (**Enterprise Code**, **Site Code**, **Area Code**, **Work Cell Code**, **Equipment Code**) are what serial-number and shipping-code formulas insert for their plant tokens — `<N...N>`, `<S...S>`, `<A...A>`, `<W...W>` and `<E...E>` — so a serial can say where the part was made.
4. Select **Create New** — you return to the tree, opened at the new item.

Note: Names must be **unique** among live items: Enterprise, Process, and Equipment names across the whole system (equipment at every site included); Site, Area, and Workcell names within their parent. For equipment this is functional, not tidiness: machines identify themselves by equipment name, so a duplicate or a rename has real downstream effects — a machine still sending an old name is refused as an unknown equipment module. Only a new equipment's name is checked on the form (**Equipment name already exists in the database**). Any other duplicate, including renaming equipment to a name already in use, is refused by the database: you land on the general error page and nothing is saved.

Edit, view, or delete a level

- **Details** views a node read-only; **Edit** (Engineering or Administrators) changes its name, description, code, or parent — moving an Area to another Site, for example, takes everything beneath it along.
- **Delete** (Engineering or Administrators) is a soft delete that **cascades**: deleting a Site retires its Areas, their Workcells, and all their Equipment, and deleting an Area or a Workcell does the same for what lies beneath it. Deleting a piece of equipment on its own also permanently removes the routing paths that start or end at it. Deleted items are flagged and hidden, not physically removed, and each is renamed with a `-DEL:` suffix and the date and time of the deletion, which frees its name for a new item. An administrator can bring it back — see [Restore a deleted item](#).
- An **Enterprise** cannot currently be deleted from the web: its Delete page opens, but confirming it ends on the general error page and nothing is deleted. To retire what lies beneath an Enterprise, delete its Sites.

Warning: Because delete cascades down the tree, removing a high-level node quietly retires everything beneath it. Confirm the scope before deleting.

Restore a deleted item (Administrators)

Deleted enterprises, sites, areas, work cells, equipment, locations, processes and attributes stay in the database, hidden from the tree and from every list production uses. An administrator sees them by ticking **Show deleted** under the **Plant Model** heading: the tree then shows deleted items too, greyed, struck through and marked **Deleted**, and the address keeps the choice (`?includeDeleted=true`), so a reload or a shared link opens the same tree. Nobody else sees the box, and asking for deleted items in the address does nothing for them. Engineering users delete plant items but cannot restore them; every restore needs the **Administrator** role.

Select a deleted item and its card shows when and by whom it was deleted, with **Details** and **Restore** in place of **Edit**, **Delete** and **Create**. Its **Details** page is read-only under a red banner that says the same and carries **Restore**. Locations, processes and attributes have their own lists, and each has the same **Show deleted** box (an attribute's is on its process's list).

Only the item you choose comes back. Deleting a site, area, work cell or process takes what lies beneath it in the same act; restoring it does not bring those back. Open the restored item in the tree — you return there after the restore, with deleted items still shown — find what is still marked **Deleted** inside it, and restore each on its own. Work from the top down: an item whose parent is still deleted cannot come back first.

The restore page shows what will happen before anything changes:

- **Comes back as** — the item's name before it was deleted, taken from the table's history, so the **DEL:** suffix goes and a name the suffix shortened comes back whole. An item deleted before IntelliTrack kept that history comes back with the suffix removed; if its name may have been shortened, the page says so — rename it afterwards if it is incomplete.
- **It cannot be restored yet** — the page refuses, and says why:
 - it belongs to an item that is also deleted — a work cell whose area is deleted, equipment whose process is deleted, a location whose place is deleted, an attribute whose process is deleted. It would come back where nothing can reach it, so the page names each deleted item it belongs to, outermost first, and links to its restore: restore those first;
 - a live item now uses the name it would come back under (in the same parent for a site, area or work cell; anywhere for equipment, locations and processes): the page names that item and links to it. Rename or delete one of them first; nothing is renamed for you;
 - restoring equipment would take its site over its licensed number of equipment modules, or an attribute was superseded while its replacement is in use.
- **Before you restore it** — what the delete removed outright: equipment deleted on its own lost its route paths and its active, allowed and disallowed materials; a process lost its route paths and materials; an attribute lost its maps to equipment; a location's equipment defaults and the products standing at it were cleared. Set those up again after restoring.

Restore brings the item back — or, if it has changed since the page was opened, nothing — and the audit trail records it as a change, by you. A restored site does not become the current site: that stays as it is ([Set the current site](#)).

The other deletable configuration — materials, recipes, bills of materials, suppliers, dispositions, units of measure, unit dimensions, carriers and ECO keys — works the same way from its own list: **Show deleted**, then **Details** or **Restore** on the marked row. A recipe or bill of materials deleted while **Active** comes back **Approved**, and goes live again only through the normal approval transition. A bill of materials comes back only while its material has no other live one, and restoring a disposition is signed whenever deleting one is.

Restoring an **attribute** is a signed act when your site signs attribute deletes: with GxP Mode on it is signed by default, like the delete it reverses ([Signatures and what a change removes](#)).

Set the current site

Each IntelliTrack installation runs one site's production, and **Is Current Site** on the Site form is how an administrator says which one. Tick it on the site's **Edit** (or **Create**) page and save. It needs the **Administrator** role: Engineering users maintain sites, but choosing which one this installation serves is a system setting, so for them the box shows the current binding read-only, with *Only an administrator can change which site this installation is bound to*. An engineer who renames the current site does not change the binding; the menu's **Site Name** follows the new name.

Why it matters: saving it **binds this installation's database to the site** (System Parameter 1, **Site Id**), and several things depend on that binding:

- **Production.** Machine calls are checked against it: equipment that belongs to another site is refused, and while no site is bound all production is refused with *This database is not bound to a site. Designate the current site on the Site edit page before running production*.
- **Electronic signatures** record the site they were given at, so none can be given while the installation is unbound (see [Electronic Signatures & Audit Trail](#)).
- **Quarantine** searches are scoped to it ([Quarantine & Disposition](#)).

Saving it also writes the site into the application's settings, which is what the **Site Name** in the menu bar and the tree's **Current Site** read.

The database is written first, because production depends on it. If only that half succeeds, the site is bound and production runs, but a warning at the top of the page names the site and says the settings file could not be updated: the menu keeps showing the previous site until the file is corrected and the current site is set again. If this database has never received a configuration sync, there is no **Site Id** to bind yet: the site is saved, a warning says production is still blocked, and you set the current site again after the sync. Clearing the box does not unbind anything — to move the binding, tick it on the other site.

Configure Equipment — and why each field matters

Equipment carries the most configuration because it is where production actually happens. Beyond **Name**, **Description**, and **Equipment Code**:

- **Work Cell** — its parent in the hierarchy.
- **Process** — the step it performs. This is what decides which **recipes** and **attributes** apply (Chapters 5 and 2), so it is the most consequential setting.
- **Default Location** — where a product checked in at this equipment is placed, unless the machine names another location. Left at -- **None** --, the equipment itself serves as the product's location. Locations are maintained under [Inventory locations](#).
- **Allow Machine to Request Control Instances** — on by default. When on, a machine that asks for a recipe with no control instance ready gets one created for it; when off, that request is refused

until someone creates and releases a control instance in IntelliTrack ([Recipes](#)). Turn it off where every recipe must be reviewed before a machine may run it.

- **Ideal Cycle Time (sec.)** — the designed seconds per unit that OEE **Performance** is measured against. A machine left at 0 cannot be scored: the OEE Scorecard lists it among the machines it did not score, and a machine that asks IntelliTrack for its own OEE for a lot is refused ([OEE](#)).
- **OEE Event Start Delay** and **OEE Event Recovery Time** (HH:MM:SS; new equipment starts at 00:02:00 and 00:06:00) — how long a machine must be in a tracked state before an event starts, which filters out nuisance blips, and how long it must be running again before the event is over. Together they decide how raw machine activity is grouped into **OEE events** (Chapter 11). Both are required, typed in full, up to 999:59:59. A blank is refused, and so is shorthand such as 90 or 5:30 — minutes or hours? — rather than saved as a guess that would regroup every event; the save writes nothing until both read correctly, so type 00:01:30 or 00:05:30. A change reaches OEE event logging after the next synchronization run, not the moment you save.
- **Maximum Number of Attributes Allowed** — how many positions the equipment's **Attribute Map** offers; match it to the PLC's data structure. New equipment always starts at 50 (a different value typed on the Create page is ignored), so change it by editing the equipment.
- **WIP Limits** — **Max WIP count**, **Max reject count**, **Max WIP amount**, and **Max reject amount**. While the equipment is at or over any limit that is set, check-out there is refused, and so is a machine adding a new product there (the WIP gate, Chapter 8); 0 switches that check off. The limits are saved when you **edit** the equipment — values typed on the Create page are not kept, so set them after creating it.
- **Routing** — the paths into this equipment and the materials each allows. This is the first thing the production gate checks (Chapter 8): no valid route, no work. The equipment's **Details** page lists them under **Process Routings**; they are created and edited on the routing pages ([Set up routing](#)).
- **Alarm definitions** — what each alarm code the machine raises means, which decodes its alarms and faults for OEE reporting and the product record. They have their own pages, reached from the equipment's Details or Edit page (see [Manage alarm definitions](#)).

Per-machine enforcement of work orders, recipes, and quantities is not on this form: it is set on the **Equipment Enforcement** page (Engineering or Administrator role, even to open it) — see [Work Orders](#).

Note: Creating equipment counts against your site's licensed equipment modules. When the site has used them all (outside a trial), **Create New** saves nothing and opens the Licensing page with *Equipment module limit reached. In use: ..., licensed: ...*

What an equipment's Details page adds

Beyond its settings, an equipment's **Details** page shows how it stands now:

- **Last Heartbeat** — when the machine last reported in (*No heartbeat has been received from this equipment.* if it never has), with a **Communication Anomalies** link listing the last 100 outages

recorded for it (**Outage Start, Recovered, Duration (sec.), Detected**). A gap between two heartbeats longer than **Communication Outage Threshold** (System Parameter 81, default 30 seconds) is recorded as an outage. The list is read from the central reporting database, so a new outage appears after the next synchronization. This is how you tell a machine that is idle from one that has lost its connection.

- **Audit Status** (also on the Edit page) — where the equipment stands in its process's audit schedule: next audit needed, last completed audit, units since, and overrides used, with links to the audit configuration and this equipment's **Audit History** ([Product Audits](#)).
- **Process Routings** — the routes into this equipment.
- **Attribute Map** (a link, also on the Edit page) — opens the equipment's PLC attribute positions in a new tab ([Recipes](#)).

Manage alarm definitions

A machine's PLC reports alarms as numbers. Its **alarm definitions** say what each number means: what the alarm says, how severe it is, what kind of alarm it is, whether it belongs on the record of the product being worked, and what the operator should do about it. Every alarm report, the alarm board, alarm notifications, and the product record read them — an alarm whose code has no definition shows as ***** Description Not Found**. What categories, product impact, and the rationalisation fields mean for alarms, and where they show up, is explained in [OEE](#).

Open them from the equipment's **Details** or **Edit** page: **Alarm Definitions** → *View, add, import and export the alarms this equipment raises*. Anyone can view and export them; adding, editing, retiring, reinstating, and importing need the **Engineering** or **Administrator** role — so the engineer commissioning a machine can load and maintain its alarm list without being made an administrator.

Note: The definitions are kept in the central reporting database, which a new machine reaches at the next synchronization. Until it has, the page says *This equipment has not reached the reporting database yet* and offers no changes — which is also why there are no alarm definitions on the Create page.

The list. The page lists the machine's live definitions by code, with its severity, category, product impact, and whether it has been **Rationalised** (somebody has recorded a cause, consequence, operator action, or response time). The search box matches the code, the description, and the rationalisation text, so *rear guard* finds the alarm whose operator action mentions the rear guard. Tick **Show retired** to include definitions taken out of service.

What each alarm this machine raises means: how severe it is, what kind it is, whether it reaches the product record, and what the operator should do about it. Alarm reports and the product record read these definitions.

7 live, 1 retired

[Import](#) | [Export \(Excel\)](#) | [Export \(CSV\)](#)

[Create New](#)

☐ Show retired

Code	Description	Severity	Category	Product impact	Rationalised	Status	Action
1001	Emergency stop pressed	Error	Safety	None	Yes	Live	Details Edit Retire
1002	Light curtain interrupted	Error	Safety	None	No	Live	Details Edit Retire
2010	Infeed jam at the chute	Warning	Equipment	Record	Yes	Live	Details Edit Retire
2011	Low air pressure	Warning	Equipment	None	Yes	Live	Details Edit Retire
3050	Seal temperature low	Error	Process	Record and hold	Yes	Live	Details Edit Retire
4001	Vision reject rate high	Warning	Quality	Record	No	Live	Details Edit Retire
5002	Label roll empty	Information	Material	None	Yes	Live	Details Edit Retire

[Back to the equipment](#)

A machine's alarm definitions. **Import** and the two exports are at the top right of the list.

Add or edit one. **Create New** adds a definition; **Edit** changes one. The code is the definition's identity — alarms are recorded against it — so it cannot be changed once saved: a code the PLC programmer renumbered is a new definition, and the old one is retired. Blank rationalisation fields mean *nobody has said*. If somebody else saved the same definition after you opened it, your save is refused with *Somebody else changed this definition after you opened it* rather than silently undoing theirs.

Retire or reinstate one. Definitions are **retired, never deleted**: an alarm already recorded against a code keeps saying what it meant, which a deleted definition could not. **Retire** takes the code out of service (the page tells you how many recorded alarms keep its description); a new definition can then be added for the same code. With **Show retired** ticked, a retired definition's row is marked **Deleted** and offers **Details** and, to those who may edit definitions, **Reinstate** in place of **Edit** and **Retire**; its **Details** page offers **Reinstate** too. Reinstate works as long as no other definition of that code is live.

Details. Everything the definition says, how many alarms have been recorded against the code on this machine and when the last one was, and its **Change History**: every change to what the code has meant on this machine, newest first — who made it, when, and why — including earlier definitions since retired.

Warning: Product impact decides whether an alarm is written onto the record of every lot worked while it stood (1, **Record**) and whether that product is also held (2, **Record and hold**). So any change that alters it — editing it, retiring or reinstating a code that carries it, or importing over it — needs a **Reason for the change**, and is refused without one. The reason is kept in the change history and attributed to you; it is not an electronic signature, whichever role you hold.

Export and import a machine's alarm list

A machine's alarm list usually arrives as a spreadsheet from the machine builder or the PLC programmer, and machines of the same make share one. Export and import let you keep that list in a spreadsheet: document it once, and load it into every machine of that make.

- **Export (Excel)** downloads the list as an `.xlsx` workbook: the definitions on the first sheet, in import format, and a **Reference** sheet saying what the severity, product impact, and category numbers mean. **Export (CSV)** downloads the same columns as a CSV. With **Show retired** ticked, retired codes are included (never beside a live definition of the same code).
- **Import** loads a workbook (`.xlsx` , first sheet) or CSV, one alarm per row, in this column order — only the first three are required, and columns may be left off the right-hand end:

Column	What it holds
Code	A whole number, once per file.
Description	Up to 200 characters.
Severity	0 Information, 1 Warning, 2 Error.
Category	A category's number or its name (shipped: 0 Unclassified, 1 Safety, 2 Equipment, 3 Process, 4 Quality, 5 Material). Blank is Unclassified.
ProductImpact	0 not on the product record, 1 on the record, 2 on the record and holds product. Blank is 0.
Cause , Consequence , OperatorAction	The rationalisation, up to 500 characters each.
ResponseSeconds	How long the operator has, in whole seconds greater than zero.
ResponsibleGroup	The maintenance group that owns the fix, up to 50 characters.

A header row starting `Code, Description, Severity` is skipped. Older three- and nine-column files still import.

To import:

1. On the list, choose **Import**, and choose the file.
2. Under **Codes already defined but missing from the file**, choose **Keep codes that are not in the file** (add and update only) or **Retire codes that are not in the file** — use Retire when the file is the machine's complete list.
3. Enter a **Reason for the change** if the import may change product impact.
4. Choose **Preview**. Nothing is saved yet. Every row is checked first: a bad row is reported by its row number (*Row 4: code 101 appears more than once in the file.*) and nothing can be imported until the file is fixed. Otherwise the preview lists what the import would add, change (and which fields), and retire, and names every code whose product impact would change.

5. Choose **Import**. The import is worked out again at that moment, against the definitions as they are then, and written all at once — or, if a required reason is still missing, not at all.

What importing press-line-alarms.csv would do. Nothing has been saved yet.

1 to add, 2 to change, 1 to retire, 4 unchanged

These codes would change whether they reach the product record, so a reason is required: 4001.

Code	Change	Description	Severity	Category	Product impact	Fields changed
1002	Changed	Light curtain interrupted	Error	Safety	None	Operator action, Response time (seconds)
4001	Changed	Vision reject rate high	Error	Quality	Record → Record and hold	Severity, Product impact
5002	Retired	Label roll empty	Information	Material	None	
6100	Added	Door interlock open	Warning	Safety	None	

Codes already defined but missing from the file

☐ Keep codes that are not in the file

☒ Retire codes that are not in the file

Use this when the file is the machine's complete alarm list. Retired codes are kept, so alarms already recorded against them keep their description.

Reason for the change

Kept in the history of every definition this import changes. Required when the import changes whether any code reaches the product record, including by retiring or reinstating one; the preview names those codes.

The import is worked out again when you confirm it, against the definitions as they are then.

Import

[Upload a different file](#) | [Back To List](#)

An import preview with **Retire codes that are not in the file** chosen. Code 4001 would move to **Record and hold**, so the import needs a reason before it can be confirmed.

How an import changes the list:

- A code that is already live is updated only where the file differs; an identical row changes nothing.
- **Severity**, **Category**, and **ProductImpact** are always taken from the file. A blank value — or a file without those columns — sets them to 0, so importing an old three-column file makes every code it lists **Unclassified** and takes it off the product record.
- The rationalisation, **ResponseSeconds**, and **ResponsibleGroup** columns are only ever filled in: a blank value, or a file without those columns, leaves what is already written. Clear one on the definition's **Edit** page.
- Retired definitions are kept, so alarms already recorded against them keep their description. A retired code that appears in a later import comes back as a new live definition.

Manage Processes

A **Process** is a named manufacturing step. Equipment is assigned to a process, and attributes and recipes are defined against processes — so a process is the hub that connects equipment, what gets measured, and what controls it. It also carries the step's own production rules — how long a part must have waited since its previous operation, how many times it may be run here, whether this is the final operation that issues the shipping code — which production applies at every piece of equipment assigned to the process (Chapter 8). The [Field Reference](#) lists each setting and its effect.

1. **Configuration** → **Processes** lists the processes with their **Is Final Operation** and **Is Rework Operation** flags; search by name or description (paged). Each row links to **Details** (the settings and the **Process Routings** into the process) and **Attributes**, and, for Engineering users and administrators, to **Edit** and **Delete**.
2. Engineering users and administrators add a process with **Create New**.

Warning: A process that still has equipment assigned cannot be deleted: confirming the delete ends on the general error page and nothing changes. Move that equipment to another process, or delete it, first. Once no live equipment uses the process, deleting it also retires its attributes and permanently removes every route to or from it.

Manage Process Attributes

Process Attributes define the data points captured against a process. They are why a recipe has anything to set limits on and why production has anything to measure.

1. Open a process's **Attributes** link on **Configuration** → **Processes**, or go to **Configuration** → **Additional Configuration** → **Equipment & Processes** → **Attributes** (page **Processes and Attributes**), pick a process — or **Global** — and select **Get Started** ».
2. The list shows that process's attributes together with the **Global** ones — attributes with no process, which every process can use. Engineering users and administrators add an attribute with **Create New**, and **Edit** or **Delete** one from its row. Global attributes are edited from the Global list, and the reserved system attributes (numbered below 1000, such as the GTIN and expiry-date attributes) cannot be edited or deleted.
3. Attributes have a data type; numeric attributes can be used everywhere (including SPC), while non-numeric types (boolean, string, curve) are excluded where only numbers make sense. An attribute can be flagged **Use for Product Evaluation** (it participates in pass/fail) and **Audit Point** (QA re-measures it — Chapter 9).

Use for Product Evaluation and **Required for Check-In** are separate switches, and both are on by default:

- **Required for Check-In** decides whether a value must be *collected* at check-in. **Use for Product Evaluation** decides whether the collected value is *graded* against recipe limits. A value is collected when the attribute is required **or** used for evaluation, so an attribute that is required but not evaluated is still recorded and still reaches your reports — it simply is not judged. That combination is how a plant logs data while something else, such as the PLC, decides pass and fail. See [Running without recipes](#) in Chapter 5.
- Both default to **on** deliberately. An attribute nobody has got around to configuring is evaluated; you have to switch grading **off** as a conscious act. An oversight should never be what stops a characteristic being graded.
- An evaluated value must always be collected, so turning **Use for Product Evaluation** on also turns **Required for Check-In** on and locks it. Turning evaluation off unlocks it and switches **Automatically Scrap on Reject** off: only an out-of-limit measurement can scrap a part — a missing required value rejects the check-in without scrapping it.

Both flags belong to the attribute, not to the equipment mapping. Turning **Use for Product Evaluation** off applies to *every* module that attribute is mapped to. If one module must grade a characteristic while another only logs it, define two attributes rather than turning evaluation off for both.

A running control instance keeps the attributes it was built with. When a control instance is created, each attribute mapped to the module (and each one the recipe holds a value for) is frozen into its value snapshot: **Use for Product Evaluation**, **Required for Check-In**, **Allow Skip** and **Automatically Scrap on Reject**, together with the limits. Check-in judges only what that snapshot holds. So a newly created attribute, an attribute newly added to a module's **Attribute Map**, or a change to one of these flags takes effect with that equipment's **next** control instance, or sooner through an approved override that adds the attribute ([Chapter 5](#)). Until then a value the machine sends for a new attribute is recorded but **not evaluated**: it is never required, it never fails or scraps a part, and a skip request for it is ignored. Each such value is logged under **System Errors & Warnings** with the part's serial number, marked **\$** on the Product Report ([Chapter 12](#)), and raises the **Attribute Recorded Outside Control Instance** notification once per control instance and attribute ([Chapter 15](#)).

Data types, units, and curves

The **Data Type** is **Boolean**, **Numeric**, **String**, or **Curve**. It decides how a value is recorded and graded, and which features can use it — SPC, for instance, offers only numeric attributes. The editor has the same fields for every type: an attribute carries no limits of its own, because its targets and limits are recipe values ([Recipes](#)).

Units records what the reading is measured in. Every unit of measure is offered, including those classified as measurement only ([Units of measure](#)); for a Curve the field is labelled **Y-Axis Units**.

A **Curve** attribute records a whole X-Y trace. Saving one creates a set of numeric attributes derived from it, each named after the curve: **Max Y Value**, **Min Y Value**, **X at Max Y**, **X at Min Y**, **Average Y Value**, **Area Under Curve**, **Max Positive Slope**, **Max Negative Slope**, **First Threshold Crossing**, **Last Threshold Crossing**, **Width Above Threshold**, **Knee**, **Elbow**, and **Inflection Point**. When a machine logs the curve, IntelliTrack calculates the derived metrics that the machine's active recipe includes, records each as an ordinary reading, and grades it against that recipe's limits — which is how you put pass/fail limits on the shape of a curve. The derived attributes follow the curve: they are rebuilt when it is saved and deleted with it, and **Recreate Curve Metric Attributes** on the curve's **Details** page (Engineering or Administrators) rebuilds and re-enables them.

Signatures and what a change removes

An attribute is the schema of the quality record, so with **GxP Mode** on, creating, editing, and deleting one is **electronically signed by default** — meaning **Approved**, reason optional — unless your site's signature policy says otherwise ([Which acts can be signed](#)). The form then shows an **Electronic signature** panel above its button, and what you sign is the whole definition. See [When you are asked to sign](#). An attribute's **Details** page lists its signatures under **Electronic signatures**, and every change, signed or not, is recorded field by field in the audit trail. As shipped, anyone who may change an attribute may sign the change; a site that wants engineering changes signed by an engineer can name the **Engineering** role as the signer in these acts' policies ([Who may sign: roles](#)).

- **Moving an attribute to another process, or changing its data type**, removes every recipe value and attribute-map position built on it, in the same save — they no longer fit. When the change is signed, the signed record notes how many were removed.
- **Deleting an attribute** removes it from every equipment's attribute map; a curve takes its derived metric attributes with it. When the delete is signed, it is signed against the definition shown on the page: if someone changes the attribute after you opened it, the delete is refused — reload and review it.

Field Reference

Common model item (Enterprise/Site/Area/Workcell):

Field	Required?	What it does / valid values
Name	Yes	Display name. Unique among live Enterprises; unique within the parent for Sites, Areas, and Workcells.
Description	No	Free text, shown in the tree and detail panel.
Enterprise Code / Site Code / Area Code / Work Cell Code	No	Filled into the <N...N> , <S...S> , <A...A> , and <W...W> tokens of serial-number and shipping-code formulas.
Parent (Enterprise , Site , or Area)	Yes (Site, Area, Workcell)	Where the item sits; changing it moves the item and everything beneath it.

Field	Required?	What it does / valid values
Is Current Site	No (Site only)	Binds this installation to the site; Administrator role only — see Set the current site .

Equipment-specific:

Field	Required?	What it does / valid values
Name	Yes	Unique across every site; machines identify themselves by it.
Work Cell	Yes	Parent workcell.
Process	Yes	Drives applicable recipes and attributes.
Default Location	No	Where products checked in here are placed; -- None -- = the equipment itself.
Equipment Code	No	Filled into the <E...E> token of serial-number and shipping-code formulas.
Allow Machine to Request Control Instances	No (on by default)	Off = a user must create and release a control instance before the machine can retrieve it.
Ideal Cycle Time (sec.)	For OEE	The Performance baseline; 0 = the machine cannot be scored.
OEE Event Start Delay / OEE Event Recovery Time	Yes	HH:MM:SS in full, up to 999:59:59; 00:02:00 and 00:06:00 on new equipment. A blank or shorthand (90 , 5:30) is refused and nothing is saved. Group machine activity into OEE events.
Maximum Number of Attributes Allowed	No	Attribute Map positions; 50 on creation, changed on Edit.
Max WIP count / Max reject count / Max WIP amount / Max reject amount	No	WIP gate limits; 0 = that check off. Saved on Edit only.

Process:

Field	Required?	What it does / valid values
Name	Yes	Unique among live processes.
Description	No	Free text.
Genealogy Preservation on Rerun	Yes (default Do Nothing: Leave Genealogy Records Untouched)	What happens to a re-run product's genealogy when it is checked out here again: leave it, or delete its first-generation parent record, all its first-generation relatives, all its parent records (every generation), or all its genealogy records (every generation).

Field	Required?	What it does / valid values
Shipping Verification Code Formula	No	Pattern for the shipping code issued at this process, using the serial-number formula codes (Help lists them). Blank = a random code whose length and characters are set by System Parameters 51 and 52. On a new process, available once Is Final Operation is ticked.
Minimum Time Since Last Operation / Maximum Time Since Last Operation	Yes (00:00:00 on a new process)	HH:MM:SS dwell-time gate: a product must have waited at least the minimum, and no longer than the maximum, since its previous operation to be OK to work here — for example, a glue that must cure but must not sit too long. 00:00:00 = not checked. Typed in full, up to 999:59:59: a blank or shorthand (90 , 5:30) is refused, and so is a minimum longer than the maximum while both are checked, which no product could meet; nothing is saved until both read correctly.
Maximum Number of Reruns Allowed	No (0 on a new process)	Total runs a product may make at this process: 2 allows the first run plus one re-run. 0 = not tracked.
Reset the Rerun Counter After a Rework Bench	No	Counts only the runs since the product last passed a process marked Is Rework Operation .
Is Final Operation	No	The finishing step: a check-in here runs the shipping check, which issues the shipping code when the product passes; products a machine creates here start with the final-step default disposition (System Parameter 31 instead of 30); and a product presented for check-out here is refused as already completed.
Is Rework Operation	No	Marks a rework bench, which Reset the Rerun Counter After a Rework Bench counts from.

Process attribute:

Field	Required?	What it does / valid values
Process	No	The owning process; Global = available to every process.
Attribute Number	—	Assigned automatically when saved, from 1000 up; pickers show it before the name, for example 1003 - Torque (Assembly) .
Name / Description	Name: Yes	Up to 200 characters each.
Data Type	Yes	Boolean, Numeric, String, or Curve.
Units (Y-Axis Units for a curve)	No	Any unit of measure.

Field	Required?	What it does / valid values
Retention Period	Yes (default 180-Days (~6 months))	How long readings are kept: 30, 90, 180, 365, 1260, or 2557 days, or Indefinitely . A reading is removed when this period runs out or when its product is purged (System Parameter 11), whichever comes first.
Required for Check-In	No (on by default)	A value must be collected at check-in, or skipped with the skip token; a missing one rejects the check-in. Locked on while evaluated.
Use for Product Evaluation	No (on by default)	The value is graded against recipe limits.
Allow Skip	No	The PLC may skip the value with the check-in skip token; the most recent prior value is evaluated instead.
Automatically Scrap on Reject	No	A product rejected on this attribute is scrapped without the option of a re-run. Only while evaluated.
Audit Point	No	QA re-measures it in product audits (Chapter 9).
Non-Conformance Message	No	Shown for an out-of-limit result; blank = the attribute's name. Up to 500 characters.

Related Topics

- [Products](#)
- [Recipes](#)
- [Work Orders](#) — equipment enforcement
- [Production Execution](#) — routing & WIP gates
- [OEE](#) — events, alarms & fault codes
- [System Configuration](#) — locations, units of measure, synchronization
- [Electronic Signatures & Audit Trail](#) — signed attribute changes

Chapter 3: Products

Overview

A **product** is any tracked item in IntelliTrack — a serialized unit, a lot, a batch, or a quantity of material. It is the thing the whole system exists to follow: as a product moves through production it accumulates a **status**, a **processing history**, a **genealogy** (what it was built from and what it became), and the process data measured against it. This chapter covers finding products, reading that accumulated record, creating and editing them, receiving incoming inventory, and the operations that change a product's identity — **Split**, **Merge**, and **Consume**.

Why this matters: the product record is your answer to the questions that traceability exists for — *"where is this unit, what happened to it, what went into it, and where did those components come from?"* When a customer reports a field failure or a supplier issues a recall, the genealogy and history on the product are what let you trace forward and backward in minutes instead of days.

Before You Begin

- Looking up, searching, and viewing a product (status, history, genealogy) is available to **any signed-in user**.
- **Any signed-in user** can mark a product as a reject (**Make Reject**), and it is not signed as shipped: anyone who finds a bad unit should be able to pull it out at once. Changing a reject back to a pass is the controlled act — see [Edit, reject, or delete a product](#).
- **Production** (line operators) records what happens to product on the floor: receiving material (**Inventory** → **Receiving**) and, on a product's **Edit** page, its description, location, carrier and position, its reject attributes, placing it on **Hold** (or another restrictive status), rejecting it, and assigning a work order to a product that has none.
- **Production Supervisor** (or **Administrator**) is needed for everything that releases, corrects or re-identifies a product: releasing it from Hold or any other widening status change, any pass/fail change other than to reject, renaming it, correcting its material, amount, units, expiry, shipping code or equipment, moving it to another work order or off one, and creating a product by hand, scrapping (deleting), splitting, merging, consuming and removing genealogy links. A production supervisor can also do everything Production can. Your administrator decides who holds each role ([Security and Sign-In](#)).
- With **GxP Mode** on, releasing or widening a product's disposition, changing its pass/fail result (other than to reject), renaming it, correcting its record and scrapping it ask for your **electronic signature** by default; rejecting it and placing it on Hold do not — see [Signed product changes](#) below. Releasing a product and changing its result are **release decisions**, which must be signed by someone who holds Production Supervisor ([Getting Started](#)).

- Serial numbers can be generated from the **material's serial number formula** (Chapter 4) — by machines that add products, and by **Split** when you leave the new product's serial blank. The **New Product** page does not generate one: you type it.

How to Get Here

Production → **Products** opens **Product Management**, the hub for look-ups and inventory operations. The home page's **Product Search** box goes straight to the product you name, and **Inventory** → **Receiving** opens the receiving page.

Look up a product

Finding a single unit fast is the everyday traceability task:

- **Quick lookup:** under **Current Status**, type a serial number and use **Get Product Details**.
- **Inventory Search** (the home page's **Product Status** tile, or **Back to Search Page** on any product page): **Search by Serial Number** or **Search by Carrier and Position**.

IntelliTrack matches your entry against the serial number and the **Alias**, then the shipping code, and last the serial numbers the product used to carry — so a unit that was renamed, at check-in or on its Edit page, is still found by its old serial. Deleted products are not returned. A shipping code finds a unit only while it is the unit's current code: a check-in clears it, and a check-in at the final step issues a new one (Chapter 8), so to find a unit from an old label, search by its serial number instead. The same lookup underlies recall and field-return investigations.

What a product carries — status, history, genealogy

Opening a product (**Product Details**) shows its current status and details, and from there:

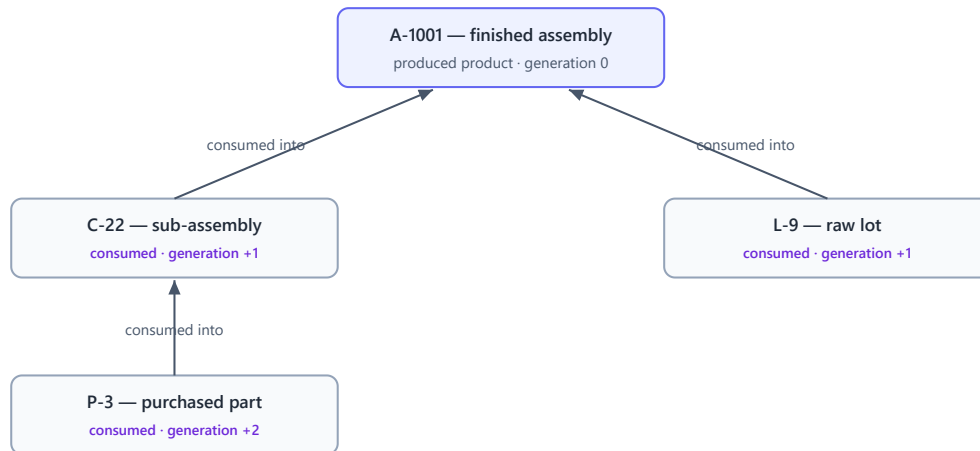
- **Processing History** — every processing step, most recent first: when it started and ended, the equipment that ran it, the material the unit ended as, and the pass/fail result. A change made on these pages — an edit, a reject, a receipt — is flagged **User Edit** and shows the person who made it where a machine step shows the equipment. **Failed check-outs** are included and flagged **Failed Check-Out**, so you can see *why* a unit was repeatedly refused at a step (Chapter 8). A step that renamed the unit shows both serials (*old --> new*).
- **Genealogy** — a parent/child tree of what the product was built from and what it became.
- **List Format** (the genealogy list) — a flat, paginated view; you set how many **generations** to show (IntelliTrack remembers your choice) and can print it.
- **BOM enforcement log** — the record of bill-of-material refusals for this product: which operation, whether the refusal was strict/blocking, the message (the refusing rule's range and the quantity already consumed against it), and the source/target serials (Chapter 6).

- **Full Report** — opens the [Product Report](#), which also lists, for every edit made on these pages, each field that changed and from what to what.
- **Electronic signatures** — appears once the product has been signed: who signed each disposition, result, reject, scrap, rename or correction, when, with what meaning and reason, and whether the signature still verifies ([Seeing who signed a record](#)).

How genealogy is recorded

IntelliTrack records genealogy with the ISA-95 convention of a **produced** product (the assembly/receiver) and a **consumed** product (the component). Building a sub-assembly increases the generation number.

Product genealogy — “produced” (assembly) vs. “consumed” (component)



Built from the genealogy rows recorded at check-in, and from Consume / Merge / Split. Each consumption is checked against the BOM; a refusal is logged in the BOM enforcement log. Trace forward (where did this part go?) or backward (what went into this unit?) from any node.

Create a product

You create a product when an item enters tracking outside the normal machine flow — a manual entry, a sample, or a unit being set up before processing. Because a record made here can start with any status and any pass/fail result, outside any process step, creating one is a **Production Supervisor** or **Administrator** task. The everyday ways product enters tracking need no one to type a record: the equipment adds it as it processes it (see the note below), and receiving supplier material (**Receiving**, below) is Production's.

1. From **Product Management**, open **New Product (Production Supervisor or Administrator)**.
2. Type the **Serial Number** — this page does not generate one — and, if the unit has a second identity such as a customer serial, an **Alias**.
3. Choose the **Material, Operational Status** and **Disposition**, enter the **Amount** with its **Units of Measure**, and say where the unit is: its equipment (the plant-model lists), **Location**, and optionally a **Carrier** and **Position**. Only units that can hold a quantity of stock are offered ([Chapter 17](#)). There

is no supplier on this page: a product created here is recorded against your own plant, the house supplier.

4. If the unit expires, set **Expiration Time** to a date and time in the future — this is what the production gate's expiry check later enforces (Chapter 8). **Expiration Date String** is the same expiry written out for printing or marking; if you fill it in, it must name the same moment, typed in your site's date format ([Date and time format](#)).
5. Optionally assign a **Work Order**. The assignment runs through the work-order procedures, so it counts the product against that order and can auto-start it (Chapter 7). If the order refuses the product — for example, quantity enforcement blocks an order that is already full — the product is still created and a warning says why; assign it from its Edit page once the order can take it.
6. Save. IntelliTrack creates the product, records an initial processing-history entry naming you, and creates the genealogy "added" record so the unit has a traceable origin from moment one.

Note: A serial number is a product's permanent identity: it must be 3 to 82 characters long, and no two products — deleted ones included — may share one. A serial that is too long is reported beside the field; one that is too short or already taken is refused by the database, so the save fails with IntelliTrack's error page instead. An **Alias** (up to 82 characters; spaces at either end are removed) must be unique too: one that another product already carries, even a deleted one, is refused with *Alias "... is already in use by serial number "...*, naming the product that holds it.

Note: If you place the product in a **Carrier**, the **Position** must be one the carrier declares *and* one no other live product already holds — the same two rules the Receive page applies (*Put the received product in a carrier*), because the database itself allows only one live product per carrier position. A position outside the carrier's capacity, or one another product occupies, is refused on the form with the reason — naming the product in the way — and nothing is saved. A deleted product holds no position, so a slot it used to fill is free to re-use.

Warning: If the serial number is on the **Blacklist**, creation is blocked unless the blacklist entry maps to a status that is OK to work — in which case the product is created with that status. A blocked creation saves nothing and shows IntelliTrack's error page with a reference ID for your administrator (the logged reason is *"Serial Number found on Blacklist..."*). This is how a quarantine blacklist (Chapter 10) reaches back to the point of creation.

Note: Machines create products too. When equipment introduces a new item, the PLC adds it on demand (default status *Released*), generating the serial from the material's formula when none is supplied, and reporting a clash with an existing serial or alias instead of creating a duplicate.

Edit, reject, or delete a product

- **Edit** (Production, Production Supervisor or Administrator; what each may change is under [Who may change what on Edit](#)) — every save writes a processing-history entry flagged **User Edit**, recording who made it and exactly which fields changed, from what to what (the list appears on the **Full Report**); any **Reject attributes (if any)** you pick are recorded against that entry. Only what is on the form can change: the supplier is not on it, and **Is In Rework State** is shown but cannot be edited here.
 - Changing the **Serial Number** renames the unit. The history keeps the old serial, so a look-up by it still finds the unit.
 - Changing the **Disposition Status** is how a unit is held or released: what a status allows comes from its **OK To Work** and **OK To Ship** flags (Chapter 10). The seeded *Hold*, for example, lets the unit go on being worked but stops it shipping. Putting a unit on Hold is Production's; releasing it again is a Production Supervisor's **release decision**.
 - Changing or clearing the **Work Order** moves the product through the work-order procedures so neither order's counts drift. A product belongs to one work order at a time, so when it is **leaving** one the page asks what that order keeps — **Hand-off** (the order built the unit and keeps credit for it: rework, a return, the next stage's order) or **Correction** (it was assigned by mistake, so it is taken back out of that order's quantities) — and will not save until you choose. The move is all-or-nothing: if the new order refuses the product, it stays where it was and you are told why (the rest of the edit is still saved, unless the edit is being signed — then nothing is). The change of work order appears in the product's history, and on both orders' Membership Changes lists ([Chapter 7](#), *Move a product to another work order*).
 - Moving the product to a **Carrier** position follows the same two rules as creating one — in range, and not already occupied — except that the product is never treated as clashing with itself, so re-saving it in the position it already holds is accepted.
 - An edit that puts the unit on *Hold*, changes its **Disposition Status** or **Operational Status**, or lowers its **Amount** raises the matching notification — **HoldPlaced**, **ManualProductAdvancement** or **NegativeInventoryAdjustmentPosted** — for anyone subscribed ([Notifications](#)).
- **Make Reject** (any signed-in user, on **Product Details**) — sets the product's pass/fail result to reject (–1) and writes history. Use it to pull a bad unit out immediately. It is offered only while the unit is not already a reject. The button opens a short form: pick the **Reject attributes (if any)** the unit failed on when you know them — they are recorded with the reject and shown on its history, but they are never required — then press **Make Reject** and confirm. (Pass/fail convention: **1 = Good**, **0 = In Process**, **negative = Reject**.) A Production user can also reject a unit on its **Edit** page by setting **Operational Status** to a reject; both ways record the same act, *Mark a product as rejected*.

Rejecting is deliberately the one change every signed-in user may make, and no role or signature policy can take it away: it identifies nonconforming product, it does not decide what happens to it. The control is on the way back. Changing a reject to a pass — or any pass/fail change other than to reject — is a **release decision** for a Production Supervisor (signed, with a reason, when GxP Mode is on), and scrapping a rejected unit is a supervisor's **Delete**. Rejecting is also not the same as a hold: putting a unit on *Hold* is a status change, which needs the Production role.

- **Delete** (Production Supervisor or Administrator) — scraps the product: it gets the *Deleted* status and is hidden from look-ups (it is not physically removed), and a history entry records the change of status, so the deletion itself is traceable. A deleted product gives up its carrier position but never its serial number or alias. Scrapping is irreversible and removes the unit from the inventory count, which is why it is a supervisor's decision.

Product Details shows **Edit** only to Production users and above, and **Split, Merge** and **Delete** only to production supervisors and administrators. **Make Reject** is shown to everyone.

Note: Whether a unit counts as completed *good* output on its work order is driven by its status becoming ship-able-and-not-workable — so changing a product's disposition is what moves work-order completion counts (Chapter 7).

Who may change what on Edit

The **Edit** page opens for the **Production** role, but what a save may change depends on the change, not just on the page. Production records what happened to a unit on the floor; a Production Supervisor releases it, corrects its committed record, or re-identifies it. Each field you change is judged on its own, and the whole edit needs the highest role any of its changes needs:

Change on Edit	Production	Production Supervisor or Administrator
Description, Location, Carrier, Position, Reject attributes	Yes	Yes
Disposition Status — a restrictive change, such as <i>Released</i> to <i>Hold</i>	Yes	Yes
Operational Status — to a reject	Yes	Yes
Work Order — assigning one to a product that has none	Yes	Yes
Disposition Status — any other change: releasing from <i>Hold</i> , to <i>Shipped</i> or <i>Scrapped</i> , and so on	—	Yes (a release decision)
Operational Status — any other change, such as a reject back to good	—	Yes (a release decision)
Serial Number or Alias (a rename)	—	Yes

Change on Edit	Production	Production Supervisor or Administrator
Material, Amount, Units of Measure, Expiration Time, Expiration Date String	—	Yes
Shipping Verification, the equipment lists	—	Yes
Work Order — moving the product to another order, or off one	—	Yes

A disposition change is **restrictive** when it only narrows what the unit may do: it turns **OK to Work** or **OK to Ship** off (or both), turns neither on, and leaves **Include In Count** as it was. IntelliTrack decides this from the two statuses' own flags, so a status your plant adds is judged the same way. *Released* to *Hold* is restrictive — the unit can still be worked but can no longer ship. Anything else is permissive or lateral and needs a supervisor: releasing from Hold widens what the unit may do, and a move to *Scrapped* also takes the unit out of the inventory count.

What this means on the page and when you save:

- **A Production user sees only what they may change.** The fields only a supervisor may change are shown read-only. **Disposition Status** offers the unit's current status and the statuses that would restrict it; **Operational Status** offers its current result and, unless it is already a reject, **Failed Products**.
- **A refused edit changes nothing.** An edit that needs a Production Supervisor is refused before any password is checked and before anything is written: you see the **Unauthorized** page, naming the role the change needs. Save the parts you may change on their own, and ask a supervisor for the rest.
- **One edit that mixes the two** — a new location and a release from Hold, say — needs a Production Supervisor for the whole save.
- The rule under [Signed product changes](#), that one edit cannot mix signature meanings, still applies on top of this.

Signed product changes

With **GxP Mode** on, the changes below are electronic-signature acts — signed by default unless the table says otherwise ([Which acts can be signed](#)). Your site's signature policies can change that per site, area, workcell, process or material group ([Signature policies](#)), and with GxP Mode off nothing is signed. What you sign is exactly the change: the product as it stands, and each field from its old value to its new one.

Where	Change	Signed as	Meaning	Reason
Edit → Save	Disposition Status , released or widened (a permissive or lateral change) — a release decision	Change a product's disposition	Disposition Decision	Optional

Where	Change	Signed as	Meaning	Reason
Edit → Save	Disposition Status , restricted (placing a unit on <i>Hold</i> , for example)	Restrict a product's disposition	Disposition Decision	Optional — signed only if your site's policy turns it on
Edit → Save	Operational Status , any change other than to a reject — a release decision	Change a product's pass/fail result	Disposition Decision	Required
Edit → Save	Operational Status to a reject	Mark a product as rejected	Disposition Decision	Optional — signed only if your site's policy turns it on
Edit → Save	Serial Number or Alias	Rename a lot or serial number	Authored	Optional
Edit → Save	Material, Amount, Units of Measure or Expiration Time	Correct a committed product record	Authored	Required
Edit → Save	Work Order	Move a product to another work order	Authored	Optional — signed only if your site's policy turns it on
Delete → Delete	Status becomes <i>Deleted</i>	Scrap (delete) a product	Disposition Decision	Optional
Product Details → Make Reject	Result becomes –1 (reject)	Mark a product as rejected	Disposition Decision	Optional — signed only if your site's policy turns it on

The two **release decisions** in the table ship with a required signer role: with GxP Mode on they must be signed by someone who holds **Production Supervisor**, so an administrator who does not also hold that role cannot sign them ([Who may sign: roles](#)). Rejecting a unit and restricting its status are the opposite: they only stop product, so they ship unsigned, and a signature policy may turn a single signature on for them but can never require a particular role or a second signer — configuration cannot quietly take them away from anyone.

The rest of the form — description, location, carrier, position, shipping code — is never signed; like every change, it is recorded in the audit trail.

An edit that needs a signature is saved in two steps. The first **Save** writes nothing: the page says *These changes must be signed before they are saved* and shows an **Electronic signature** panel above **Save** (*I sign the changes to Product ... listed below*); **Show exactly what is being signed** lists the product and every change. Check them, enter your password (and a reason, if asked) and **Save** again: the signature, any work-order move and the edit are saved together, or not at all. One signature carries one meaning, so an edit that mixes a disposition decision (status or result) with an authored change (rename or correction) is refused — the message names the fields behind each — and you save them as two edits. On **Delete** the

panel is above the button; where your site has turned on the signature for rejects, **Make Reject** opens its panel instead of asking you to confirm. How to sign, and what each refusal means: [When you are asked to sign](#).

Receive incoming inventory

Use **Receiving** (Production, Production Supervisor or Administrator) to bring supplier material into the system. Receiving is floor work, not a look-up — it puts new product into inventory — so it needs the Production role, and the **Inventory** menu that holds **Receiving** is shown only to those who hold it. The receipt creates the product with a good pass/fail result, records who received it and when, and, when you give a **Supplier Lot Id**, links the product to that lot so it can be traced back to it (the first receipt of a lot also creates a product for the lot itself).

1. Open **Inventory** → **Receiving (Receive Material)**.
2. Choose the supplier in **Supplier Id** — or leave it at -- *Select* -- for material from your own plant, the house supplier — and the **Material**; enter the **Serial Number**, **Amount** and **Units of Measure**, the **Disposition Status** and the expiry, and optionally the **Supplier Lot Id**, a **Work Order**, a **Location** and a **Carrier**.
3. Select **Receive Material**. IntelliTrack evaluates the receipt and shows **Material Receipt Accepted** or **Material Receiving Rejected** with the reason.

A receipt is rejected when its serial number or supplier lot is on the blacklist, when the serial number already belongs to a product, or when the supplier lot is already a product from a different supplier or is the serial being received. A blacklist entry whose status is OK to work does not stop a receipt, and the product keeps the status you chose — unlike **New Product**, which gives it the blacklist entry's status. An accepted page returns to **Receiving** by itself after a few seconds — unless the work order refused the product, in which case the product is still received and the page stays to say why.

If something on the form stops the receipt, such as no material, no status, a status the supplier does not allow, a unit that cannot hold a quantity of stock, or a carrier position that is out of range or already taken, the form comes back with the reason under that field and nothing is received. **Units of Measure** offers only units that can hold stock ([Chapter 17](#)); left empty, the product is received in *Each*.

The supplier's status limits the starting status

A supplier has a status of its own, its **Default Material Status** (Chapter 4), and material received from it cannot start in a status that allows more than that one does. If the supplier's status is not *OK to work*, the received product can only start in a status that is not *OK to work* either, so production refuses it at check-out (Chapter 8) until a Production Supervisor releases it to a workable status (**Edit**, above — a release decision). If the supplier's status is not *OK to ship*, the product can only start in a status that shipping verification refuses (Chapter 10). This is the supplier-quality interlock: putting a supplier on probation holds everything received from it at the point of receipt, without relying on whoever receives it to remember.

When you choose a supplier, the statuses it rules out are greyed out in **Disposition Status**, a note under the list names the supplier's status, and a status you had chosen that it rules out is replaced by the first one it allows. A supplier whose status allows both work and shipping, such as the seeded *Released*, limits nothing; the status then starts at *Released*.

Note: The seeded supplier statuses *Contained*, *Probation* and *Closed* allow neither work nor shipping, and the only seeded product statuses that allow neither are *Closed*, *Scrapped*, *Deleted* and *Destructive Audit*. Before receiving from such a supplier, add a product status for material awaiting inspection that is neither OK to work nor OK to ship, with **Available To Users** ticked so the receiving page offers it ([Manage dispositions](#)). Until one exists, those four are all its material can be received in, and the first of them, *Closed*, is chosen for you — so check **Disposition Status** before you receive: a product received as *Deleted* cannot be looked up again. The note under the list says that no status is available only when every product status allows work or shipping.

Put the received product in a carrier

Carrier is optional. Leave it at *None* for material that does not go into one of the plant's carriers. To place the product in a carrier, choose the carrier and a **Position**; the hint under **Position** shows the carrier's valid range. A position outside the carrier's capacity, or one that another product already holds, is refused with the reason. A deleted product holds no position.

Set the received lot's expiry

The **Expiration Time** a lot is received with becomes its expiry date: the production expiry gate refuses the lot at check-out once that date has passed, and the queue works lots with the soonest expiry first (Chapter 8). It decides how long the material can be used, so choose the **Expiration Time Units** first — they decide what else you enter:

- **None** — the lot never expires. It has no expiry date, is never refused for expiry, and in the queue it comes after every lot that has one.
- **Minutes, Hours, Days, Months (30-Day Months), Months (Calendar Months) or Years** — enter the shelf life in **Expiration Duration or Shelf Life**, and IntelliTrack works out the Expiration Time. The shelf life is counted from when you press **Receive Material**, which is when the product is created: the date shown while you fill in the form is a preview, and it is worked out again as you submit, so time spent on the rest of the form is not taken off the shelf life. A calendar month keeps the day of the month where it can (31 January plus one month is 28 February); a 30-day month is always 30 days.
- **Custom Expiration Date** — enter the date yourself, for example the expiry the supplier printed on the label. The duration is not used.

Read a supplier barcode

With **Read Barcode** ticked, scanning a supplier's label into **Barcode String** fills in the form from the label, then locks the barcode field so you can correct the form by hand. Tick **Read Barcode** again to scan another label. From an AIAG label:

Label field	Fills in
Date (D , 1D , 2D , 3D , 4D , 7D)	Expiration Time , with the units set to Custom Expiration Date .
Serial number (S)	Serial Number .
Lot (T , 1T)	Supplier Lot Id .
Quantity (Q)	Amount .
Quantity and unit (7Q)	Amount and Units of Measure . The two are run together on the label ("2.5KG"), and are split at the end of the number — decimal point included. A quantity printed with no unit ("100") still fills Amount .
Work order (W)	Work Order , when a work order by that name is in the list.
Supplier (V)	Supplier Id (the supplier list) — the supplier whose Supplier Id is the label's vendor id.
Part number (P , 1P)	Material — the material of that Name , at its latest revision that is enabled for production.
Unit (3Q)	Units of Measure — the unit of that Abbreviation , or failing that of that Name .

Those last three are matched on the fields a label actually carries rather than on the text the dropdowns show, which is why what you enter elsewhere in IntelliTrack decides whether a scan fills them in:

- A vendor id is a supplier's **Supplier Id** (Suppliers → Supplier Id), not its name. A supplier you receive from with that field left empty can never be chosen from a label — fill it in with the id that supplier prints.
- A part number is a material's **Name**, which names a family of revisions rather than one revision, so the scan takes the same revision the shop floor gets when a machine asks for that part by name: the latest one whose approval status is enabled for production, which is **Active** (Chapter 4). A part with no Active revision — every revision still being authored or approved, or retired — is not chosen at all.
- A unit code is a unit's **Abbreviation** (Units of Measure), so a label's "EA" finds *Each* (*ea*). Capitals do not matter on any of the three.

A value none of them matches leaves its dropdown exactly as it was, rather than guessing at a near miss — so check **Supplier Id**, **Material** and **Units of Measure** before you receive, and if one is empty after a scan, the label's value is not in IntelliTrack under the field above.

A label date carries no time of day, so the lot expires as that day begins, and a month-and-year date (**7D**) expires as that month begins. Reading the date early refuses a lot a day before its label says rather than working it a day after. A one-digit year (**3D**) is read as the nearest year ending in that digit (the earlier one if two are equally near), so a lot that expired recently is not read as good for another decade. A date that is not a real calendar date is ignored and leaves the expiry as it was.

Split, Merge, and Consume — changing identity, recording lineage

These change what a product *is*, and each is recorded in genealogy so the lineage is never lost. All require the **Production Supervisor** or **Administrator** role, and **Product Details** offers **Split** and **Merge** only to those roles. Each writes genealogy with no equipment or process step behind it, and a link a Merge or Consume makes can never be removed afterwards (see [Remove a genealogy link](#)), so they are supervisor decisions. The genealogy an operator records every day is captured at check-in (Chapter 8) and by the machines.

- **Split** — divide one product into itself plus a new product (e.g., splitting a lot). The **Split** page opens empty, even from a product's page: enter the product to split in **Produced Serial Number**, the new product's serial in **Consumed Serial Number** — or leave it blank to have one generated from the material's serial number formula — and the quantity to move in **Split Amount (for Second Product)** with its **Units of Measure**. The new product is linked to its origin. If the original is on a **work order**, the new product joins the same order in the same step, so the order's counts are unchanged by the split; to send the new lot to a different order, move it afterwards (Edit, above).

Splitting a part that is in a carrier. The original keeps its position — splitting does not move it. The new product is created on the same carrier but in **no position**, and is then placed in the target carrier and position you entered on the form. Give it a position the target carrier declares and that no other live part holds, or the split is refused before anything is created and nothing about the original changes. You cannot send the new product to the original's own position: the original is still in it. Leave the target carrier at *None* and the new product is created outside any carrier, ready to be placed from its Edit page. The placing needs the new product's serial: when you leave **Consumed Serial Number** blank for IntelliTrack to generate, the new product stays on the original's carrier with no position, and you place it from its Edit page.

- **Merge** — combine two products **of the same material type** into one. The product in **Main Product Serial Number** survives and takes the whole quantity of the one in **Produced Serial Number**, which is emptied and closed. Tick **Generate New Serial Number**, or type a **New Resulting SN**, to give the survivor a new serial.
- **Consume** — associate a component (source) into a parent (target), recording the consumption in genealogy. This is the manual equivalent of the component-consumption rows captured at check-in.

Units on a split or merge. A split's amount can be given in a different unit from the lot's own when both measure the same kind of thing — kilograms off a lot kept in pounds — and IntelliTrack converts it, so both lots still add up; a merge converts the merged quantity into the survivor's unit the same way. The unit lists on these pages offer only units that can hold a quantity of stock ([Chapter 17](#)). A pair of units that cannot be converted, or a unit classified as measurement only, is applied as entered and logged as a warning under the shipped setting of System Parameter 26 (*Skip Units of Measure Verification*); set it to 0 to have such a split or merge refused instead. Splitting off more than the lot holds is refused while System Parameter 25 (*Check Consumed Quantity*) is on, as it ships.

Remove a genealogy link

From the genealogy list (**List Format**), Production Supervisor and Administrator users can delete an individual genealogy link — a supervisor decision, because it removes part of the manufacturing record. Links created by a **Merge** or **Consume** cannot be removed this way — those represent physical combination, not a correctable mistake — and neither can the product's own birth record, the entry that anchors it to genealogy: the page explains why and offers no delete button, because without it the product would drop out of genealogy, where-used, recall and quarantine searches, and expiry checks. A removed link is not erased: it is marked removed, with who removed it and when, and traces no longer follow it.

Field Reference

Inventory Search:

Field	Required?	What it does / valid values
Serial Number	Yes (serial search)	Serial number, alias, shipping code or former serial to find.
Carrier	Yes (carrier search)	Carrier name to search within.
Position	Yes (carrier search)	Numeric position of the product in the carrier.

Create / Edit product:

Field	Required?	What it does / valid values
Serial Number	Yes	The product's permanent identity: 3 to 82 characters, never shared with another product (deleted ones included). Typed, not generated, on this page. Changing it on Edit renames the product.
Alias	No	A second identity (a customer serial, a laser mark), matched by look-ups and by machines at check-out and check-in. Up to 82 characters, unique, spaces at either end removed.
Description	No	Free text, up to 500 characters.
Material	Yes	The material this product is an instance of.

Field	Required?	What it does / valid values
Operational Status	Yes	The pass/fail result: Good Product (1), In-Process Inventory (0) or Failed Products (–1, a reject).
Disposition (Create) / Disposition Status (Edit)	Yes	The product's status; its OK to work and OK to ship flags decide whether production and shipping accept the unit (Chapter 10).
Is In Rework State	No	Set on Create; shown but not editable on Edit.
Amount / Units of Measure	Yes	The quantity (0 or more) and its unit. Only units that can hold a quantity of stock are offered, plus the unit the product already uses.
Expiration Time	No	If set, must be in the future; enforced by the production expiry gate.
Expiration Date String	No	The expiry written out for printing or marking (up to 82 characters). If set, must name the same moment as Expiration Time, in the site's date format.
Equipment (plant-model lists), Location	No	Where the product is.
Carrier / Position	No	Must be a position the carrier declares and that no other live product holds.
Shipping Verification	No	The product's shipping code (up to 100 characters), normally issued when it passes its final step (Chapter 8).
Lot Number	—	Edit only. Not used: it is always empty and is not saved.
Work Order	No	Work order to assign; updates that order's counters. A product is on at most one order at a time.
Leaving Its Work Order	When the product leaves an order	Edit only, shown once the Work Order is changed or cleared. Hand-off: the order it leaves keeps credit for it. Correction: its quantity is taken back out of that order. No default.

The page has no supplier field: a product created here belongs to the house supplier, and an edit leaves the supplier as it is. **Create** needs the Production Supervisor or Administrator role; on **Edit**, which of these fields a Production user may change is under [Who may change what on Edit](#).

Related Topics

- [Materials & Suppliers](#) — materials, SN formulas, supplier statuses
- [Bills of Material](#) — the BOM the enforcement log checks against
- [Work Orders](#) — assigning products and fulfillment counting
- [Production Execution](#) — how products are processed and genealogy is built

- [Quarantine & Disposition](#) — statuses, blacklist
- [Reports](#) — the Product Report and traceability reporting
- [Notifications](#) — the hold, manual-advancement and negative-adjustment events
- [System Configuration](#) — units of measure, the site's date format, system parameters
- [Electronic Signatures & Audit Trail](#) — signing product changes, and the audit trail

Chapter 4: Materials & Suppliers

Overview

A **material** is the definition of a thing you make or consume — its name, revision, shelf life, serial-number rule, and approval state. Products are *instances* of materials; recipes and bills of material are built *for* materials. A **supplier** is a source of incoming material and carries a status of its own, which limits the status material received from it can start in. This chapter covers maintaining both, plus the **approval lifecycle** that materials (and recipes and BOMs) move through.

Why this matters: a material sits at the centre of a web of dependents. Its **serial-number formula** generates product serials; its **shelf life** is what the production expiry gate enforces; its **approval state** controls whether recipes and BOMs built on it may go live — and retiring or deleting an active material **cascades** to those dependents. Understanding that web is what keeps a routine material change from quietly invalidating production downstream.

Before You Begin

- **Viewing** materials and suppliers is available to **any signed-in user**.
- **Creating, editing, duplicating, and deleting** materials, requesting and approving their status changes, and reading their **Approval History** require the **Recipe** or **Administrator** role. Your administrator decides who holds each role ([Security and Sign-In](#)).
- **Managing suppliers** requires the **Administrator** role.
- How many approvals a status change needs, whether they must come from different people, and whether each act is electronically signed are set centrally by the site's **signature policies** (**Configuration** → **Additional Configuration** → **Compliance** → **Signature policies**, Administrators — [Chapter 20](#)), not on the material. The material's **Required Material Approvals** field is read-only: it shows the count the policy set when the last status change was requested. When your installation was upgraded to this release, each material's old count was carried into the policies — a material that differed from the rest got a row of its own, marked *Migrated exception* — so no requirement changed on the day.
- A material's **ECO Pattern** can be changed by an Administrator at any time, and by a Recipe user only while no other revision of that material name exists.

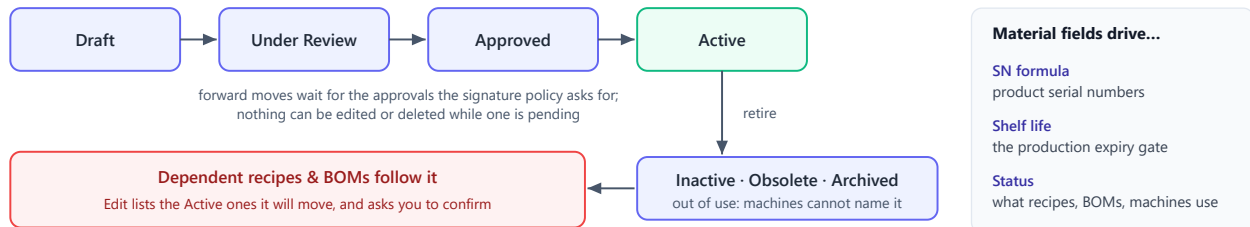
How to Get Here

- Materials: **Product Definition** → **Materials**.
- Suppliers: **Configuration** → **Additional Configuration** → **Materials & Logistics** → **Suppliers**.

The approval lifecycle — and how it cascades

Materials, recipes, and BOMs share one controlled workflow: a revision is written in **Draft**, submitted for review, approved, put into service (**Active**), and eventually retired. Recipes (Chapter 5) and bills of material (Chapter 6) follow the same rules. The states and the *effect* of moving between them:

Material lifecycle — and what it touches



Only Draft can be edited. Back to Draft only while no recipe or BOM beyond Draft uses the material.

A supplier's status limits what it sends in →

received product cannot start in a status that allows more (on probation: held until QA releases)

Status	Meaning — and what it allows
Draft	Newly created, or sent back. The only status in which the material's fields can be changed.
Under Review	Submitted for review; the fields are locked.
Approved	Reviewed and approved, not yet in service. Recipes and BOMs for the material can now be approved too.
Active - Pending Effective Date	Set by the system, never chosen by hand: a move to Active with a future Effective Date waits here, and the material becomes Active on that date by itself.
Active	In service, and the only status enabled for production: a machine or a scanned label that names the material without a revision gets its highest Active revision, and its recipes and BOMs can now be activated.
Inactive, Obsolete, Archived	Retired: kept for the record but out of use — a machine can no longer name the revision. Obsolete and Archived materials are hidden from the list unless you tick Show All Statuses? .

The moves you can make (the **Status** list on **Edit** offers these and shows the rest greyed out):

From	To
Draft	Under Review
Under Review	Draft, Approved
Approved	Draft, Active

From	To
Active	Inactive, Obsolete, Archived — or Draft
Inactive	Under Review, Obsolete, Archived — or Draft
Obsolete	Inactive, Archived — or Draft
Archived	Inactive, Obsolete — or Draft

Key rules and effects:

- A new material starts in **Draft**, and only a Draft material can be changed: once it leaves Draft, **Edit** changes only its **Status** (with the **Effective Date** a move to Active carries) and its **Image**, and every other field is locked. To change a material that has moved on, duplicate it as a new revision.
- **Back to Draft** is refused while any recipe or BOM beyond Draft uses the material (*This material cannot be returned to Draft because a recipe or BOM beyond Draft depends on it. Create a new revision instead.*).
- **Moving forward waits for approval.** A move to Under Review, Approved or Active — and bringing a retired material back to Under Review or Draft — is an approval step: it needs the approvals your signature policy sets for that move ([Chapter 20](#)), whether or not GxP Mode is on. With one approval and no distinct-signer rule — as shipped — your own change is that approval and takes effect at once. When the policy asks for more than one, or for distinct signers, the change is **pending**: the material stays where it is and shows the status it is waiting for until enough other people approve (nobody counts twice). With distinct signers, the person who asked does not count and cannot approve. The requirement is fixed when the change is requested; a later policy change applies to the next request, not this one. Everything else takes effect at once: retiring a material, moving it between the retired states, and sending a material that is not retired back to Draft.
- While a change is pending the material **cannot be edited or deleted**.
- **Active + future Effective Date** → **Active - Pending Effective Date** until that date — letting you stage a changeover.
- **Dependents follow the material.** A status change (other than to Active) brings the material's recipes and BOMs into line: retiring the material retires every dependent at Approved or beyond with the same status, any other move brings every dependent that is further along — or retired — to the material's new status, and a pending change of theirs that the move overtakes is cancelled. When you change the status of an **Active** material on **Edit**, **Confirm material status change** lists the Active recipes and BOMs that will change, and nothing is saved until you **Confirm** — so the impact is never a surprise.
- Every completed status change raises a **MaterialApprovalStateChanged** notification for anyone subscribed ([Notifications](#)).

Electronic signatures on the lifecycle

With **GxP Mode** on, every lifecycle move is an electronic-signature act, signed by default; your signature policy decides, and with GxP Mode off nothing is signed ([Which acts can be signed](#)). The approval counts above apply either way. The act, and the default meaning of the signature, follow from the move:

Move	Signed as	Meaning
to Under Review	Submit for review (Draft to Under Review)	Reviewed
to Approved	Approve (Under Review to Approved)	Approved
to Active	Activate (Approved to Active)	Approved
to Inactive, Obsolete or Archived	Retire (to Inactive, Obsolete or Archived)	Approved
to Draft	Return to Draft	Rejected
Reject Pending Status	Reject a pending status change	Rejected
Cancel Pending Status	Withdraw a pending status change	Rejected

A reason is optional by default; a signature that means *Rejected* needs a comment. When you pick a new **Status** on **Edit**, the page shows the **Electronic signature** panel for that move and you sign as you **Save** — the signature covers the material as it is saved. Approving someone else's pending change signs the act of the move it completes (approving a pending move to Active signs *Activate*). The signatures appear in the **Electronic signatures** section of **Details** and beside each act in **Approval History**.

Approve, reject, or revoke a pending change

From **Details**, when a pending approval exists: **Approve Pending Status** (records your approval; the change takes effect when enough are gathered), **Reject Pending Status** (ends the request; the material stays where it was), or **Cancel Pending Status** (the person who asked withdraws the request) — each with a **Comment**. Approving and rejecting need the Recipe or Administrator role; only the person who asked can cancel. Nobody can approve twice in one request, and when the policy requires distinct signers the person who asked is not offered **Approve Pending Status** at all. When a signature is required, each of the three opens as its own section with its own panel and a **Sign and approve**, **Sign and reject** or **Sign and withdraw** button, because each is signed with its own meaning. An act that no longer applies — someone decided the change first — changes nothing and says so. **Approval History** shows every request: who acted, when, their comments and, where signed, each signature.

The warning banner at the top of **Details** names the change in flight — the status the material is in now and the status it is waiting to move to, e.g. *from Draft to Under Review* — so whoever approves, rejects, or cancels can see exactly which transition they are acting on without opening the history. The same wording explains why an **Edit** is refused while the request is open. Below it, **Pending Status, Approvals** (for example *1 of 2*) and **Fully Approved** show how far the request has got.

Create a material

1. Open **Product Definition** → **Materials** and create a new material (**Recipe** or **Administrator**).
2. Enter **Name** and **Revision** (the pair must be unique) and a **Description**.
3. Optionally set the fields that drive downstream behavior:
 - **Expiration Duration or Shelf Life** with its **Expiration Time Units** — what the production expiry gate later enforces on products of this material.
 - **Serial Number Formula** — the rule that generates serials for those products (so you don't have to assign them by hand); **Help** beside the field explains what a formula can contain.
 - **Material Code** — what the formula's material part (`<M>`) puts into a serial; left empty, the **Name** is used instead.
 - **ECO Pattern** and **Engineering Change Order (ECO)** — to record the engineering change order behind this revision, choose the pattern your site uses for change-order numbers; the **ECO** is then required and must match it. With no pattern chosen, no ECO is kept. (ECO patterns are set up with the recipe ECO keys — [Recipes](#).)
 - An **Image** (smaller than System Parameter 100, *Max Image Size*), and **Synchronize Across Sites** (a flag kept with the material; every material reaches every site whether or not it is ticked).
4. Save — the material is created in **Draft**, with its **Required Material Approvals** taken from the signature policy. **Status** and **Effective Date** cannot be set here; they belong to the approval lifecycle above.

Warning: A duplicate Name + Revision is rejected (*Material name already exists in the database*); an over-size image is rejected with the maximum-size message; an ECO that does not match its pattern is rejected (*ECO does not match the selected ECO pattern*).

Duplicate a material

Duplicate Material copies a definition as either a **New Revision** (same name, next revision proposed) or a **New Material** (new name/description). The copy carries over the image, shelf life, serial-number formula, code, and cross-site setting — so iterating a material keeps its operational behavior intact — and starts in **Draft**, whatever the status of the one it was copied from. The ECO and its pattern are not copied: give the new revision its own. This is also how you change a material that has left Draft: duplicate it as a new revision, edit that, and take it through the lifecycle. (Anyone can open the **Duplicate Material** page; saving the copy needs the Recipe or Administrator role.)

Edit or delete a material

- **Edit** — in **Draft**, change the fields and, if appropriate, request a status change; outside Draft, only the **Status** (with its **Effective Date**) and the **Image** can change. The image identifies the material to

operators rather than being part of its approved specification, so it can be replaced at any status. While a change is pending, **Edit** is not offered and opening it takes you to **Details**.

- **Delete** — soft-deletes (renamed and hidden). It is not offered while a change is pending. Deleting an **Active** material first moves the recipes and BOMs built for it back to **Approved** — the delete page does not list them — and removes routes that referenced only this material. Deleting is not a signed act.

View a material

Details (any signed-in user) shows the material's fields and status, any pending change with the actions above, its **Electronic signatures** once it has been signed, and where it is used:

- **Where Used - Bill of Material Membership** — every bill of material that consumes this material as a component, with the BOM's revision and status and links to its composition and details (Chapter 6). Check it before you retire or replace a component.
- Its GTINs (Global Trade Item Numbers) with their pack level and quantity, maintained from here by Administrators ([Recipes](#)).

Manage suppliers

1. Open **Suppliers** (search by name/description; paged).
2. **Administrators** can **Create**, **Edit**, or **Delete** a supplier.
3. A supplier has a **Supplier Id**, a **Name**, a **Description**, and a **Default Material Status**: the supplier's own status (*Released*, *Contained*, *Probation*, *Closed*, or one you add), which limits the status material received from it can start in.

The **Supplier Id** is the code the supplier prints on its labels as the vendor id: receiving reads it to choose the supplier from a scanned label, so a supplier without one can never be picked from a label (Chapter 3, *Read a supplier barcode*). It is optional, but no two suppliers may be created with the same one (*Supplier Id already exists in database*). The **Name** is required and must be unique.

Warning: Delete is a soft delete: the supplier is hidden from the lists (its **Supplier Id** is given a deleted suffix), and the products already received from it keep it.

Why the Default Material Status matters: it is how you put a supplier "on probation". Material received from a supplier cannot start in a status that allows more than the supplier's status does, so give the supplier a status that is not OK to work and everything you receive from it starts in a status that is not OK to work either. It is held until QA releases it, automatically, at the point of receipt (Chapter 3). The limit reads the status's **OK To Work** and **OK To Ship** flags, so a status you add behaves the same way.

Units of measure

A material has no unit of its own: the **units of measure (UoM)** belong to the quantities made of it — a product's or received lot's **Amount**, a split, a work order, a bill-of-material line, a carrier's contents. Defining units is an administrator task, and so is classifying each one by what it measures ([Units of measure](#)). That classification is what everyday users see: wherever a quantity of stock is entered, the unit list offers only units whose dimension can count stock (plus the unit a record already uses), and receiving, work orders, carriers and BOM lines refuse one that cannot ([Inventory quantity or measurement only](#)). Quantities in two units of the same dimension — kilograms and pounds — are converted for you when material moves between lots (Chapter 3).

Field Reference

Material (create/edit):

Field	Required?	What it does / valid values
Name	Yes	Material name, up to 82 characters; the name a machine or a scanned label uses for the material. Name + Revision must be unique.
Revision	Yes	Revision identifier, up to 20 characters.
Description	No	Free-text, up to 200 characters.
Material Code	No	Up to 82 characters; replaces <code><M></code> in the Serial Number Formula (the Name is used when it is empty).
Serial Number Formula	No	Rule for generating product serial numbers, up to 50 characters.
Expiration Duration or Shelf Life + Expiration Time Units	No	Drives the production expiry gate for products.
Status	Edit only	Requests a lifecycle move; only the moves allowed from the current status can be chosen.
Required Material Approvals	Read-only	The approvals a status change needs, set from the signature policy each time a change is requested.
Effective Date	No	Only with a move to Active: a future date parks the material in Active - Pending Effective Date until then.
ECO Pattern	No	The pattern a change-order number must match. Changeable by an Administrator, or while no other revision of the name exists.
Engineering Change Order (ECO)	With an ECO Pattern	The change order behind this revision, up to 100 characters; must match the pattern.

Field	Required?	What it does / valid values
Synchronize Across Sites	No	A flag kept with the material. Every material is shared with every site through the Config database whether or not it is ticked.
Image	No	Picture shown to operators; must be smaller than System Parameter 100. Can be replaced at any status.

Supplier (create/edit):

Field	Required?	What it does / valid values
Supplier Id	No	The vendor id the supplier prints on its labels (up to 82 characters); receiving matches scanned labels on it. No two suppliers may be created with the same one.
Name	Yes	Supplier name, up to 50 characters; must be unique.
Description	No	Free-text, up to 200 characters.
Default Material Status	Yes	The supplier's own status. Material received from it cannot start in a status that is OK to work or OK to ship when this one is not — your supplier-quality interlock.

Related Topics

- [Products](#) — instances of materials; receiving
- [Recipes](#) — built for a material; follow it down the lifecycle; ECO keys and GTINs
- [Bills of Material](#)
- [Production Execution](#) — the expiry gate
- [Notifications](#) — the MaterialApprovalStateChanged event
- [System Configuration](#) — units of measure, system parameters
- [Electronic Signatures & Audit Trail](#) — signature policies, approval counts, signing

Chapter 5: Recipes

Overview

A **recipe** is the master specification for making a product: it ties a finished-product **material** to the **limits** each measured attribute must meet. But a recipe sitting in the library does nothing on its own — its value is that it becomes the **yardstick every production measurement is graded against**. This chapter covers authoring and approving recipes, and — just as important — *how a recipe actually reaches a machine and controls a run*, because that is where the "why" of every recipe field lives.

Why this matters: when an operator or a machine records a measurement (Chapter 8), IntelliTrack compares it to the recipe's limits to decide pass or fail. Product audits (Chapter 9) check against those same limits, and SPC charts draw them as the specification lines their capability figures are measured against. Get the recipe right and good product passes and bad product is caught; get it wrong and you either scrap good parts or ship bad ones. The recipe is the single, change-controlled source of those limits.

Before You Begin

- **Viewing** the recipe list, a recipe's details, its values, and its control instances is available to **any signed-in user**. The printable **Report**, **Approval History**, and **Validate** need the **Recipe** or **Administrator** role.
- **Creating, editing, duplicating, deleting, and approving** recipes requires the **Recipe** or **Administrator** role.
- **Approval counts are not set on the recipe**. How many approvals a status change or an override needs comes from your site's **signature policies** (Chapter 20). The recipe's **Required Recipe Approvals** and **Required Override Approvals** fields are filled in from those policies and cannot be edited.
- With **GxP Mode** on, moving a recipe through its lifecycle — and approving, rejecting, or withdrawing a pending change — is **electronically signed** by default. See [Electronic Signatures & Audit Trail](#).
- Define the finished-product **material** (Chapter 4) and the relevant **process attributes** (Chapter 2) first — a recipe references both.
- In a multi-site system, recipe names are **per-site** — the same name at two sites is two different recipes. See *Recipes and sites* below before naming anything.

How to Get Here

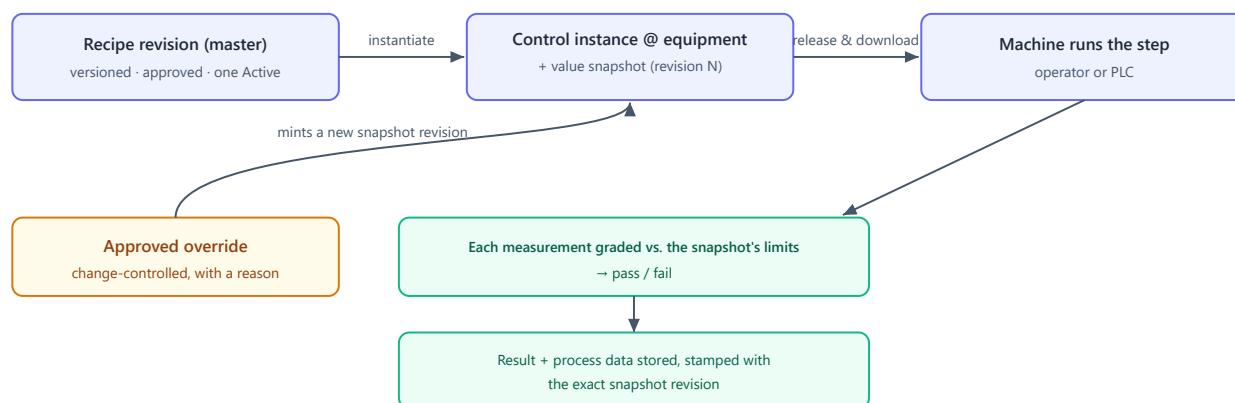
Product Definition → **Recipes** opens the recipe list. Each revision's row links to its **Details**, **Values**, **Control Instances**, printable **Report**, and **Approval History** (and, for the Recipe and Administrator roles,

Edit and **Duplicate**). Deviations and overrides are reached from a control instance's detail page.

From recipe to the machine — the pipeline that makes a recipe matter

This is the part that turns a recipe into control. Understanding it explains why recipes are versioned, why limits can't be edited ad hoc, and why an override is a controlled event.

From recipe to the machine — and back



Product audits grade against this same snapshot, and SPC draws its spec lines from it — limits are fixed per revision, never set ad hoc at the machine.

A deviation (actual differs from the recipe) is logged against the control instance and can seed an override request.

In words:

1. You author a **recipe** and its **values** (the limits), and move it to **Active** through the approval lifecycle.
2. The active recipe is **instantiated** as a **control instance** for a specific piece of equipment. Each instance carries an immutable **value snapshot** (the limits as they stood). Instances are created either by the equipment's download or on the web, where they wait **In Review** until someone releases them (see *Control instances* below).
3. When the equipment runs a product, IntelliTrack resolves that equipment's **active control instance** and grades each measured value against the snapshot's limits — pass or fail. Only attributes in the snapshot are graded, required or scrapped on: one mapped to the equipment after the instance was built is recorded but not evaluated until the next instance ([Chapter 2](#)). The result and process data are stored **stamped with the snapshot revision**, so you can always answer "which exact limits was this unit made under?" The Product Report (Chapter 12) links each processing event to the control instance recipe at the revision in force at the time.
4. **Product audits** resolve their limits from that same stamped snapshot, and **SPC** charts draw its alarm limits as their specification lines (the control limits themselves are calculated from the process data, and a chart only covers samples made under one control-recipe revision) — which is why this guide says repeatedly that limits are **recipe-driven**.

5. A controlled change to a live instance — an **override** — does not edit the snapshot in place. Once approved, it is **applied as a new snapshot revision** (when the machine confirms it, or at the next web check-out), so the change is auditable and the prior history stays intact.

Running without recipes

Data-logging modules, and modules where the PLC grades the part.

Not every plant grades its parts in IntelliTrack. Some let the PLC decide pass and fail and use IntelliTrack to record what happened. That is a supported configuration, and what selects it is the **attributes mapped to the module** — there is no separate setting on the equipment.

A module needs an active control instance **only if at least one attribute mapped to it is marked Use for Product Evaluation**. If none is, the module is treated as **data-logging only**: check-out proceeds without a recipe, and no recipe has to exist for that equipment at all.

To run a module that collects data but does not grade it:

1. In **Configuration** → **Additional Configuration** → **Equipment & Processes** → **Attributes**, switch **Use for Product Evaluation** off on each attribute mapped to that module.
2. Leave **Required for Check-In** on (while evaluation is on it is forced on; switching evaluation off makes it editable). Values are collected whenever an attribute is required *or* used for evaluation, so the data keeps flowing to your reports and SPC while the PLC's own verdict decides the part.

Both flags belong to the attribute, not to the mapping (Chapter 2). Switching evaluation off applies everywhere that attribute is mapped. If one module must grade a characteristic while another only logs it, define two attributes rather than turning evaluation off for both.

The half-configured module

A module with evaluation attributes mapped but nothing usable to grade against is neither of the above: it is set up to judge parts, but cannot. **Check-out is refused**. The part is not worked, and the check-out result records *"No usable recipe for this equipment,"* which appears among the reasons on the Product Report.

Two states count as "nothing usable", and both refuse:

- the module has **no active control instance** at all; or
- it has an active control instance that carries **no value snapshot** — it has been released, but holds no limits, so there is still nothing to grade against.

The result message tells the two apart, so you can see which one you are looking at without decoding anything.

Why refuse rather than warn: a module that grades is the thing standing between a defect and the next operation. If it cannot grade, letting the part through produces a part that *looks* inspected and was not — which is worse than a stopped line, because it is discovered later and by somebody else.

To clear it, do one of two things: release a control instance for the module (and make sure the revision it points at actually carries values), or switch **Use for Product Evaluation** off on its attributes if the module was never meant to grade in the first place. A module with no evaluation attributes is data-logging only and never needs a recipe — that exemption is what keeps plants that do not use evaluation running normally.

Understand the recipe list

The list groups recipes by **Name and owning site** — in a multi-site system the same name can exist at several sites as genuinely different recipes, so each family shows its site next to the name (no site shown = a shared General recipe). Each revision beneath shows its **Revision, Description, Finished Product Material, Recipe Type, and Status (Pending)** — the approval status, followed in brackets by the status a pending change is waiting to move to. Search by name or description. The list hides **Obsolete** and **Archived** revisions until you tick **Show All Statuses**.

Recipes and sites — namespaces and shadowing

Each site has its own recipe **namespace**: a name (and its revisions) is unique *within* a site, so Site A and Site B can each own a "WIDGET-01" that evolves independently. Recipes of the **General** type belong to no site — they live in one shared namespace and are visible everywhere. Two rules make this predictable on the floor:

- **A site's recipe shadows a General recipe of the same name.** When a machine asks for a recipe by name, its site's own recipe wins over a General one — and when it asks for *the latest revision*, the site's whole family wins: the site's highest revision is used even if a General recipe of the same name carries a higher revision number. A site can therefore take local ownership of a shared recipe simply by creating its own version under the same name.
- **Pickers show what applies here.** The production check-out recipe list offers the current site's recipes plus General ones — another site's same-named recipe is never offered to this site's floor.

Editing is deliberately *not* site-restricted: recipe definitions live in the shared configuration database, so an engineer can maintain every site's recipes from one place. Instead of blocking, IntelliTrack shows a prominent **cross-site warning banner** on the edit and values pages whenever the recipe belongs to a different site than the one you are signed in to — you always know whose production you are about to change. General recipes never raise the banner.

The recipe approval lifecycle — and its effects

Recipes use the approval workflow they share with materials and BOMs (Chapter 4). A revision moves one step at a time: **Draft** → **Under Review** → **Approved** → **Active**, with **Active - Pending Effective Date** in between when an activation is scheduled, and three retired states — **Inactive**, **Obsolete**, and **Archived**. The **Status** list on the Edit page offers only the moves allowed from where the revision stands; the others are shown greyed out. The effects that matter:

- **Only an Active recipe is instantiated to equipment.** A Draft recipe cannot control production — this is the safety interlock between authoring and the floor.
- **Fields are editable only in Draft.** Once a recipe leaves Draft, the only change it accepts is a status transition — its name, material, values, and policy fields are locked. A recipe also **cannot be edited while an approval is pending**.
- **Promotion is gated by a readiness check.** Moving a recipe to **Approved** (or beyond) is blocked until it passes validation — every value used for evaluation has a limit, numeric limits are in order, value lists are complete, and calculated values are well formed — so a half-configured recipe cannot be approved into service. Use the **Validate** button on the Edit page to run the same checks on demand as a report *before* requesting approval; it also lists informational notices that never block.
- **System-managed statuses cannot be assigned manually.**
- Setting **Active** with a future **Effective Date** parks the revision in **Pending Effective** until that date — so a change can be approved now and take effect at a planned cutover.
- **A revision that has run cannot go back to Draft.** Once a control instance has been created from a revision, returning it to Draft is refused ("Create a new revision instead") — use **Duplicate** for that.
- **The recipe's material must be ready first.** To approve a recipe, its material must be Approved, Pending Effective, or Active; to activate it, the material must be Active, or scheduled to become Active no later than the recipe's own effective date.
- **Only a few revisions stay in circulation.** Activating a revision can demote older Active ones, and retired revisions can be archived automatically — see *How many revisions stay in circulation* below.

Who approves, and how many approvals a change needs

The forward moves — **Draft** → **Under Review** → **Approved** → **Active** — and bringing a retired revision back to **Draft** or **Under Review** are *approval steps*: the change is requested on the Edit page and waits as a **pending** change until enough people approve it. Everything else applies at once: retiring a revision (to Inactive, Obsolete, or Archived), moving it between the retired states, and sending a revision that is not retired back to Draft.

How many approvals a step needs is set by your site's **signature policies** (see [Signature policies](#)), not on the recipe — the policy for the act (for example *Recipe — Approve*) can differ by site, recipe type, material group, or for one named recipe, and it applies whether or not GxP Mode is on. The requirement is fixed

when the change is requested: changing the policy later does not alter a request already waiting. The recipe's **Required Recipe Approvals** field shows the count most recently applied. When your installation was upgraded to this release, each recipe's old count was carried into the policies — recipe types and individual recipes that differed from the rest got rows of their own, a recipe's own row marked *Migrated exception* — so no requirement changed on the day.

- If the policy does **not** require different people, your own request counts as the first approval — with a count of 1 the change applies immediately.
- If it **does** require distinct signers, the request is only a request: the requester is not offered **Approve**, and the full count must come from other people.
- Nobody's approval counts twice.

Act on a pending change

While a change is pending the recipe cannot be edited — its Details page drops the **Edit** and **Delete** links — and the banner at the top of **Details** names the change: the recipe's current status and the one it is waiting to move to. Below the banner, Recipe and Administrator users see the acts open to them, each with a **Comment**:

- **Approve Pending Status** — records your approval. It is not offered if you have already approved this change, or if you requested it and the policy requires distinct signers. The change applies when the last required approval arrives.
- **Reject Pending Status** — turns the request down; the recipe stays where it was.
- **Cancel Pending Status** — offered only to the person who requested the change, to withdraw it; the recipe stays where it was.

When no signature is needed the three share one form. When the policy asks for a signature, each opens as its own section with its own **Electronic signature** panel, and its button reads **Sign and approve**, **Sign and reject**, or **Sign and withdraw**. If someone else decided the change while you were reading, IntelliTrack tells you that nothing was changed.

Approval History (Recipe or Administrator role) lists every approval cycle — when it started and ended, the status it started from, the status proposed, and where it ended — and each act in it: who, when, and the comment. Once any act in the history was signed, an **Electronic signature** column shows the signature beside it.

Electronic signatures on recipe changes

With GxP Mode on, every step of the lifecycle is signed by default: submitting for review, including bringing a retired revision back to Under Review (meaning *Reviewed*); approving, activating, and retiring (*Approved*); and returning to Draft, or rejecting or withdrawing a pending change (*Rejected*, which needs a comment). A reason is optional unless your policy says otherwise. On the Edit page the panel appears for the status you select, and the save and the signature are recorded together — if either fails, neither is kept. Once a recipe has been signed, its Details page shows an **Electronic signatures** section listing each

signature, its meaning, and whether it still verifies. See [When you are asked to sign](#) — including [Which password you sign with](#), because not every way of signing in can sign — [Which acts can be signed](#), and [Seeing who signed a record](#).

How many revisions stay in circulation

Three system parameters (Chapter 17) keep a recipe family — every revision of one name at one site — from piling up. The two status caps act only when a revision's status changes, including when a scheduled activation takes effect:

Parameter	Default	What it does
61 — Max Active Recipe Revisions	1	How many revisions of one recipe may be Active at once. When an activation would exceed it, the oldest other Active revisions are set to Inactive ; the revision just activated always survives, even if its revision number is older. 0 = no limit. If you raise it, production uses the highest Active revision number.
60 — Max Inactive Or Obsolete Recipe Revisions	0 (off)	How many Inactive or Obsolete revisions are kept before the oldest are Archived . Only those two statuses are counted or archived, so archiving can never take a recipe out of production.
14 — Max Recipe Revisions	20	How many of a family's newest revisions keep their values in the site database. When the site's pruning runs, older revisions lose their values and can no longer run; the revisions themselves stay readable. 0 = keep everything.

Tick **Do Not Archive** on a revision to pin it: a pinned revision is never demoted, archived, or stripped of its values. It still counts toward the limits, so pinned revisions can hold a family above them.

Create a recipe

1. From **Product Definition** → **Recipes**, start a new recipe.
2. Enter the **Name** (unique within the owning site's namespace) and **Revision** (defaults to 1), a **Description**, and optionally the **Finished Product Material**.
3. Choose the **Recipe Type** and **Site**. A **General** recipe belongs to no site — it is shared everywhere and any Site selection is ignored. **Site** and **Master** recipes belong to the selected site (defaulting to the one you are signed in to), which is the namespace the name must be unique in. The system-managed **Control** type cannot be chosen manually.
4. Optionally tick **Do Not Archive** to pin this revision — for a revision you must keep runnable and referenceable indefinitely (see *How many revisions stay in circulation* above).
5. If your site controls engineering changes, choose the **ECO Pattern** and enter the **Engineering Change Order (ECO)**; once a pattern is chosen the ECO is required and must match it, so engineering-change numbers stay well-formed and traceable.
6. Save. The recipe is created in **Draft** — authored, but not yet controlling anything. Its **Required Recipe Approvals** and **Required Override Approvals** are filled in from the signature policies.

Warning: A duplicate name is rejected *within the owning site's namespace*. The same name at a different site — or as a shared General recipe — is a separate recipe family, not a conflict; see *Recipes and sites* above for which one a machine will get.

Define recipe values — these become the limits

From a recipe, open its **Values** (from the recipe list or its Details page) and add a value per process attribute you want to control: choose the process, then the attribute, then fill in what that attribute's data type needs (Recipe or Administrator role):

Attribute data type	What the value holds
Bool	Boolean Target — the state a good part must report.
Numeric	Low Alarm, Low Warning, Target, High Warning, High Alarm . The alarms are the pass/fail limits and the warnings the early-warning band; they must ascend ($\text{Low Alarm} \leq \text{Low Warning} \leq \text{Target} \leq \text{High Warning} \leq \text{High Alarm}$).
String	Match String Target , or Use Value List with the Allowed Values a part may report (Allow Any List Value accepts any entry on the list).
Curve	Nothing here — a curve has no single target. What a recipe can hold for a curve is a whole reference curve: see A nominal reference for a curve .

Fields that do not belong to the attribute's data type are cleared when you save, so a value can never carry a stray limit into the snapshot.

A value's **Value Source** is **Constant** (the figures you type) or **Calculated** — worked out from another attribute's value on the *context* product named when the control instance is created (on the web, the **Recipe Context S/N**) and then frozen into that instance's snapshot. Each calculated rule either rejects on failure — creating the control instance is refused when the context product cannot supply the base value — or falls back to a constant you supply. A calculated member has no figure of its own until then, so the **Values** list marks it **Calculated** where the figure would be; hover the mark to see its operator, source attribute and failure mode, or open the value's **Details** for the whole rule.

What this changes in the system: these values are exactly what get captured into the **value snapshot** when the recipe is instantiated, and therefore what every production measurement and audit reading is graded against, and what SPC charts draw as their specification lines. Editing a value here is editing the limit production will be held to — which is why values, like the rest of the recipe, are editable only while the revision is in **Draft**.

Value history. A value is never overwritten in place: saving a change, or restoring an old value, archives the value it replaces in the same step and records who superseded it. **Archive History** — from a value's Details, Edit, or Delete page — lists the current value and every earlier one, with who set each (**User**), when it was archived (**Date Archived**), and who replaced it (**Superseded by**). While the recipe is still

editable, a Recipe or Administrator user can **Restore Value(s)** from that list; the restored value becomes the current one and is credited to the person who restored it.

A nominal reference for a curve

A curve attribute has no single target, so a recipe value cannot hold one. What a recipe CAN hold for it is a whole curve: a **nominal reference**, the shape a good part makes. Every curve measured under a revision that holds one is then judged against it — the **Max Deviation From Nominal** metric is stored with the curve, the reference is drawn beside it on the curve report, and a comparison of such curves can be shown as **% of target**.

It is captured, not typed. A reference is taken from a curve a qualified part actually produced: open that curve on **Reports** → **Performance** → **PPD Curves by Serial Number**, then **Use as nominal reference** and choose this revision ([Chapter 12](#)). Drawing one by hand, importing one from a drawing and averaging one over a population are not offered.

It then belongs to the revision exactly as its values do: **one per curve attribute**, capturable only while the revision still accepts edits, copied into a new revision when the recipe is duplicated, and **frozen into each control instance** created from the revision — so a curve is always judged against the reference that was in force when it was measured, whatever the recipe says later.

Control instances — putting a recipe on a machine

A **control instance** is one recipe revision prepared for one equipment module, carrying its own frozen copy of the limits. **Control Instances** on the recipe list shows every instance made from a revision, with its status, grouped by **Campaign** — the run of a recipe on a workcell, named so operators can quote it — with each campaign's start, stop, and status (the campaign name is searchable). An instance prepared on the web and not yet run is listed as *Not part of a workcell campaign*; it joins a campaign when it is checked out or downloaded.

- **Create** (Recipe or Administrator role): choose the target equipment modules. The web creates each instance **In Review** (a machine's own download creates it **Requested** — but only while the equipment's **Allow Machine to Request Control Instances** switch is on; with it off, the machine is refused until an instance has been created and released here, see [Configure Equipment](#)). Only one open instance per equipment module and recipe is allowed — cancel, reject, or finish the existing one first. The optional **Recipe Context S/N** names the product that calculated values are worked out from; they are frozen into the instance's snapshot, and a reject-on-failure rule that cannot be resolved refuses the creation.
- **Review, then release.** A control instance prepared here must be **reviewed and released** before a machine will download it — this is the gate that stops a half-prepared recipe from controlling production. On an In Review instance, a Recipe or Administrator user selects **Release to Equipment** (a comment is optional), which makes it **Requested**, or **Reject** (a comment is required).

Cancel (a comment is required) withdraws an instance that is In Review, Requested, or Downloaded.

- **Web Check-Out does not wait for this review.** When an operator checks a product out on the web ([Production Execution](#)), the recipe they choose — one already approved for production — becomes the module's running recipe at once: IntelliTrack reuses the module's live instance of that recipe or creates one and makes it **Active** directly, applies any overrides that have finished their approvals, and closes every other live instance on the module (*Closed by web recipe change.*). An instance still In Review is not used by the web check-out. So the review step governs what machines download, not what an operator starts from the Check-Out page. Changing the recipe a module is already running is itself controlled: with **Allow Recipe Change** (system parameter 22) at 0, as shipped, only a Production Supervisor or an Administrator may do it; with 1, Production may too ([Who may change the recipe](#)).
- **Details** shows the instance's **Attributes**, **Overrides**, and **Deviations**; tick **Include Unapplied Overrides** to see the values as they will be once pending overrides apply. **Report** prints the instance. Opened from a Product Report, it shows the revision that was in force for that product, with a link to the latest; an instance already retired from the site database is shown read-only from the reporting archive.

Deviations and overrides — controlled change on a live instance

Deviations

A **deviation** records that a machine is running values different from its control instance. The machine's recipe monitoring detects it, and it is listed on the instance's **Deviations** tab, viewable by any signed-in user. A deviation moves **Detected** → **Acknowledged** → **Resolved**. It becomes **Superseded** when an applied override makes the recipe match what the machine is doing, and IntelliTrack resolves it on its own (recorded as *System Detection*) when the machine reports the expected values again.

On a deviation's details page:

- **Accept the deviation** (Recipe, Production Supervisor or Administrator role; a Detected deviation) acknowledges it — you have seen it and judged it. The floor's supervisor may accept one because the machine it describes is running on their line; resolving it stays with the recipe side.
- **Resolve the deviation** (Recipe or Administrator role; Detected or Acknowledged) closes it.
- **Add Comment** (Recipe, Production, Production Supervisor or Administrator role) adds to a running, time-stamped comment history that is never overwritten; a comment given with an accept or resolve is added the same way.
- **Create Override to Match Deviation** (Production Supervisor or Administrator role — requesting an override is theirs, below) starts an override pre-filled from it. The link is shown only to those roles.

With GxP Mode on, accepting (meaning *Deviation Accepted*, which needs a comment) and resolving (*Verified*) are signed by default, and the page then shows an **Electronic signatures** section. If someone else has already acted on the deviation, IntelliTrack tells you that nothing was changed.

Overrides

An **override** is a controlled change to one attribute's values on a live instance. A **Production Supervisor or an Administrator** requests one — it changes what a running machine enforces, so it is an act of authorization, not of operation — from the instance's **Attributes** tab or from a deviation (the **Create Override** links are shown only to those roles): change at least one value, choose a **Reason**, and add a **Comment** — required when the reason is *Other (explain in the comment)*. The reasons are a list your administrator maintains ([Reason codes](#)); a request made from a deviation starts with *Match detected deviation*.

How many approvals an override needs comes from the signature policy for *Control-instance override* — *Approve*, which can differ by the machine's site, area, workcell, or process, by material group or recipe type, or for one recipe; it is fixed when the override is requested. The requester never approves their own override, and nobody counts twice. If the policy does not require distinct signers, the request counts as the first approval — with a count of 1 the override is approved as soon as it is requested.

On the override's details page:

- **Approve** — a user in the **Recipe or Administrator** role, other than the requester, can approve until the count is met (your signature policy can require a particular role). The recipe side approves a change to a running machine; the floor asked for it.
- **Reject Pending Status** — a Recipe or Administrator user other than the requester can turn it down before it reaches the machine.
- **Cancel Pending Status** — the requester can withdraw it before it reaches the machine.
- **Revert to the master recipe** — **Production, Production Supervisor or Administrator** role; offered on an override that has been applied and is still the latest change to its attribute. Going back to the approved master recipe is always safe for the floor to do. It creates an already-approved override that puts the master recipe's values back (a reason is required), applied like any other; the original is marked **Reverted**.

An approved override is **applied automatically**: for machine-connected equipment when the machine confirms it is running the new values, and for web-run equipment at the next web check-out. Applying it **mints a new snapshot revision**, so from that point the machine is graded against the new limits, while every earlier unit still points at the revision it was actually made under. Until it is approved, an override changes nothing — neither the next download nor the grading — and a machine already running its values shows a deviation. An override's status moves **Requested** → **Approved** → **Issued**, or ends **Cancelled** or **Reverted**.

With GxP Mode on, every act on an override is signed by default: the request (meaning *Authored*; a reason is required), each approval (*Approved*), a rejection or withdrawal (*Rejected*), and a revert (*Approved*;

a reason is required) — see [Which acts can be signed](#). A signed override shows an **Electronic signatures** section on its details page, and its **Approval History** shows each act with its signature.

Warning: Overrides change what the equipment enforces. That is precisely why they are change-controlled: give a clear reason and route them for the required approvals.

Duplicate, edit, delete, and report

- **Duplicate** copies a recipe (runs `recipe.Copy`) — its values, value lists, and calculated-value rules — into a new **Draft**: either the next revision of the same recipe (**New Recipe Revision**, proposed revision pre-filled) or a **New Recipe**, optionally pointing at a different material. The copy always lands in the **same site's namespace** as the source, and its audit-trail entry names you, your roles, and the site. Use it to iterate without disturbing the active revision.
- **Edit** changes recipe fields (Draft only) and requests status transitions (subject to the lifecycle rules; with GxP Mode on, a transition may need your signature). The **ECO Pattern** can only be changed by an Administrator or while the recipe is the only revision in its family.
- **Delete** soft-deletes the recipe (renamed and hidden, not physically removed).
- **Report** (Recipe or Administrator role) opens a printable recipe report listing every value by process and attribute — the document you hand an auditor to show what the limits are.

GTINs, ECO keys, and attribute maps

GTINs and ECO keys are managed by **Administrators**; attribute maps, which belong to the plant and equipment model, by **Engineering** or **Administrators**. A GTIN's and an attribute map's details can be viewed by any signed-in user:

- **GTIN** — Global Trade Item Numbers assigned to a material.
- **ECO Keys** — named regular-expression patterns that validate the **ECO** value on a recipe (and on materials and BOMs), keeping change-order references consistent. Open them from **Configuration** → **Additional Configuration** → **Quality & Change Control** → **ECO RegEx Definitions**.
- **Attribute Maps** — the per-equipment mapping of process attributes: which equipment data points feed which attribute, and in what position. The map says *where* an attribute is collected; whether it is graded is **Use for Product Evaluation** on the attribute itself (Chapter 2). Together they decide whether a module needs a recipe at all — see [Running without recipes](#) above.

Field Reference

Recipe (create/edit):

Field	Required?	What it does / valid values
Name	Yes	Recipe name; unique within the owning site's namespace (General recipes share one site-less namespace).
Revision	Yes	Revision identifier (defaults to "1"). How many revisions of one recipe may be Active at once is System Parameter 61 (default 1).
Description	No	Free-text description.
Finished Product Material	No	The finished-product material this recipe makes; only Active recipes control production. The material must be Approved, Pending Effective, or Active before the recipe can be approved, and Active before it can be activated.
Recipe Type	Yes	General (shared, no site), Site , or Master ; the Control type is system-managed and not selectable.
Site	For site-bound types	The owning site for Site/Master recipes (defaults to the current site); ignored for General recipes. Decides the naming namespace and which floor the recipe shadows for.
Status	Edit only	The approval status. Offers only the moves allowed from the current status; a new recipe always starts in Draft.
Do Not Archive	No	Pins this revision: it is never demoted by the Active limit, archived, or stripped of its values.
ECO Pattern / Engineering Change Order (ECO)	Conditional	Choosing a pattern makes the ECO required, and it must match the pattern. Only an Administrator can change the pattern once the recipe has other revisions.
Effective Date	No	Future date to auto-activate (parks in Pending Effective).
Required Recipe Approvals / Required Override Approvals	Read-only	Filled in from the signature policies when the recipe is created (Required Recipe Approvals again each time an approval step is requested). What a change actually needs is decided by the policy; change it under Signature Policies (Chapter 20).
Recipe values	—	Per-attribute targets/limits; captured into the snapshot and used to grade production and audits, and drawn as SPC specification lines. Editable in Draft only.

Related Topics

- [Plant & Equipment Model](#) — processes & attributes
- [Materials & Suppliers](#) — the finished material & approval lifecycle
- [Production Execution](#) — where measurements are graded against the recipe
- [Product Audits](#) — audits graded against the same snapshot
- [Reports](#) — SPC and quality reporting

- [System Configuration](#) — system parameters 14, 60, and 61
- [Electronic Signatures & Audit Trail](#) — signature policies, reason codes, and how to sign

Chapter 6: Bills of Material

Overview

A **bill of material (BOM)** lists the components that go into a finished product and the rules for consuming them. IntelliTrack supports **static** BOMs (a fixed list of component materials with quantity ranges, every one of them enforced) and **dynamic** BOMs (specific produced→consumed product pairs loaded from a file, each with a strict flag). This chapter covers creating and maintaining both — and, crucially, what a BOM *does* during production.

Why this matters: a BOM is only valuable if it is *enforced*. In IntelliTrack the BOM is the **plan** that the production gate checks the **actual** consumption against, in real time, at the moment of build. A rule will **refuse** a build that doesn't satisfy it, and every refusal is recorded in the product's enforcement log. That is the difference between a BOM that prevents the wrong component going in and one that just documents what should have.

Before You Begin

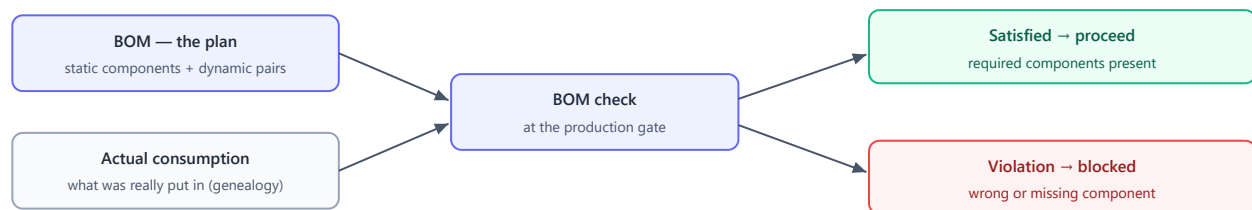
- **Viewing** the BOM list, a BOM's details, and its **Composition** is available to **any signed-in user**. The static-item pages, **Approval History**, and the dynamic BOM upload need the **Recipe** or **Administrator** role.
- **Creating, editing, duplicating, approving, and deleting** BOMs and items requires the **Recipe** or **Administrator** role.
- **Approval counts are not set on the BOM**. How many approvals a status change needs comes from your site's **signature policies** (Chapter 20); the count on the BOM form is filled in from them and cannot be edited.
- With **GxP Mode** on, moving a BOM through its lifecycle — and approving, rejecting, or withdrawing a pending change — is **electronically signed** by default. See [Electronic Signatures & Audit Trail](#).
- Define the finished-product **material** and the **component materials** first (Chapter 4).

How to Get Here

Product Definition → **Bills of Material** opens the BOM list. Each BOM's **Composition** is its starting point — the full structure, with links to the BOM's **Details**. Static items are added from the BOM's **Edit** page (**Add Item**). The **Dynamic BOM Upload** page has no menu entry: open it at `/DynamicBom/Upload`.

BOM vs. genealogy — plan vs. record, and the check between them

A BOM is the plan — the production gate enforces it



Every refusal is recorded in the product's BOM enforcement log (which operation, whether it blocked, the message, source/target serials).

Static items enforce a component material; dynamic-BOM rows enforce specific produced→consumed serial pairs.

A BOM is the **plan** — what *should* be consumed. **Genealogy** (Chapter 3) is the **record** — what *was* consumed. The production gate compares them at build time; a refusal lands in the product's [BOM Enforcement Log](#).

The static check uses the **effective** revision of the product material's BOM and ignores a deleted one. A material has one BOM, whose revisions stand side by side; the effective revision is the **Active** one or, while none is Active yet, the newest. So a first BOM is enforced whatever its status: items saved on it in Draft are enforced as soon as the change reaches the site. Build and approve a BOM before its material goes into production — while that first BOM is in Draft, every change to its items reaches the floor. A new revision of a BOM that has an Active revision governs nothing until it is activated.

When a component is refused

The check asks the **static** BOM first (is this material on the list, and would this take its running total past the maximum?) and then the **dynamic** BOM (is this exact unit paired with this product, and would this take the pair past its maximum?). The refusal is reported by the half that made it, in that half's terms:

- A **static** refusal names the static item and its range, and **Existing** is the total of that material already in the product — every unit of it, not just the one being scanned — because that total is what the maximum limits. A material the static BOM does not list shows a range of 0..0. A static refusal **always blocks**, whatever dynamic pairing the unit has: the static list is what stops a part that looks right and fits from being built in unless it is specified.
- A **dynamic** refusal names the pairing and its range, and **Existing** is what this one unit has already contributed. A unit the dynamic BOM does not pair with this product shows 0..0 and is blocked. A strict pairing blocks when its maximum would be passed; a non-strict pairing does not limit quantity, so the most it can do is record a warning while the build goes ahead.

Why it matters: each refusal is filed under the rule that made it, so a static item's page lists only the refusals *that item* made, and the figures in the message add up to the decision — an investigation starts at the rule that refused, not at a neighbouring one the part satisfied.

When a product is checked in at a station

The static BOM is checked again when a product is checked in. At each station, the static items whose **Process Applied At** is that station's process must be satisfied — at least their **Min. Amount** and no more than their **Max. Amount**, counting only components recorded at that process's equipment. If they are not, the part is **rejected**, and *Incomplete BOM* appears as the reject reason on its Product Report. It is a completeness failure, not a defective part, so it is never scrapped automatically (Chapter 8).

An item with no process is not part of any station's check: it is enforced as components are added (whether the material is listed at all, and its maximum), but no station checks its minimum. Give an item a process when its minimum matters.

At check-out: the dynamic BOM context

On the web check-out page, an operator can enter a **Dynamic BOM Context S/N** — the assembly the part is about to be built into (Chapter 8). Check-out is then refused unless the dynamic BOM pairs this exact part with that assembly. This admission check asks only whether the pairing exists; the Strict flag, the amounts, and the process are left to the moment the component is actually added. Leave the box blank and the check is skipped — as it is for check-outs from equipment, whose standard interface does not supply a context serial.

The BOM approval lifecycle

BOMs use the approval workflow they share with recipes and materials (Chapter 4), one step at a time: **Draft** → **Under Review** → **Approved** → **Active**, with **Active - Pending Effective Date** when a future **Effective Date** is set on the move to Active, and the retired states **Inactive**, **Obsolete**, and **Archived**. A new BOM always starts in **Draft**; its status is changed from the **Edit** page, whose **Bill of Material Status** list offers only the moves allowed from where the BOM stands. The effects that matter:

- **Only Draft is editable.** Outside Draft the only change a BOM accepts is a status transition — its header is locked and its items can no longer be added, changed, or deleted.
- **Forward moves need approvals.** How many comes from the signature policy for the act (for example *Bill of materials* — *Approve*), fixed when the change is requested, and applied whether or not GxP Mode is on; your own request counts as the first approval unless the policy requires distinct signers. The rules are the same as for recipes — see [Who approves, and how many approvals a change needs](#).
- **The material must be ready first.** To approve a BOM, its material must be Approved, Pending Effective, or Active; to activate it, the material must be Active, or scheduled to become Active no later than the BOM's effective date.

- **An enforced BOM cannot go back to Draft.** Once the BOM has been enforced in production (it has entries in the enforcement log), returning it to Draft is refused.
- **Activating a revision switches enforcement to it.** The previous Active revision becomes **Obsolete** in the same step. Otherwise the status does not switch enforcement on or off — see *BOM vs. genealogy* above.

While a change is pending the BOM cannot be edited, and the banner at the top of its **Details** page names the change — the BOM's current status and the one it is waiting to move to. Below it, Recipe and Administrator users act on the change exactly as for a recipe — **Approve Pending Status**, **Reject Pending Status**, or (the requester only) **Cancel Pending Status**, each with a comment; see [Act on a pending change](#). **Approval History** (Recipe or Administrator role) lists every cycle and act.

With GxP Mode on, each lifecycle step and each of those acts is signed by default, with the same meanings as for recipes: *Reviewed* for submitting for review; *Approved* for approving, activating, and retiring; *Rejected* for returning to Draft and for rejecting or withdrawing a pending change (see [Which acts can be signed](#)). A signed BOM shows an **Electronic signatures** section on its Details page, and its Approval History shows each act's signature.

Create a static BOM

1. From **Product Definition** → **Bills of Material**, create a new BOM.
2. Enter **Name** and **Revision**, a **Description**, and the finished-product **Material**. If your site controls engineering changes, choose the **ECO Pattern** and enter the **Engineering Change Order (ECO)** — the same rule as for recipes (Chapter 5).
3. Save — the BOM is created in **Draft** and its Details page opens. Its approval count is filled in from the signature policies.

Warning: A BOM's **Name** must not belong to any other BOM, and **each material can have only one BOM** — its revisions aside, a second BOM with the same name or for the same material is refused.

Add component items

Items are added from the BOM's **Edit** page: **Add Item** appears there while the BOM is in Draft, the only status in which its items can be added, changed, or deleted.

1. Select the **Component** material (the BOM's own material is not offered), its **Units**, and optionally the **Process Applied At**.
2. Set the quantity range (**Min. Amount** / **Max. Amount**). Every item is enforced at the production gate: leave a bound blank to skip that side of the check (no Min = no minimum required, no Max = no upper limit), and set Min = Max to require an exact quantity. A Min of 0 is saved as blank. An item with both bounds blank simply allows the material with no quantity constraint.

3. Save; repeat per component. A BOM holds one item per component material and process.

Why two of the fields matter:

- **Process Applied At** decides where the item's quantity is checked when the product is checked in — see *When a product is checked in at a station* above. Leave it at -- **Any** -- and the item is enforced only as components are added.
- **Units** lists only units that can hold a quantity of stock — units whose dimension is marked as an inventory dimension and that have a conversion ([Chapter 17](#)). Any other unit is refused, but an existing item keeps the unit it was saved with.

A static item's own page (Recipe or Administrator role) shows its **Genealogy usage count** — how many times the component has been recorded in products of the BOM's material — and the item's 20 most recent BOM enforcement events.

See a BOM's full composition

Composition — the action on every row of the BOM list — shows the whole multi-level structure: each component and, where a component is itself made to a BOM, that BOM's components beneath it, level after level. Use it to answer *what really goes into this product?* before approving or changing a BOM, because a change to a sub-assembly's BOM reaches every assembly built from it.

- **Used In** names every BOM that consumes this BOM's material, so you can see what a change reaches upward as well.
- Select a node to see its details — quantity range, units, process, and the nested BOM with its status. For Recipe and Administrator users an item node also offers **Edit** and **Delete**, which work only while that BOM is in Draft.
- A nested BOM that is Obsolete or Archived is shown but not expanded until you tick **Show All Statuses**.
- **List Format** shows the same structure as a paged table with a **Level** column and a **Max Depth** setting that is remembered for you; **Tree Format** returns to the tree.
- *Circular reference detected* means a material appears in its own ancestry — a data problem to correct. A structure too deep or too large to display is cut off with a warning.

Create a dynamic BOM

A dynamic BOM defines allowed **produced** → **consumed** product pairs (down to the serial number) — for when the valid components are specific serialized units rather than a material in general.

There is **one rule per produced/consumed pair**. A pair is the rule's identity, so re-uploading a pair you have already loaded *updates* that rule rather than adding a second one — which is what makes the upload safe to re-run whenever the build changes, and why you never have to clear the old rules out first.

1. Open the **Dynamic BOM Upload** page (/DynamicBom/Upload ; Recipe or Administrator role).
2. Prepare a **CSV or JSON** file: source (produced) serial, target (consumed) serial, a strict flag, and min/max amounts. In CSV the columns are `sourceSerialNo,targetSerialNo,strict,minAmount,maxAmount` , the last three optional; a header line beginning `source` and lines starting with `#` are ignored. JSON is an array of objects with the same property names. **Strict** is on unless the file says `false` .
3. Optionally fill in **Process (optional)** to scope the rules in this file to one station — with the process's identifier (a GUID), not its name. An entry that cannot be read as an identifier, or that names no process, stops the upload with nothing imported. Leave it **blank** and the rules apply at *every* process — that is the permissive default, and it is what all existing rules use. Name a process and the rules apply at *that station only*: elsewhere the pairing is invisible, so it neither permits nor refuses anything. Because a pair has only one rule, the process is a property *of* that rule, not a way to load two rules for the same pair — re-uploading the pair with a different process **moves** it, and re-uploading with the box blank returns it to applying everywhere.
4. Upload. IntelliTrack adds or updates one rule per valid row and reports how many imported and how many invalid rows were skipped. A row is skipped when a serial is blank or unknown, when source and target are the same product, or when the amounts cannot be stored (a min of zero or less, or a min above the max). A skipped row costs only itself — the rest of the file still imports. A database problem that is not about the row itself (a timeout, say) stops the upload and is reported as such, rather than being counted as a skipped row.

Duplicate, edit, and delete a BOM

- **Duplicate** copies a BOM header and all its static items into a new **Draft** BOM, as a **New Bill of Material** (a new name, and the material it is for) or a **New BOM Revision** (the same name and material, next revision). A new BOM must pass the same rules as one created from scratch — its name unused and its material without a BOM. A new revision is created beside the one it copies and governs nothing until it is activated; activating it makes the previous Active revision **Obsolete**.
- **Edit** changes the header (Draft only) and requests status transitions (blocked while a change is pending; with GxP Mode on, a transition may need your signature). Items can be added, changed, or deleted only while the BOM is in Draft.
- **Delete** — a BOM past Draft is marked **deleted but retained** for history and disappears from the list; a **Draft** BOM is removed outright. Delete a Draft BOM's items first: deleting a Draft BOM that still lists items fails.

Field Reference

BOM header (create/edit):

Field	Required?	What it does / valid values
Name	Yes	BOM name; must not belong to another BOM.
Revision	Yes	Revision identifier.
Description	No	Free-text.
Material	Yes	Finished-product material; a material can have only one BOM (its revisions aside). It must be Approved, Pending Effective, or Active before the BOM can be approved, and Active before it can be activated.
Bill of Material Status	Edit only	The approval status. Offers only the moves allowed from the current status; a new BOM always starts in Draft.
Required Material Approvals	Read-only	The approval count, as the BOM form labels it. Filled in from the signature policies when the BOM is created and again each time an approval step is requested; what a change actually needs is decided by the policy (Chapter 20).
Effective Date	No	Future date to auto-activate (parks in Pending Effective).
ECO Pattern / Engineering Change Order (ECO)	Conditional	Choosing a pattern makes the ECO required, and it must match the pattern. Only an Administrator can change the pattern once another BOM shares the name.

Static BOM item:

Field	Required?	What it does / valid values
Component	Yes	Component material to consume; the BOM's own material is not offered. One item per component and process.
Min. Amount / Max. Amount	No	Enforced quantity range; a blank bound skips that side of the check, Min = Max requires an exact quantity. A Min of 0 is saved as blank.
Units	No	Unit for the quantity; only units that can hold a quantity of stock are offered.
Process Applied At	No	The process whose check-in checks the item's range. -- Any -- (blank) means the item is enforced only as components are added.

Dynamic BOM upload row:

Field	Required?	What it does / valid values
Source / Target serial	Yes	Produced and consumed serials (must exist). Together they identify the rule, so re-uploading a pair updates it.
Strict	No	Whether the pairing's Min / Max quantity is enforced; on unless the file says <code>false</code> . Off, the pairing still decides <i>which</i> unit may go into the product, but any quantity of it is accepted. It never relaxes the static BOM.

Field	Required?	What it does / valid values
Min / Max amount	No	Allowed quantity range. Min must be greater than zero, and may not exceed Max.
Process	No	Set once for the whole file, on the upload page rather than in it, as the process's identifier. Blank means the rule applies at every process; a named process narrows it to that station, where the rule is simply not seen elsewhere. Re-uploading a pair overwrites it, and leaving the box blank returns the rule to applying everywhere.

Related Topics

- [Products](#) — genealogy and the enforcement log
- [Materials & Suppliers](#) — the materials a BOM lists
- [Recipes](#)
- [Production Execution](#) — where the BOM is enforced
- [System Configuration](#) — units of measure and their dimensions
- [Electronic Signatures & Audit Trail](#) — signature policies and how to sign

Chapter 7: Work Orders

Overview

A **work order** is IntelliTrack's link between *demand* and *what actually gets made*. It records an intent to produce a planned quantity of a particular material — usually for a customer, on a particular line — and then keeps a live count of how much has been **started** and **completed** against that intent as production happens.

Why this matters: without a work order, production is just a stream of units with no answer to "how much of *this order* is done, and are we on track?" The work order is what turns individual check-ins on the floor (Chapter 8) and at the machines into a fulfillment number you can schedule against, dispatch to equipment, report on, and close out for a customer. Crucially, work orders are not just a screen — the **machines themselves** read and drive them, and the **system** advances them automatically as good product is made.

Before You Begin

- **Viewing** work orders is open to **any signed-in user** so supervisors and operators can check progress.
- **Starting and stopping** an order (**Start**, **Stop**) take the **Production** role: they record what the floor is doing with an order that has already been authorized.
- **Creating, editing, deleting and rebalancing** an order, and **every other status change** — **Return to Draft**, **Schedule**, **Release**, **Complete**, **Reopen**, **Cancel** and **Close** — take the **Production Supervisor** or **Administrator** role (a production supervisor can Start and Stop too). These authorize and size production: they change what the floor is allowed to build and how fulfillment is counted. The **Details** page shows you only the buttons you may use, and an administrator decides who holds each role ([Security and Sign-In](#)).
- If the order names a **material** to produce, that material must already exist (Chapter 4). Naming one also focuses the counters — see *Assign products* below.
- Four system parameters change how strict work orders are — see *Quantity enforcement and required work orders* below. Each is a site-wide default that an engineer or administrator can override for an individual equipment module on the **Equipment enforcement** page (**Engineering** or **Administrator** role), so confirm how your site (and the station in question) has them set.
- Where your site uses electronic signatures (GxP Mode — see [Electronic Signatures & Audit Trail](#)), releasing or closing an order, or changing a released order's quantity beyond its tolerance, can ask for your signature — see *Sign a release, a close, or a quantity change* below.

How to Get Here

Production → **Work Orders** opens the work-order list (titled **Work Orders**).

Find a work order and read its progress

When you open the list you are usually answering "what is open right now?" or "where does order *X* stand?"

1. The list shows work orders **newest first**.
2. **Search** by number, description, or **Group** to pull up a family of related orders. (Group exists precisely so you can batch related orders — by line, customer, or campaign — and find them together.)
3. **Filter by status** to answer "what is still running?" or "what is waiting to be released?"

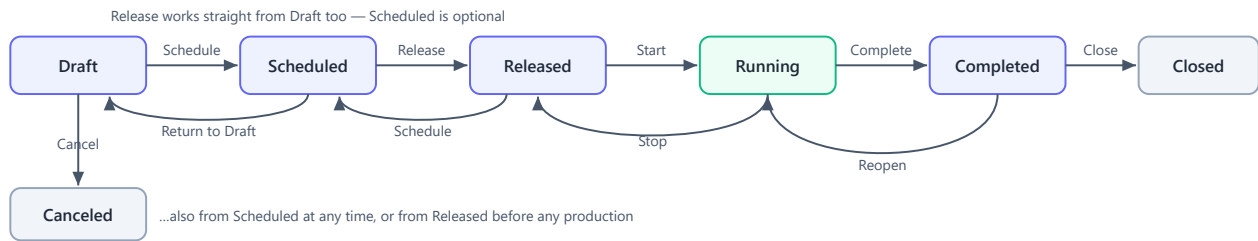
Open a work order to see its **Details**: the **Planned**, **Started**, **Completed**, and **Rejected** quantities, a **progress percentage** ($\text{Completed} \div \text{Planned}$, capped at 100%), and — once anything has reached a final disposition — a **yield percentage** (good output as a share of everything finished, good plus rejected). An order with an overbuild tolerance also shows the **Maximum** it may build (see *Quantity enforcement* below). These numbers are the point of the work order — they are maintained automatically (see below), so the Details page is always the current truth about fulfillment without anyone updating it by hand.

Below the numbers, the Details page shows *why* they are what they are: the order's products grouped by status (**Good**, **Rejected**, **In Process**), each serial number linking to that product's own page (Chapter 3); the units it **handed off** to other orders with the credit it kept for each (see *Move a product to another work order* below); its **Lifecycle History** — every status change, edit and recount, who made it (a person, a machine, or an integration) and, where an electronic signature authorized it, that signature's meaning and signer; and its **Membership Changes** — every product moved onto or off the order by hand, and by whom. When a release, a close or a quantity change on the order was signed, an **Electronic signatures** section at the top of the page lists each signature (see *Sign a release, a close, or a quantity change* below). The **Report** link opens a printable **Work Order Report** of the order's numbers, products and history.

The lifecycle — and who drives each change

A work order's **status is a gate**, not just a label: it controls whether the floor may build against it, whether a machine will offer it, and how it behaves in reporting. The diagram shows every state, the action that moves between them, and which of the three actors — **you**, a **machine**, or the **system** — can trigger it.

Work order lifecycle — and who drives each change



You — the web UI

Buttons on the Details page move the order through every transition above: Schedule, Release, Start, Stop, Complete, Reopen, Close, Cancel, Return to Draft.

Edit the header (locked once production starts) and Rebalance the counters.

Release, Close and a quantity change beyond tolerance may need a signature.

A machine — PLC / AOI

Fetches its dispatch queue — only Released/Running orders planned for this station's hierarchy, or planned for no resource at all.

Drives Start, Stop, Balance, Close and Cancel via WorkOrderEvent — the same state machine as the web buttons. (Modify records an audit marker.)

The system — automatic

First product produced against a Released order auto-starts it (→ Running) and counts Started. Good output raises Completed, scrap raises Rejected; in Block mode the order auto-completes at its plan. Expired orders are pruned once empty and no longer dispatchable.

What each status means in practice:

Status	What it means on the floor
Draft	A plan in progress. Nothing is committed; you can still edit every field. Machines do not see it.
Scheduled	The plan is firm and slotted, but you are not yet inviting production. Visible for planning; still not dispatchable.
Released	Dispatchable. This is the status the floor and the machines look for — releasing is the act that says "you may now build this."
Running	Production has started against it. The system moves the order here on the first unit, so "Running" reliably means "live on the floor right now."
Completed	Production finished (reached automatically in Block enforcement mode — see below). No <i>new</i> product can be introduced to it, but parts already started against it may finish their routing (each such finish is logged). Not yet archived, so you can still review, correct, or Reopen it.
Canceled	Stopped without completing. Terminal — kept so the record of <i>why</i> it ended is never lost.
Closed	Finished and formally closed out. Terminal.

The actions offered on the **Details** page depend on the current status, which is how IntelliTrack prevents illegal jumps (you cannot complete an order the floor never started) — and on your role: a Production user is offered only **Start** and **Stop**, because running an authorized order is operator work, while authorizing, sizing and ending it is a supervisor's. Note one deliberate rule: once any product has been produced against a **Released** order, **Cancel disappears** — you protect production history by completing rather than abandoning a run that already made parts.

How machines use work orders

This is what makes a work order more than a planning note. Equipment running the IntelliTrack **AOIs** (the PLC code, available from the Help center) talks to IntelliTrack directly:

- **Dispatch queue.** A machine asks IntelliTrack for its work-order queue and is offered **only dispatchable orders** — those in a status that is *OK to work* (Released or Running). The **Planned Resource** field is a dispatch *filter*, not a requirement: an order whose Planned Resource names the equipment — or any parent level of its plant hierarchy (workcell, area, site, enterprise) — is offered to that station, and an order that names **no resource at all is offered to every station** (it is unplanned, so any station may take it). Only an order planned for a *different* resource is filtered out, which is how the field routes the right orders to the right line. An order still in Draft or Scheduled never appears at a machine.
- **Machine-driven lifecycle.** The equipment can **Start, Stop, Balance** (recompute counters), **Close**, and **Cancel** the order from the floor, and can record a **Modify** audit marker (an event note only — it changes no status). These run through the **same state machine** as the web buttons — there is one source of truth, so an order started at the PLC behaves identically to one started in the browser. A machine **Close** on a Running order completes it first, then closes it. These are the machine's own acts, recorded under its name in the order's **Lifecycle History**; they never ask for an electronic signature, even where a person's **Close** does.
- **The station remembers its order.** Every reported production also records which order that equipment module is running. With **Bind Work Order To Equipment** (parameter 92) enabled for the station, a product presented at the station *without* a work order is attributed to the station's current order — so a PLC restart, or a part introduced without its order, does not become untracked production. The binding is made and released automatically: the moment the order stops being workable (completed, canceled, closed, or pulled back to Draft/Scheduled) every station bound to it is released, so a stale binding can never attribute the next shift's product to a finished order. Leave the parameter off for stations that run mixed product, where each part carries its own order or none at all.
- **One order, one recipe.** With **Work Order Recipe Enforcement** (parameter 93) enabled for the module, the first production against the order at each module **pins** a control recipe — the order's own recipe when its header names one, otherwise whatever that module was running at the time. Every later part at that module is compared against the pin, *including after a stop, complete, and reopen*, so the two halves of a reopened run are always built the same way. A mismatch is logged (**Warn**) or refused (**Block**).
- **Starting an order can trigger an audit.** If the order's process is set to **Trigger at Work Order Start**, the machine's **Start** event marks an audit as due on that equipment, which asks the machine to pull a sample for QA (see [Product Audits](#)). On its own this never holds the line. It is also what first starts audit tracking on a module: until a work order has started there on such a process, no audit comes due on it at all.

Create a work order

You create a work order whenever you have a commitment to produce a known quantity and want IntelliTrack to track and dispatch it. The form has two sections — **General Information** and **Planning & Scheduling**.

1. From **Work Orders**, create a new one (**Create a Work Order**; Production Supervisor or Administrator).
2. **General Information:**
 - **Work Order Number** — the order's unique key, used by operators, the machines, and any ERP integration. Use your real order number; it is *not* editable after creation.
 - **Description** and **Group** (optional). Group is your handle for finding related orders later.
 - **Customer** and **Material** to produce.
3. **Planning & Scheduling:**
 - **Planned Quantity** — the target the progress percentage is measured against (must be ≥ 0).
 - **Overbuild Tolerance (%)** — how far past the planned quantity production may run, as a percentage (0 to 1000). This raises the ceiling that **Block** enforcement mode gates on — see *Quantity enforcement* below. Leave it at 0 to cap the order at exactly its plan.
 - **Unit of Measure** — only units that can hold a quantity of stock are offered, because the order's quantity is reconciled against what production actually made. A unit that cannot (a temperature, say, or a unit nobody has classified yet) is refused — see [Units of measure](#).
 - **Planned Resource** — where the order is dispatched. Name a real plant-model resource (a module or any parent level) to route the order to those stations, or leave it **blank to offer the order to every station**.
 - **Planned Start** and **Required By** — the schedule dates the floor and reporting use.
 - **Expiry Date** — how long the order remains valid before it becomes eligible for automatic pruning. Leave it blank and IntelliTrack sets it to the creation date plus the **Work Order Expiry Time** (parameter 13, 365 days unless your site has changed it) — see *Retention* below.
4. Save. The order is created in **Draft** with its counters at zero; nothing on the floor can build against it until you **Release** it.

Warning: The **Work Order Number** must be unique — a duplicate is rejected with "A work order with this number already exists." Because it is the key the floor and integrations reference, it cannot be changed later.

Edit a work order — and what locks

Open the order and choose **Edit** (Production Supervisor or Administrator). You can change the header while the order is **not** terminal (Canceled or Closed). One important lock: **once production has started**

(Started Quantity > 0), the Material can no longer be changed — the parts already built define what the order is, so IntelliTrack freezes it to keep genealogy and fulfillment honest.

Three more rules apply when you save:

- The **Unit of Measure** must be one that can hold a quantity of stock, as on create — although an order already saved in another unit keeps it.
- Leaving **Expiry Date** blank does not remove the date: the order keeps the one it has.
- On an order that has been released, changing the **Planned Quantity** by more than its **Overbuild Tolerance (%)** allows, or changing the tolerance itself, is a signed change where your site's policy requires one — see the next section.

Sign a release, a close, or a quantity change

Three work-order decisions — all of them a production supervisor's or an administrator's — can be made **signed acts**: **releasing** an order to production, **closing** it, and **changing how much a released order may build** beyond its tolerance. Releasing is the moment a plan becomes something the floor may build; closing ends the order's production record; and a large change to a released order's quantity changes what was released. Where a site's quality system treats these as approvals, the signature records who made each decision, when, and exactly what they approved.

All three ship **unsigned**. They are signed only when **GxP Mode** is on and your site's signature policy for the act requires a signature — an administrator turns that on under **Configuration → Additional Configuration → Compliance → Signature policies** (see [Electronic Signatures & Audit Trail](#)). As shipped, the signature's meaning is **Approved** and a reason is optional.

- **Release and Close.** On the Details page, a button whose act needs a signature carries its own **Electronic signature** panel. Type your password (and choose a reason if you want to give one), then select **Release** or **Close**. The signature and the status change are saved together: if either is refused, neither happens (a refused signature is explained in that button's panel). The signature covers the order as the page showed it, so if someone changed the order in the meantime it is refused — reload the page and sign again.
- **A quantity change beyond tolerance.** Once an order has been released (Released, Running or Completed), an edit needs a signature when it moves the **Planned Quantity** by more than the order's own **Overbuild Tolerance (%)** allows — any change at all when the tolerance is 0 — or when it changes the tolerance itself. For an order of 100 with a 10% tolerance, a new plan anywhere from 90 to 110 needs no signature; 89 or 111 does. The first **Save** saves nothing: the page says *This order has been released, and the change moves its quantity beyond its tolerance, so it needs your electronic signature*, and **Show exactly what is being signed** lists the order and every field your edit changes, before and after. Sign, then select **Save** again. Before release the quantity moves freely — releasing is the act that is signed, over the quantity the order then carries.

What this changes in the system:

- The Details page lists the order's signatures in its **Electronic signatures** section, and the **Lifecycle History** shows, on the event each signature authorized, its meaning and signer (for example *Approved by* and the signer's name). How to sign, and what each refusal means, is in [Electronic Signatures & Audit Trail](#).
- Only a person's own act is signed. A machine's **Start**, **Stop**, **Close** and **Cancel**, and IntelliTrack's automatic changes (the first product starting an order, **Block** mode completing it), never ask for a signature.
- The work-order API cannot sign, so it is refused any of these acts while your policy requires a signature for it — see *Bulk scheduling and the work-order API* below.

Assign products — and the counters that move automatically

Products are tied to a work order from the **product** screens, not here, which matches how work flows (you scan a part, then say which order it belongs to):

- When **creating or editing a product**, choose the **Work Order** (Chapter 3). Giving an order to a product that has none is Production work; creating a product by hand, and moving a product that already has an order (below), are a Production Supervisor's.
- When **receiving** material (Production role), choose the **Work Order**.

Both lists offer only orders the floor may build against — **Released** or **Running** — so an order must be released before anything can be assigned to it here.

What this changes in the system:

- Associating a product **counts it against the order**: its amount is added to **Started Quantity** — and, if the product is *already* finished good or scrap when it is assigned (a finished part assigned from the product screen, say), to **Completed** or **Rejected** as well, so the order's numbers describe the part as it actually is. The **first** production against a **Released** order **automatically moves it to Running**.
- Every association is **recorded on the order** as a *Product Added* event — when, by whom, and at which station — so an order's membership can always be reconstructed, even after a product later leaves it.
- Counting is **material-aware**: an order that names a material moves its counters only for product of *that material*. On an assembly, the component parts stay associated with the order for traceability, but only the finished material's output counts — so an order for material D built from A, B, and C parts reports D's progress, not the sum of everything scanned. An order with no material counts everything assigned to it.
- **Completed and Rejected Quantities are maintained automatically by the system**: a unit counts as completed *good* output the moment its status becomes **shippable and no longer workable** (OK to ship, not OK to work), and as **rejected** when it ends up neither shippable nor workable

(scrap). A disposition change moves the quantity between the two counters — you never tally either by hand; changing a product's disposition (Chapter 10) is what moves the numbers.

- **Splitting a lot keeps both halves on the order.** When a lot on a work order is split (Chapter 3), the new lot joins the same order in the same step, so the split moves quantity *within* the order and its counters come out exactly where they went in. To send part of a lot to a *different* order, split it, then move the new lot (below).
- A product already committed to one order **cannot be built under a different one** at a machine — the floor is told to fix the mismatch (Chapter 8). Moving it is a deliberate step, taken on the product screen.

Move a product to another work order — hand-off or correction

A product belongs to **one work order at a time**. That is what keeps an order's numbers trustworthy: a unit counted by two orders at once would fill both orders' plans, be counted twice in everything reported from them, and leave every machine unsure which order it is building. Over its *life*, though, a unit can legitimately pass through several orders, one after another — and IntelliTrack records each step.

To move a product, open it, choose **Edit** (Chapter 3), and pick the new **Work Order** (or clear it). Moving a product to another order, or off one, needs the **Production Supervisor** or **Administrator** role — it rewrites what an order has been credited with — and for a Production user the **Work Order** of a product that already has one is shown read-only. Because the product is leaving an order, the page then asks **what the order it is leaving keeps**, and it will not save until you answer — each answer rewrites a different order's numbers if it is the wrong one:

Choose	When	What happens to the order it leaves
Hand-off	That order really built the unit, and the unit is moving on: back for rework under a rework order, repaired under a return (RMA) order, or carried on by the next stage's order in a multi-stage build.	Keeps its credit. Its Started/Completed/Rejected quantities are left exactly as they are, and what the unit contributed when it left is frozen on the order's Handed Off (Credit Kept) list. Rework under the next order cannot change it.
Correction	The unit should never have counted against that order: it was scanned or assigned by mistake , or it was re-pegged to another order before it was built.	Gives it back. Exactly what the unit contributed is taken back out of its counters, as if it had never been assigned.

What this changes in the system:

- **The move is all-or-nothing.** If the new order refuses the unit — under **Block** enforcement, because it would take the order past its ceiling — the product stays on the order it was on, nothing is half-done, and you are told why.

- The order the unit joins **counts it as it is now** (see *Assign products* above). Anything else you change in the same edit — sending a finished part back to an in-process status for rework, say — is counted against the **new** order, not the one handing it off.
- Both orders record the move on their **Membership Changes** list (on the work order's Details page): a *Product Handed Off* or *Product Removed* event on the order it left, a *Product Added* event on the order it joined, each naming the other order, who made the change, and when. The product's own history (Chapter 3) records the change of work order as well.
- **Returning to an order it was handed off from** reclaims that credit rather than adding to it, so an order can never count the same unit twice.
- Which order each **production event** actually ran under is recorded on the event itself, at the machine, and never changes afterwards — so moving a unit never rewrites the history of what was built under which order.
- Where your site's signature policy requires it, moving a product to another work order is a signed act (*Move a product to another work order*; shipped unsigned, with the meaning **Authored** and an optional reason). The first **Save** on the product's **Edit** page then saves nothing and shows the **Electronic signature** panel with what will be signed; sign and select **Save** again. The move and the signature are saved together — if the new order refuses the unit, nothing is signed or saved. See [Electronic Signatures & Audit Trail](#).

Quantity enforcement and required work orders

Four system parameters (set by an administrator in [System Configuration](#)) decide how strict work orders are. Each is a **site-wide default that can be overridden for an individual equipment module** (see *Override enforcement for one machine* below) — so one cell can hard-stop at its planned quantity while the line beside it never stops — which means the same order can legitimately behave differently at different stations. Knowing your site's settings explains behavior you will see on the floor:

Parameter	Effect
Work Order Quantity Enforcement (param 90)	Off — counters tracked, nothing enforced. Warn (the default) — reaching the planned quantity is logged as a warning and the order keeps running ; nothing is refused. Block — reaching the plan auto-completes the order, and introducing product that would push it past its ceiling is refused.
Work Order Required (param 91)	When on, production without a work order is refused — every unit must belong to an order.
Bind Work Order To Equipment (param 92)	When on, each station remembers the order it is producing against and attributes order-less scans to it — see <i>How machines use work orders</i> above.
Work Order Recipe Enforcement (param 93)	Off / Warn / Block — holds an order to one pinned control recipe per equipment module — see <i>How machines use work orders</i> above.

How the **Block** ceiling works:

- The ceiling is **Planned Quantity plus the order's Overbuild Tolerance** — an order for 100 with a 10% tolerance may start up to 110.
- The gate measures **net good progress**: what has been started **less what has been scrapped**. An order for 100 that rejects 10 parts needs to start 110 to ship 100, and because scrap is credited back, those 10 replacements are never refused — nobody has to edit the plan to finish an order that had fallout.
- Enforcement gates the *introduction* of new product, never the completion of work in process: a part **already started** against the order is always allowed to finish its routing, whatever the order's status has since become. (Each such finish is logged, so production against a finished order stays visible.)

Note: Auto-completion at the planned quantity happens in **Block** mode only — that is why an order can reach **Completed** without anyone clicking Complete. In **Warn** mode the crossing is recorded in **System Errors & Warnings** and the order keeps running.

Override enforcement for one machine (Equipment enforcement)

The four parameters above are site-wide, and for most plants that is right. But stations are not all alike. A plant can run one work-order-controlled cell beside flow lines that have no orders at all — turning **Work Order Required** on for the site would refuse untracked production everywhere. One cell may need a hard stop at the planned quantity while a downstream washer must never stop. A validated cell may need to hold every order to one recipe while a general-purpose station does not. **Equipment enforcement** lets an administrator give one equipment module its own setting for any of the four, and leave the rest to the site.

Production → **Work Orders**, then the **Equipment enforcement** link above the list (**Administrator** role).

The list shows every equipment module by **Workcell** and **Equipment**, with what actually applies there for **Quantity enforcement**, **Work order required**, **Bind work order to equipment** and **Recipe enforcement**. A value followed by **(site)** — *Warn (site)*, say — is inherited from the site parameter; a value without it is the module's own. Modules with any setting of their own are shown in bold.

To change a module:

1. Select **Edit** on its row. The page (**Enforcement:** and the module's name) offers each setting as a list.
2. Choose **Off**, **Warn** or **Block** (quantity and recipe enforcement) or **Yes** or **No** (work order required, bind work order to equipment) — or **Use the site setting**, which shows the site's current value in parentheses and makes the module follow the site from then on, including any later change to the site parameter.
3. Select **Save**. The list confirms *The work order enforcement of ... was saved*. A save that changes nothing writes nothing (*Nothing changed.*), and returning every setting to **Use the site setting**

removes the module's override altogether.

What this changes in the system:

- The OK-to-Work gate at that module — at the machine and on the web **Check-Out** page alike, since both run the same gate ([Production Execution](#)) — uses the module's setting instead of the site's: the **Block** ceiling, whether an order is required, the fallback to the station's bound order, and the recipe pin. It takes effect from the next part at that module; nothing already produced is re-evaluated.
- Work done without a module — assigning a product on the product screens, receiving — follows the site settings.
- **What happens when an order reaches its plan follows the site, not the module.** That moment comes when a product's status changes, not at a station, so it is decided by the site's **Work Order Quantity Enforcement**. With the site at **Block**, an order still completes itself at its planned quantity — and then accepts no new product at any station — even where one module is set to **Warn** or **Off**. With the site at **Warn**, an order is never completed automatically, even though a module set to **Block** refuses new product past the order's ceiling.

Relaxing enforcement on a machine — letting it run past an order, or against no verified recipe — is the same kind of decision as approving a recipe, so every change is an approved change. With GxP Mode on it is **signed by default** (meaning **Approved**, with a reason required): the **Electronic signature** panel above **Save** covers every setting before and after, and the change and the signature are saved together. An administrator can change that policy for a site, area, workcell or process ([Electronic Signatures & Audit Trail](#)). A module whose enforcement has been signed for shows those signatures in an **Electronic signatures** section at the top of its **Edit** page. Every change — signed or not, with GxP Mode on or off — is recorded in the audit trail (**Configuration** → **Additional Configuration** → **Compliance** → **Audit trail**).

Retention — how long an order lives

Every work order carries an **Expiry Date**, defaulted at creation to a year out (**Work Order Expiry Time**, parameter 13). Expiry does not stop or hide the order — it only marks when the order *may* be cleaned up. An order is pruned automatically only when **all three** are true: it is past its expiry date, **it holds no product any more** — none associated with it, and none it **handed off** that still exists (products are removed first by their own retention rules) — and it is **not dispatchable** (not Released or Running). An order still holding product is never removed, however old — the order always outlives the production it records, including units that went on to a rework or next-stage order, since the order's handed-off list is their link back to it — and an order the floor can still select never disappears from under it.

Keep the counters honest (Rebalance)

If the Started/Completed totals ever look out of step with the products actually assigned — usually after an out-of-band data change — use **Rebalance Quantities** (on the Details page, offered until the order is

Canceled or Closed, to production supervisors and administrators), or the machine's Balance event. Because a recount rewrites the order's plan-against-actual numbers, it is a supervisor's act, not an operator's. It recomputes the counters from the underlying assignments — the products on the order **plus the credit it kept for products it handed off** — restoring agreement between the Details page and reality. (The handed-off credit is counted as it was recorded at hand-off, so a Balance can never strip out what the order built before a unit moved on.) Use it to fix a wrong-looking progress percentage, not as routine maintenance.

Each recount is recorded in the **Lifecycle History** as a **Balance** event naming who asked for it — the only record of a recount, because it does not change the order's **Last Updated By**. A recount is refused where it would destroy the record rather than correct it: on a **Canceled** or **Closed** order, whose quantities are its final production record, and on an order whose products have all been pruned (and that kept no hand-off credit), whose counters are then the only surviving record of what it built.

Cancel, complete, or delete — and why delete is restricted

Cancelling, completing, closing and deleting an order all take the **Production Supervisor** or **Administrator** role: each ends or erases what the floor was authorized to build.

- **Delete** is permitted **only** for a Draft order that has **never** had a product produced against it — no product on it, and no credit kept for a unit it handed off to another order. Once anything was built, deleting would erase the record of why those units exist (and, for a handed-off unit, its only link back to the order that built it), so IntelliTrack blocks it: *"Only a draft work order with no production history can be deleted. Cancel it instead."* The work-order API applies the same rule.
- For any order that has seen production, **Cancel** it (to stop early) or **Complete** then **Close** it (to finish normally). These keep the order — and the story of what happened — permanently.

Tip: If you want to delete a produced order, what you usually want is **Cancel** — it ends the order while preserving the history.

Bulk scheduling and the work-order API (for integrators)

So a scheduling system or ERP can create orders and read fulfillment **without anyone retyping**, IntelliTrack supports a REST API and a scheduled import from the ERP:

- The REST API lives at `api/v1/work-orders`, authenticated with an **API key** in the `X-API-Key` header (keys are issued on the [API Clients](#) admin page). It writes through the **same stored procedures** as the screens and the machines, so an order created by the ERP behaves identically — same state machine, same counters, same dispatch rules. Its changes are recorded under `api:` followed by the client's name.
- **An API key cannot sign.** While your site's signature policy requires a signature to release or close an order, or to change a released order's quantity beyond its tolerance, the API refuses that request

(HTTP 409 Conflict) with a message asking for the change to be made in IntelliTrack, where a person can sign it. Asking for a status the order already has is not refused, so an integration can safely retry.

- The **work order import** of the ERP integration fetches the ERP's production orders on a schedule and creates or updates the matching orders in a batch. New orders arrive in **Draft**, with an expiry date as on the web, and are released in IntelliTrack; Canceled and Closed orders are never changed, and an order's material is left alone once production has started. The import does not consult the signature policies.

Setting this up is an administrator/integration task; see [ERP Integration](#).

Field Reference

The **Enforced during production?** column says whether a field can make IntelliTrack refuse, log, or change anything while parts are being built. A field marked **No** is recorded and reported only: nothing on the floor is refused because of it.

Field	Required?	What it does / valid values	Enforced during production?
Work Order Number	Yes	Unique key referenced by operators, machines, and ERP. Set at creation; not editable afterward.	Yes — a number a machine supplies that matches no order is refused, and so is a product that already belongs to a different order (the Work order gate, Production Execution).
Description	No	Free-text context.	No
Group	No	Batching label; the basis for searching/filtering related orders.	No
Customer	No	Who the order is for; flows into customer-facing reporting.	No
Material	No	The material to produce. Locked once production starts. An order that names a material counts only that material's output; one without counts everything assigned to it.	Counting only — a part of another material is still accepted and associated, but does not move the order's counters.
Planned Quantity	Yes	Target (≥ 0); the denominator of the progress percentage and the basis of the Block-mode ceiling. On a released order, a change beyond the tolerance can require a signature.	Yes, in Block mode — together with the tolerance it sets the ceiling past which new product is refused, and reaching it completes the order (see <i>Quantity enforcement</i> above). In Warn mode, reaching it is only logged.

Field	Required?	What it does / valid values	Enforced during production?
Overbuild Tolerance (%)	Yes (0 = none)	Percentage (0 to 1000) production may run past the plan; raises the Block-mode ceiling to $\text{Planned} \times (1 + \text{tolerance})$, shown as Maximum on the Details page. Changing it on a released order can require a signature.	Yes, in Block mode — part of the ceiling.
Unit of Measure	No	Unit for the planned/started/completed quantities. Only units that can hold a quantity of stock are accepted; an order already saved in another unit keeps it.	No — checked when the order is saved, not during production.
Planned Resource	No	Dispatch filter: naming a plant-model level (module or parent) routes the order to those stations; blank offers it to every station .	Dispatch only — it decides which machines are offered the order, but a part presented with this order at another station is not refused for that reason.
Planned Start	No	Scheduled start, for planning and reporting.	No — an order can be released and built before or after it.
Required By	No	Due date, for planning and reporting.	No — a late order is not stopped or flagged.
Expiry Date	Defaulted	When the order becomes eligible for pruning. Blank on create = creation + parameter 13 days (default 365); blank on edit keeps the existing date. Only an <i>empty, non-dispatchable</i> order past this date is removed.	No — an order past its expiry date can still be built against; the date only affects clean-up.

The order's **status** is enforced as well, although you change it with the lifecycle buttons rather than on the form: new product can join an order only while it is Released or Running (see *The lifecycle* above).

Equipment enforcement (one equipment module; **Engineering** or **Administrator** role):

Setting	Choices	Site parameter it overrides
Quantity enforcement	Use the site setting, Off, Warn, Block	90 — Work Order Quantity Enforcement
Work order required	Use the site setting, Yes, No	91 — Work Order Required
Bind work order to equipment	Use the site setting, Yes, No	92 — Bind Work Order To Equipment
Recipe enforcement	Use the site setting, Off, Warn, Block	93 — Work Order Recipe Enforcement

Related Topics

- [Products](#) — assigning products to a work order
- [Materials & Suppliers](#) — defining the material

- [Production Execution](#) — how units get built
- [Product Audits](#) — audits triggered at work-order start
- [Quarantine & Disposition](#) — the statuses that count a unit as "good"
- [System Configuration](#) — parameters 90–93 and 13, API Clients, units of measure
- [ERP Integration](#) — the work order import, and what is reported back to the ERP
- [Electronic Signatures & Audit Trail](#) — signing a release, a close, a quantity change or a machine's enforcement

Chapter 8: Production Execution

Overview

This is the heart of IntelliTrack: the moment a product is processed at a piece of equipment. Whatever the step — a manual station or an automated cell — the sequence is the same. The product is **checked out** to the equipment, which asks one question: *is this product OK to work here, right now?* If yes, the run is recorded (measurements and any component consumption) and the product is **checked in**, possibly becoming a new material and, if it is the last step, earning a shipping code. If no, IntelliTrack refuses the work and says exactly why.

Why this matters: the "OK to Work" gate is where every rule you configured elsewhere — routing, recipes, work orders, BOMs, WIP limits, audits, expiry — is actually enforced. It is what prevents the wrong part being built at the wrong place in the wrong order, *before* it happens rather than after. And it behaves identically whether a person drives it from the **Check-Out** screen or a **machine** drives it through the PLC integration, because both call the same engine.

Before You Begin

- **Check-Out**, the **OK to Work** screen that records a run (its process data and genealogy rows included), **Shipping Verification** and re-ordering the queue require the **Production** role — the line operators' role; production supervisors and administrators can do them too. They run the process as it was defined and record what happened. **Check-Out** (on the **Production** menu) and the **Inventory** menu, with **Shipping Verification**, are shown only to people who hold one of those roles. The production landing page and the queue itself are viewable by any signed-in user. An administrator decides who holds each role on the **Role mapping** page ([Chapter 19](#)).
- **Changing the recipe** a machine is running, at check-out, needs the **Production Supervisor** or **Administrator** role while **Allow Recipe Change** (System Parameter 22) is 0, as it ships; set to 1, Production users may change it too. Checking out on the recipe the equipment is already running never needs more than Production — see [Who may change the recipe](#).
- Routing is configured by **Engineering** or **Administrators**; carriers by **Administrators**.
- For the gate to pass, the supporting setup must exist: the equipment's **process** and **routing** ([Chapter 2](#)), an active **recipe/control instance** where measurements are evaluated ([Chapter 5](#)), and — if your site uses them — the **work order** ([Chapter 7](#)) and **BOM** ([Chapter 6](#)).
- A recipe is only part of that list when the module actually grades parts. A module whose mapped attributes are all marked *not* used for evaluation is data-logging only, and a machine checks parts out there without one — see [Running without recipes](#) in [Chapter 5](#). The web **Check-Out** page always asks for a recipe, because it puts that recipe on the equipment.

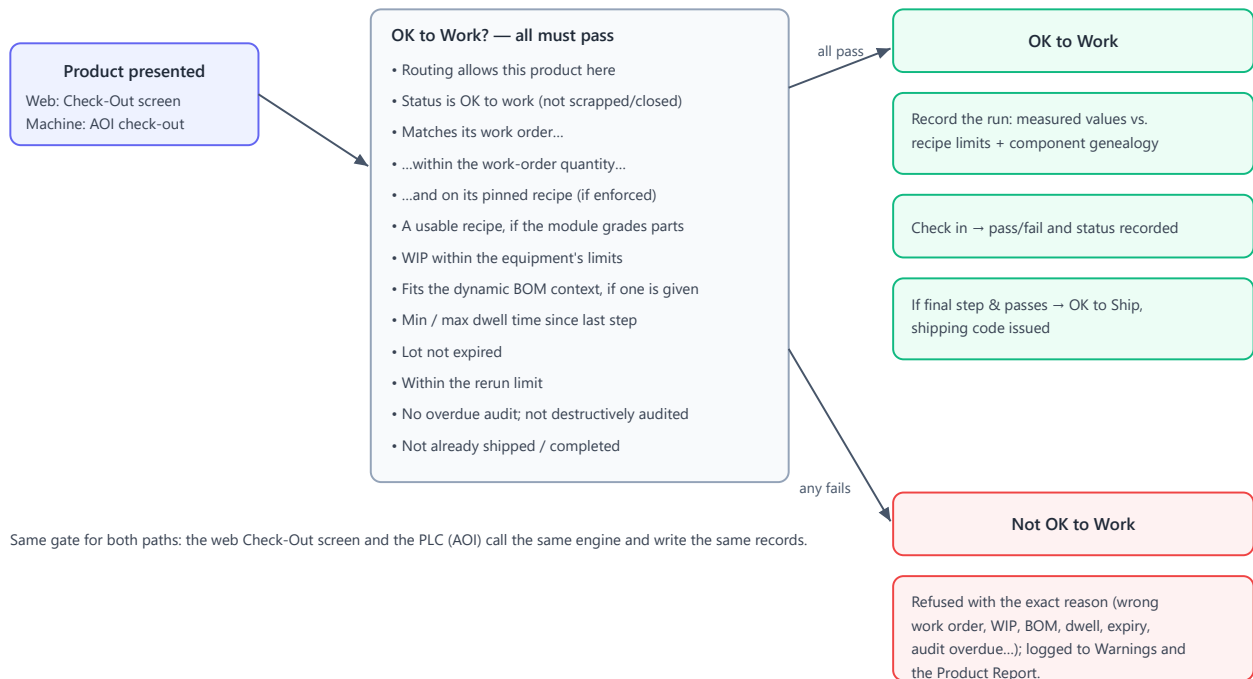
How to Get Here

- Production landing: **Production** → **Production** (page **Production Processing**), with cards for **Check-Out Inventory**, **Product Queue**, **Shipping Verification** and **Quarantine Product**.
- Check-Out: **Production** → **Check-Out** (listed only for Production users and above).
- Queue: the **Product Queue** card on the production landing page.
- Shipping verification: **Inventory** → **Shipping Verification** (listed only for Production users and above).
- Routes: the **Process Routings** list on a process's or an equipment module's **Details** page (Chapter 2).
- Carriers: **Configuration** → **Additional Configuration** → **Materials & Logistics** → **Carriers**.

The OK-to-Work gate — what it checks and why

When a product is checked out to equipment, IntelliTrack evaluates a battery of conditions. **Every one must pass** for the product to be OK to Work; if any fails, the product is refused and the specific reason is returned. The same checks run for the web Check-Out screen and for a machine's check-out.

The OK-to-Work gate — every check must pass



What each gate means, and the effect of failing it:

Check	Why it exists / what failing means
Product found	The serial number (or the product's alias), or the carrier and position, must identify a product — otherwise <i>Serial number not found</i> .

Check	Why it exists / what failing means
Routing	The product must have a valid route from where it was last processed to this equipment, for its material and its current pass/fail state. No route = the part is at the wrong place (<i>No route path found for product</i>). A process with no route paths into it at all is not checked while No Route Approval (System Parameter 40) is on, as it is by default. Configure routes in this chapter.
Status (disposition)	The product's status must have its OK to work flag set; a scrapped or closed product is refused. The flags are what decide, so this follows whatever you configure in Product Status (Chapter 10). Note that Hold blocks shipping, not work — a part under investigation can still be processed, it just cannot leave.
Work order	If the product belongs to a work order, it must match the one being run; with Work Order Required on, untracked production is refused (Chapter 7).
Work-order quantity	Under Block enforcement, introducing product that would push the order past its ceiling — planned quantity plus overbuild tolerance, net of scrap — is refused (Chapter 7).
Work-order recipe	With Work Order Recipe Enforcement (System Parameter 93) at Block , a module that is not running the control recipe pinned to the order refuses the part (<i>Incorrect recipe for this work order</i>); at Warn the mismatch is only logged (Chapter 7).
Usable recipe	A module that grades parts must have an active control instance carrying a value snapshot — something to grade against — or the part is refused (<i>No usable recipe for this equipment...</i>). Data-logging-only modules are exempt (Chapter 5).
WIP	The equipment's work-in-process limits (counts, amounts, age) must not be exceeded (Chapter 2).
BOM	Only when a Dynamic BOM Context S/N is entered at check-out: this exact part must be configured in the dynamic BOM as an allowed component of that assembly, or it is refused (<i>The product is not part of the allowable bill of materials</i>) — see Chapter 6. Machines never send a context, so the check does not apply to them. The static BOM is not checked here: it is enforced as components are added, and at check-in, where a station whose static BOM items are not satisfied rejects the part (<i>Incomplete BOM</i>) — see Chapter 6.
Dwell time	The process's Minimum Time Since Last Operation and Maximum Time Since Last Operation (e.g., a cure that needs time, or a bond that expires).
Expiry	The lot's shelf life must not have passed (Chapter 4).
Rerun limit	A product can be run through the same process only as many times as its Maximum Number of Reruns Allowed permits; after that it is refused here (<i>This product has been processed too many times already</i>).
Audit status	If the equipment's audit is overdue — it came due, and its grace period ran out before the equipment pulled a sample — every check-out at that module is refused until the samples are pulled or the audit is skipped or postponed. A part the equipment flags as an audit sample is still let through. A part once pulled for a <i>destructive</i> audit is refused everywhere. Completing the audit at the audit station does not release the line — the pull does (Chapter 9).

Check	Why it exists / what failing means
Already shipped/final	A finished product cannot be reprocessed.

Note: A **failed** check-out is not silent — IntelliTrack writes a warning and a failed check-out record to the product's history, so the refusal (and its reason) shows up in the [Product Report](#) and in **Log Search (Configuration → System Errors & Warnings — Chapter 17)**. This is how a supervisor sees *"why did this part keep getting rejected at station 4?"* after the fact. (A serial number that matches no product has no history to write to, so that refusal is in the warnings only.)

Check out equipment and record a run (operator)

1. Open **Production → Check-Out**.
2. Select the location down the plant model: **Enterprise → Site → Area → Workcell → Equipment**. IntelliTrack remembers your last selections (and your recipe, material and work order), so you usually only adjust what changed.
3. Select the **Recipe** to run. The list offers only revisions approved for production (an approval status enabled for production, such as **Active**), from your site plus the shared General recipes (Chapter 5). If you may not change the equipment's recipe (see [Who may change the recipe](#)), then once you have chosen equipment that is running a recipe, the list offers only that one. There is no operator to enter: IntelliTrack records you, the signed-in user, as the operator of the check-out and of the check-in.
4. Optionally choose the **Material** the equipment is set up to run. Routing checks the part against it; left blank, the recipe's material is used.
5. Identify the product: enter or scan its **Serial Number** (its alias works too), **or** enter a **Carrier** and **Position** — typing in one clears and locks the other. Optionally enter a **Work Order**, and either of two context serial numbers, which do different jobs:
 - **Recipe Context S/N** — the parent product that calculated recipe values are resolved against; the resolved values are frozen into the control recipe instance's snapshot. It does not affect the BOM check.
 - **Dynamic BOM Context S/N** — the assembly this part is being built into. When given, the check-out is refused unless this exact part is configured as an allowed component of that assembly (see the gate above). Leave it blank to skip that check.

A context serial number that matches no product is refused before anything runs, with the message beside the box that is wrong.

6. Select **Check-Out**. If the equipment has **no** active control recipe at all, you are asked first whether to create one from the recipe you selected: **Yes** creates it, makes it active and continues; **No** stops

at **Active Control Recipe Required** — a product cannot be checked out here without one.

7. IntelliTrack runs the gate above:

- **OK to Work** → you go to the **OK to Work** screen to record the run. This opens a processing-history record stamped with the operator and start time, and (if a work order was given) associates the product with it — which increments the work order's started count and can auto-start it (Chapter 7). A part checked out on its own is taken off its carrier; the carrier and position stay on that history record, which is what pallet analysis reads.
- **Not OK to Work** → the **Product Not OK to Work** page shows the gate's **Result** message and lists the **Reasons the product cannot be processed**.

What your recipe choice does to the equipment. The web check-out has no separate download step, so the recipe you select *becomes* the equipment's running recipe. IntelliTrack uses that recipe's live control instance on the module — creating and activating one if there is none — and closes any other live instance on the module, so a module never runs two recipes at once. Overrides that have finished their approvals are applied at that moment, and the run joins (or opens) the workcell's campaign (see below). Choose with care: checking one part out under a different recipe switches the whole module to it — which is why who may make that switch depends on your role and System Parameter 22 (see [Who may change the recipe](#) below).

Note: If the database cannot complete that recipe step — for example because it is too busy — the check-out is refused with a plain message rather than the database's own text. When the message says nothing was changed, simply try again.

The **OK to Work** screen names the part and the equipment and, beneath them, the **Campaign** the run belongs to. A workcell campaign groups every control instance of one recipe run on one workcell (for one context product): all the workcell's modules running that recipe share it, whether a machine or the web started them. It is named `<workcell>_<recipe>_<revision>_<year><day of year>_<sequence>` — for example `LINE-1_WIDGET-A_3_2026216_01` — so anyone can say which run a part was built under, and the same name heads the run on the recipe's control instances page. A campaign ends only when the last module of the workcell moves to another recipe, revision or context product, or its recipe is cleared — never on a timer.

On the **OK to Work** screen you:

1. Confirm or adjust the product details. **Serial Number**, **Description**, **Material** and **Location** stay locked until you tick **Edit Serial Number**, **Edit Description**, **Edit Material** or **Edit Location**: a new serial number renames the part at check-in (its history shows both names), a new material is what the part becomes after this step, and the location is where it is put. The **Work Order** is shown, not editable. The product's **Disposition Status** is shown, so you can see that a part is on Hold, but it cannot be changed here: checking a part in is not a status decision. The check-in changes a status on its own in just two cases — a reject on an attribute marked **Automatically Scrap on Reject**

scraps the part, and a pass at the final process completes it. To change a status by hand, use the product's **Edit** page (Chapter 3) or a quarantine incident (Chapter 10). On the Edit page, putting a part on Hold is Production's, while releasing it is a Production Supervisor's release decision, signed with GxP Mode on (Chapter 20); raising a quarantine incident is a supervisor's too.

2. Enter the measured **values** for the attributes mapped to this equipment. Each is evaluated against the limits from the **control recipe revision** the product is running under, producing a pass/fail result (limits are recipe-driven — Chapter 5); a row turns green or red as you enter it, as a preview. A value you leave blank is not sent — but every attribute this module must collect needs one: each attribute used for evaluation (the **Required** column ticks them) and any marked **Required for Check-In** (Chapter 2). A missing value is recorded as *not recorded* and rejects the part, without scrapping it.
3. Record component consumption in the **genealogy rows** — **Source, Target** (this part, filled in for you) and **Relationship**; **Add New Row** adds one. Source and target cannot be the same serial number. This is what builds the product's genealogy and satisfies the BOM. The rows you use are offered again for the next part at this equipment and recipe.
4. Select **Check-In**. IntelliTrack checks the serial number (at least 3 characters, the shortest it stores, and not another product's) and the genealogy rows, then records the values and genealogy and computes the cycle pass/fail. **Check-In Successful** confirms that the part was checked in — not that it passed: the page does not show the result, which you read on the product's page or in its Product Report. It returns to Check-Out after a few seconds.
5. At a process marked **Is Final Operation**, a check-in that recorded everything goes straight on to the shipping check (see *Verify a product is OK to Ship*) and shows the new **Shipping Verification Code**, or **Product Failed Shipping Verification** with the reasons — unless **Shipping Verification Delay** (System Parameter 27) is 0, in which case it stops at **Check-In Successful**.

Warning: If a genealogy row could not be recorded, the check-in itself still stands, but **Check-In Successful** shows a red warning naming the source serial numbers whose traceability links were **not** recorded, and it does not move on (to Check-Out or to the shipping check), so the warning cannot be missed. Tell your supervisor and re-enter the genealogy before the part moves on.

Warning: A refused submit records **nothing**: the part stays checked out, with its run still open, and the screen comes back for you to correct. The reason is listed at the top of the form, and a problem with the serial number (shorter than 3 characters, or already another product's) is also shown under the **Serial Number** field. Fix it and submit again; the serial number comes back locked, so tick **Edit Serial Number** to change it. If the part has been removed since it was checked out, the form says so (*This product could not be found, so it cannot be checked in*).

Who may change the recipe

Because a web check-out puts the recipe you choose on the equipment, choosing a different one is more than a check-out: it changes the limits every later part at that module is graded against. **Changing the recipe** means choosing a recipe other than the one the selected equipment is running now (its active control recipe). Choosing a recipe for equipment that is running none is not a change: it is how the first recipe goes on.

- With **Allow Recipe Change** (System Parameter 22) at **0**, as it ships, only a **Production Supervisor** or an **Administrator** may change the recipe. A Production user checks parts out on the recipe the equipment is already running: once the equipment is chosen, the **Recipe** list offers only that recipe.
- With the parameter at **1**, Production users may change it as well — for a plant whose operators set up their own changeovers.

The rule is checked again when you select **Check-Out**, whatever the page offered, so a changed recipe from someone who may not change it is refused and nothing is put on the equipment. An administrator sets the parameter on the **System Parameters** page ([View and change a parameter](#)).

How machines run production (PLC / AOI)

On automated equipment, the PLC runs the IntelliTrack **AOIs** (the controller code in the Help center) and performs the same two steps without an operator:

- **Check-out.** The PLC calls the check-out routine, which runs the **identical gate** and returns a single **result word** — a set of flags telling the controller whether the part is OK to Work and, if not, exactly which checks failed (wrong work order, WIP, no route, BOM, dwell time, expiry, audit overdue, and so on). It also flags conditions that do not refuse the part — for example that an audit is due, so the machine should pull a sample. The controller decides what to do (proceed, divert, alarm) from those flags. A **skip token** is also returned, which the PLC echoes back to intentionally skip an attribute it cannot measure on a given cycle.
- **Check-in.** When the cycle finishes, the PLC submits the measured attribute data in one call. IntelliTrack records the values, computes pass/fail against the recipe limits, and — when configured — **transforms the product** into its post-process material and/or a new serial number, moves it to its new carrier/location, and, on the final step, returns **OK to Ship** and the **shipping code**. Every final-step check-in issues a *new* code: checking a part in again at the final step replaces the code it carried, so a label printed from the earlier code no longer matches.
- **Audit samples.** A machine can flag a part as an **audit sample**. Such a part is let through check-out even while an overdue audit is holding the line, and its check-in opens the audit record QA works from (Chapter 9). The web screens never pull a part for audit.

The takeaway for users: the floor and the office see one consistent picture because the screens and the machines write the same history, genealogy, and process data through the same procedures.

Work the production queue

The queue shows what is waiting at a step so an operator knows what to run next.

1. Open the queue (the **Product Queue** card on the production landing page).
2. View the queue **by process** or **by equipment**, then select **Get Queue**. IntelliTrack remembers your choice.
3. The waiting products are listed, paged, and the page refreshes itself every 30 seconds. To prioritize a part, **drag** it and drop it **onto another part**: it takes that part's place — just above it when you drag up, just below it when you drag down. The new order is saved (Production, Production Supervisor or Administrator role), so prioritizing a hot job on screen actually changes the order the floor pulls them.

The order is kept per **process**. On a single module's page, the part you drop takes the target part's place in the whole process's queue — which is also its place on your page — and the parts waiting at the process's other modules keep their order among themselves. If either part has left the queue since the page was drawn (checked out, moved or scrapped in the last 30 seconds), the drag changes nothing and the next refresh shows the current order.

The order the queue is worked in

The queue is listed in the order the floor should work it. It is the same order a machine gets when it asks IntelliTrack for its queue or for the next part, so the screen and the line agree on what comes next.

1. **Manual priority.** A part you have dragged into position keeps that position, ahead of everything you have not placed.
2. **Soonest expiry.** Among the rest, the part whose expiry date comes first is worked first, so material with a shelf life is used before it goes out of date rather than scrapped after. A part with no expiry date never expires, so it comes after every part that has one.
3. **Longest wait.** Parts that tie on both of those go first in, first out.

Note: A part whose expiry date has already passed is refused at check-out (*Expiry time exceeded*), and because its date is the earliest it sorts ahead of every part you have not placed by hand. It stays there until it is taken out of production — for example, scrapped — so a part stuck at the top of the queue is worth checking for expiry first.

Verify a product is OK to Ship

Use this before shipping to confirm a product completed everything required and to obtain its shipping code — the same shipping check the final production step runs. (Production, Production Supervisor or Administrator role; the **Inventory** menu that holds **Shipping Verification** is shown only to those roles.)

The check passes only when all of these hold:

- the product itself is good — a passing **Operational Status** — and its status is **OK to ship**;
- nothing in its genealogy — no component, at any depth — carries a status that is not OK to ship. This is where **Hold** bites: a held part, or an assembly built from one, can still be worked but cannot ship;
- it was last processed at a process marked **Is Final Operation**;
- with **Include PPD validation** ticked, the latest reading of every attribute used for evaluation passed. A reading that had nothing to be judged against counts as *not* passed: what cannot be verified does not ship.

1. Open **Inventory** → **Shipping Verification**.

2. Enter the product's **Serial Number** (its alias works too) and, to have the process data checked as well, tick **Include PPD validation**.

3. Select **Check**:

- Pass → the page shows the **Shipping Verification Code**, as text and as a barcode to scan. A part that already carries a code shows that same code again; nothing new is issued.
- Fail → **Product Failed Shipping Verification** lists each reason: its category and reason code, the product it concerns (the part itself or a component, with its process and status) and, for a process-data failure, the attribute with its target and actual value. Any code the part carried is cleared, so a label printed from it no longer verifies.

A shipping code belongs to **one** product. The page also offers a **Carrier** and **Position** form, but a carrier is not accepted in place of a product: a check entered there comes back failed, with the warning below. To verify the parts on a carrier, check each one by serial number (the carrier's **Details** page lists them).

Note: If the result carries the warning *The shipping verification could not be completed because of a system error, so this result is not a quality decision*, the check did not run and the part has not been judged either way. Run it again before shipping or holding the part. A serial number that matches no product, and a carrier name, end the same way.

Note: After a pass, the screen returns to the check page after **Shipping Verification Delay** (System Parameter 27; 180 seconds as shipped) so a shipping station can scan the next item hands-free. Set to 0, the result stays on screen.

Set up routing (Engineering or Administrators)

Routing is what the gate's first check evaluates, so it directly determines where a product is allowed to be built.

1. Open the process's or the equipment module's **Details** page (Chapter 2). Its **Process Routings** list shows the route paths into it — viewable by any signed-in user — with the create, edit and delete actions for Engineering and Administrators.

2. Create a route path (**Create a Product Route**):

- **From Process / From Equipment Module** — where the part was last processed (-- *All Equipment from Selected Process* -- for any module of it);
- **To Process / To Equipment Module** — where it is going;
- **Allowed Operational Status** — which parts may take the route: **Good Product, In-Process Inventory, Rejected Product** (a route that sends rejects to a repair station admits only rejects, for example);
- **Rework Status** — whether the route is for reworked parts, the others, or **Both**;
- **Equipment Material** — the material the equipment must be running for the route to apply (**All** for any); **Allowed Materials** and **Disallowed Materials** — the product materials the route admits or refuses. **Matching Material** in the allowed list means the part's material must be the one the equipment is running.

A route that repeats an existing one for the same materials is refused, naming the materials already covered.

3. At check-out the part must match a route from where it was last processed to this equipment, or it is refused (*No route path found for product*). A process with **no** route paths into it is not checked at all while **No Route Approval** (System Parameter 40) is on, as it is by default — so the first route you add to a process switches its routing check on.

4. Deleting equipment removes its route paths automatically.

Set up carriers (Administrators)

A **carrier** is a reusable pallet, tray, magazine, or fixture that holds products at positions. Manage them on the **Carriers** page (create/edit/delete; viewable by any user). Products can then be checked out **by carrier and position** — useful where parts are not individually scannable but their carrier is.

One optional setting turns a carrier into a *managed* carrier:

- **Positions** — the carrier's capacity. Positions are numbered from zero, so a count of 8 means valid positions 0–7. Leave it blank (the field shows -- *Unlimited* --) for unlimited/unknown capacity. When set, IntelliTrack rejects any attempt to place a part in a position the carrier does not have — from the Products pages, from a split, and from the PLC — so a typo cannot put a part in a slot that does not exist.

A position also holds only **one** part at a time, whether or not the carrier declares a capacity. Placing a part in a position another part already occupies is rejected from those same places, and the refusal names the part in the way so you know what to move first. Two exceptions, because they claim no slot: a deleted part frees the position it used to fill, and a part that is on the carrier without a position (see *Split*, Chapter 3) does not block one.

The rest of the form: **Carrier Scope** — the part of the plant model (site, area, workcell, equipment) within which the carrier's name is recognised; **Default Location**; and **Tare Weight** (the empty carrier's weight; 0

if not used) with its **Units of Measure**, which must be a unit that can count material ([Chapter 17](#)).

A carrier in the middle of a group cycle — checked out as a whole tray and not yet checked back in — cannot be deleted: check it in at its current equipment first. Deleting a carrier takes every part off it, and each part's history records who did it and the carrier and position it left.

Create a whole fleet at once (ranges)

Carriers arrive by the pallet-load, not one at a time: a new conveyor loop shows up with ninety-nine numbered pallets on it. So the **Serial Number, Id, or Name** field on the Create page is not one name — it is a *batch*. Enter as many names as you like, separated by commas, spaces or new lines, and every carrier is created in the scope and with the settings shown on the rest of the form. Ranges expand, so ninety-nine pallets take four keystrokes:

You type	You get	Notes
PALLET7	PALLET7	A single carrier — the ordinary case.
A1, A2, B1	A1, A2, B1	Commas, spaces and new lines all separate names.
1-99	1, 2, ... 99	Both ends are included: 99 carriers, not 98.
001-050	001, 002, ... 050	Leading zeros are kept, so names sort correctly everywhere they are displayed.
PAL1-PAL99	PAL1, PAL2, ... PAL99	The prefix must be on both sides of the hyphen.
TRAY-01-TRAY-10	TRAY-01 ... TRAY-10	Prefixes may themselves contain hyphens.

Names that merely contain a hyphen are left alone. TRAY-01, PL1-WP-0P30 and 0P30-40 each create exactly one carrier with that name, because nothing repeats across the hyphen — IntelliTrack only reads a range when you say the same thing on both sides of it (or when both sides are plain numbers). Case must match too: PAL1-pa19 is one name.

Why it works this way: a range is only obeyed when it is unambiguous, so adopting the notation cannot quietly reinterpret the naming scheme your plant already uses. The one exception worth knowing: a name that is *purely* digits-hyphen-digits, such as 1-99, is now read as a range rather than as a carrier literally called "1-99".

What IntelliTrack refuses, and why:

- **A backwards range** (99-1) — almost always transposed digits.
- **More than 500 carriers** in one submission, per range and in total. A 1-100000 is a typo, not a plan; create large fleets in batches.
- **Names longer than 50 characters**, the limit of the stored name.

- **Stray punctuation** — note that `1 - 99` with spaces is three entries, not a range, and the lone `-` is rejected rather than becoming a carrier named `"-"`.

Every problem in the entry is listed at once, and **nothing is created until they are all fixed** — a half-expanded range would leave you guessing which pallets exist. Duplicate names are the exception: if some names in the batch already apply within the selected scope, the rest are created and the page comes back naming each rejected one and the carrier it collides with. Repeats inside your own entry are harmless — `1-5, 3-8` creates 1 through 8 once each.

A carrier whose name matches a live product serial number still saves, with a warning: when that name is typed into the Serial Number field, the product wins, so the carrier is reachable only from the Carrier field.

Group (tray) operations

Some steps process **the whole tray at once** rather than one part at a time. The classic example is a **brazing furnace**: a tray of parts goes through the oven together, so every part should be processed at the same moment, receive the same measurements, and earn the same result. IntelliTrack supports this directly.

Why this matters: it records reality. Fifty parts that physically travelled through the furnace together get one consistent set of process data and one disposition, applied atomically — not fifty independent, possibly-diverging records. And because the tray itself is tracked, you can later ask which pallet a part rode on and whether one pallet is quietly producing worse parts than the others.

How to run a tray operation. There is no separate screen: the **carrier name goes where a serial number would**, with the carrier field left blank. IntelliTrack recognises the name as a carrier and operates on **every product currently in it**. (A real serial number always wins over a carrier of the same name.) Machines do this through the PLC, with no change to their program. On the web **Check-Out** page, type the carrier name into the **Serial Number** field and leave **Carrier** blank: a tray that is refused shows the **Product Not OK to Work** page with the failing position, and a tray that passes is checked out and returns you to Check-Out. The **OK to Work** screen records one product at a time, so a whole tray is checked in by its equipment.

What happens:

- **Group check-out** — every part in the tray must pass the *entire* OK-to-Work gate; if any one part fails, the whole tray is refused and the failing position is named. On success all parts are checked out together and **stay on the tray** (it travels through the process). Trays must hold a single material (*Mixed materials in carrier*), an empty tray is refused (*Carrier empty*), and if the tray's contents change while it is being checked, the check-out is refused with a request to retry.
- **Group check-in** — the measurements submitted are applied to **every** part identically, each part is evaluated and dispositioned, and the tray's run is closed. If it is the final step, each part is checked for shipping on its own, and each one that passes gets its own shipping code; the tray is reported OK to ship only if every part is. A code belongs to one product, never to a tray.

- **Group genealogy** — component consumption you record (for example, the braze paste lot) is applied to every part on the tray.

Carrier maintenance

Trays wear out — wash cycles, furnace trips, handling. On the carrier's **Details** page, the **Tray Operations** panel shows the positions, the **cycles since maintenance** (with an *In Process* badge while a tray is mid-cycle), and the last maintenance date. The **Record Maintenance** button (Administrator role) logs a service event and resets the counter. A wash station can log the same event automatically through the PLC.

Deciding when to service a carrier is a judgement call, not an automatic block. Nothing in IntelliTrack refuses a worn carrier — you decide, from the evidence, and record the service when you do it. The evidence to use depends on how your plant runs:

- **If you check out whole trays by name**, the *cycles since maintenance* figure above counts those runs directly, and **Reports** → **Carrier Reports** → **Cycle Life** lists it for every carrier at once.
- **If you check parts out individually** — the more common case, including any plant where the carrier is not itself scanned at the station — no tray cycles are recorded, and that figure stays at zero. Use **Reports** → **Carrier Reports** → **Carrier Activity** instead: it counts how many parts each carrier carried through each equipment module, which is the traffic that actually wears it. See Chapter 12.

Earlier releases enforced a **Cycle Limit** per carrier and refused group check-out once it was reached. That setting has been removed: it could only ever count whole-tray check-outs, so on a plant that scans part serials it never triggered, and a limit that silently never fires is worse than none. Carrier Activity replaces it.

Pallet performance analysis

Because each processing event is stamped with the pallet the part rode on, you can compare pallets against each other. If you suspect one pallet is defective (for example a bent locating pin on a power-and-free conveyor), the carrier reports (Chapter 12) surface it: **yield by pallet** flags a carrier whose reject rate stands out from the pack, and **attribute analysis by pallet** shows a shifted mean or wider spread on the specific measurement that pallet distorts.

Field Reference

Check-Out:

Field	Required?	What it does / valid values
Enterprise / Site / Area / Workcell / Equipment	Yes	The location to check out; remembered between visits.

Field	Required?	What it does / valid values
Recipe	Yes	Revisions approved for production at this site, plus General recipes. Becomes the equipment's running recipe; its limits drive pass/fail. Choosing one other than the recipe the equipment is running needs Production Supervisor or Administrator while System Parameter 22 is 0 (as shipped); for anyone else the list then offers only the running recipe.
Material	No	The material the equipment is set up to run; routing checks the part against it. Blank = the recipe's material.
Serial Number	Yes (or Carrier + Position)	The product to process: its serial number or alias — or a carrier name, to check out the whole tray.
Carrier / Position	Alternative to serial	Look the product up by carrier and position (0 or greater).
Recipe Context S/N	No	Parent product that calculated recipe values are resolved against. Must be an existing product.
Dynamic BOM Context S/N	No	Assembly the part is being built into; when given, the part must be a configured component of it. Must be an existing product.
Work Order	No	Work order to associate the run with (drives WO counters/status).

OK to Work (per run):

Field	Required?	What it does / valid values
Serial Number	Yes	At least 3 characters; must be unique. Tick Edit Serial Number to rename the part at check-in. A refused serial number says why under the field.
Description / Material / Location	—	Tick Edit Description , Edit Material or Edit Location to change them; the material is what the part becomes after this step.
Work Order	—	Shown, not editable.
Disposition Status / Operational Status	—	Shown, not editable. Change a status on the product's Edit page (Chapter 3) or through a quarantine incident (Chapter 10).
Recipe values	Per attribute	Measurements evaluated against the control recipe limits → pass/fail. Every attribute used for evaluation, or marked Required for Check-In , needs a value; a missing one rejects the part.
Genealogy rows (Source / Target / Relationship)	No	Component consumption; source ≠ target; builds genealogy and satisfies the BOM.

Shipping Verification:

Field	Required?	What it does / valid values
Serial Number	Yes	The one product to verify (serial number or alias).
Include PPD validation	No	Also requires the latest reading of every attribute used for evaluation to have passed.
Carrier / Position	—	Not accepted in place of a product: a shipping code applies to a single product.

Related Topics

- [Products](#) — the products being built, and their genealogy
- [Recipes](#) — the limits that decide pass/fail
- [Work Orders](#) — how production fills and advances an order
- [Bills of Material](#) — the components the gate enforces
- [Product Audits](#) — audits that can hold the line
- [Quarantine & Disposition](#) — statuses that block work
- [Reports](#) — where failed check-outs and run history surface
- [System Configuration](#) — parameters 22, 27, 40 and 93, and the warnings log
- [Security and Sign-In](#) — who holds the Production, Production Supervisor and Administrator roles

Chapter 9: Product Audits

Overview

Product audits are the quality department's independent verification of production. When an audit comes due at an equipment module, the equipment **pulls** a product for audit; a QA technician **scans** it at the audit station, **re-measures** the audit-point attributes with calibrated equipment, and records a verdict. This chapter covers how audits come due and hold the line, running the audit station, completing audits, skipping or postponing a due audit, configuring which processes are audited, and reviewing results.

Why this matters: an audit is not just a second opinion — it is a control that can **hold the line**. When an audit comes due and no sample is taken in time, IntelliTrack refuses further production at that equipment until one is, which is what stops a drifting process from making hundreds of bad parts before anyone checks. And because QA measures against the **same recipe limits the part was produced under**, an audit is a true apples-to-apples check of the line, not a separate standard.

Before You Begin

- **Any signed-in user** can open the audit station and its backlog, the completion form, an audit record, an equipment module's audit history, and the Audit Results page, and can search Audit Results.
- **Submitting** a completed audit — closing the audit record — requires the **Quality** or **Administrator** role. Anyone else can read the completion form, but it offers them no **Submit Audit**: a note says the Quality role is needed. Neither **Production** nor **Production Supervisor** can close an audit record — the audit checks their line. An administrator maps roles to people on the **Role mapping** page ([Chapter 19](#)).
- **Skipping or postponing** a due audit requires the **Production Supervisor** or **Administrator** role. Each is a **release decision** — it waives a quality check the line would otherwise stop for — so with GxP Mode on it must be signed by someone who holds Production Supervisor: an administrator without that role reaches the page but cannot sign the override ([Getting Started](#)).
- **Configuring** which processes are audited (and how often) requires the **Administrator** role.
- A product reaches the audit backlog only when the **equipment** pulls it: a machine checks it in as an audit sample (Chapter 8). The web Check-Out and OK to Work screens never pull a part for audit.
- Each equipment module runs its own audit cycle, by its process's audit configuration.

How to Get Here

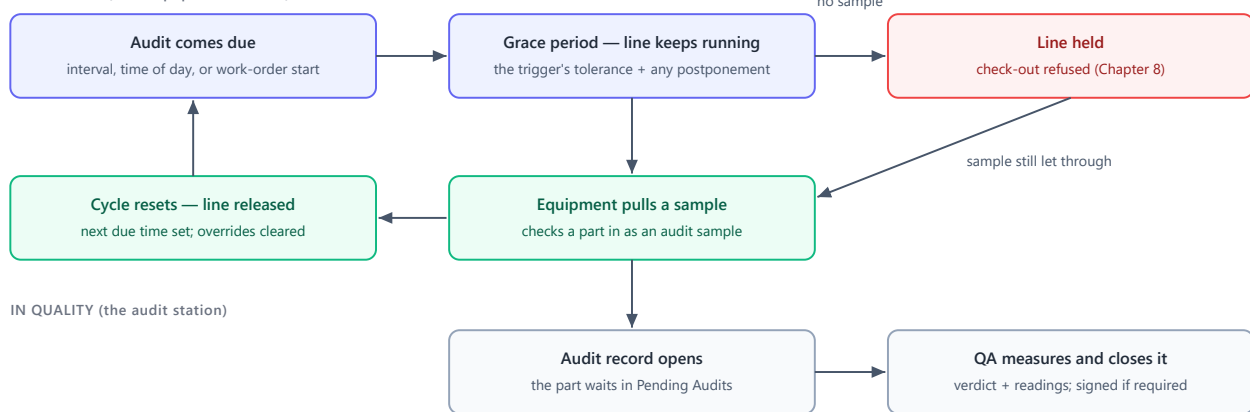
- Audit station and backlog: **Quality** → **Product Audits** or **Production** → **Product Audits** — two menu entries for the same page, the **Product Audit Station**.

- The same page from elsewhere: the **Product Audit Station** link on a process's details page, on an equipment module's audit status and on its audit history, and **Back to Audit Station** on the Audit Results page.
- Audit results report: **Reports** → **Audit Results** (also linked from the audit station).
- An equipment module's audit cycle and history: the **Audit Status** panel on its **Details** page, and its **Audit History** link.
- Audit configuration: the **Audit Configs** link on a process's details page (Chapter 2).

The audit loop — and how it gates production

The audit loop — the line's cycle, and QA's verification

ON THE LINE (each equipment module)



Closing the record is the verification — it does not change the line's cycle; the pull already released the line.

Skip or postpone (Production Supervisor; a release decision): a skip resets the cycle like a pull; a postponement extends the grace.

Limits come from the control recipe revision the part was produced under — QA grades against the line's own standard.

With Allow Product Re-Introduction off the audit is destructive: a pulled part is refused at every later check-out.

When an audit comes due. Each equipment module keeps an audit cycle, anchored on its last audit — the last pull, or skip. An audit comes due at whichever happens first:

- **Time interval** — the process's **Time Interval** has passed since the last audit;
- **Time of day** — one of the process's **Schedules** comes round;
- **Work-order start** — the equipment reports a work order starting, on a process set to **Trigger at Work Order Start** (Chapter 7). This asks the machine for a sample; on its own it never holds the line, which only the time-based grace below can do.

While an audit is due, every check-out at the module tells the machine that an audit is required. Production carries on; the machine is expected to pull a sample.

Grace, then the hold. The line may keep running for a grace period after the audit came due. When a schedule's time of day brought it due, the grace is that schedule's own **Tolerance**: once the scheduled time arrives, the process has that long to pull its audit samples. When the time interval brought it due, the grace is the process's **Interval Tolerance**; the same applies to a schedule since deleted. Either is extended

by any time an authorized user has postponed the audit (see *Skip or postpone a due audit*). Once that is used up with no sample taken, the production gate (Chapter 8) refuses every check-out at that module — *An audit is required before further parts can be processed* — except a part the equipment flags as an audit sample. This is the mechanism that turns "we should audit periodically" into "we *will*, because the line stops otherwise."

The pull releases the line. When the equipment checks a part in as an audit sample, IntelliTrack opens an **audit record** for it (*Pulled for audit by equipment ...*) and sets the part's status to **Nondestructive Audit** — or, where the process does not allow audited parts back into production, to the destructive-audit status. Once the process's **Units per Audit** samples have been pulled, the next cycle starts: the next time-based audit is due one **Time Interval** after the pull (or at the next scheduled time of day, if sooner), the unit count starts again from zero, and the postponements and skips used so far are cleared. The line does not wait for the quality department; it resumes as soon as the samples are out.

QA verifies. The part now waits in the QA backlog until a technician scans it, measures it and closes the record. Closing the record is the verification — it does not change the line's cycle, which the pull has already reset. A failed audit raises a notification instead (see *Complete an audit*).

Destructive audits. Where the process's **Allow Product Re-Introduction** is off, a part that was pulled there is refused at every later check-out, at any process (*The product was part of a destructive audit and cannot be reintroduced*).

Note: A module has an audit cycle only once IntelliTrack is tracking it. Until then its **Audit Status** panel says *Audit tracking has not been initialized for this equipment yet*, no audit comes due there, and its audit history offers no skip or postpone. Tracking starts with the first part checked in at a module whose process has an audit configuration, or when the equipment first reports a work order starting on a process set to **Trigger at Work Order Start**.

Note: The unit-count settings — **Max Units Between Audits**, **Units Tolerance**, **Max Override Units** — are measured against the module's **Units Since Last Audit**. Every check-in at the module adds its parts to that count (a tray check-in adds each part on the tray), and the pull of the required samples or a skip resets it to zero. Once **Max Units Between Audits** parts have run, the next part is the audit: the audit comes due, and after **Units Tolerance** more parts (plus any units postponed) check-outs are refused until a sample is pulled. A **Time Interval** of 00:00:00 means the process is not audited by time at all — only by units, schedule or work-order start.

Run the audit station

1. Open the audit station (**Quality** → **Product Audits**). The open audit parts are listed below the scan box — grouped by process, then by equipment module, oldest first within each. To find yours, fill in any of **Serial Number** (part of a serial number or alias), **Material**, **Process** and **Equipment** above

the list and select **Search Open Audits**; **Clear** shows the whole backlog again. Only parts are listed — the entries a skip or postponement writes have nothing to measure.

2. Scan or type the product's **Serial Number** (or its alias) and select **Find Audit**. The box keeps the focus, so a barcode scanner's Enter submits it for you.
3. IntelliTrack finds the open audit record and opens the **completion form**, or says why it cannot: nothing was entered, no product has that serial number or alias, the product *is not waiting on an audit*, or it *has already been audited* (see the audit results). A row's **Enter Values** button in the list opens its form directly.

Complete an audit

The completion form (**Complete Audit**) shows the product — serial number, material, description, **P/F Status**, **Disposition** — and the pull: process, equipment, **Pulled At**, **Pulled By** and the product's **Born On Date**. Below is the **Control Recipe** the part was run against — the recipe, its revision and the control instance revision — and one row per **audit-point attribute** of that process (attributes marked **Audit Point** — Chapter 2), with that recipe's **Target**, **Low Limit** and **High Limit** for it. For each, the **Line Value** — the equipment's most recent reading of that attribute for this part — is shown so you compare your calibrated measurement against what the line recorded. It is coloured by the line's own verdict: green passed, red failed, orange out of limits on an attribute that is not used for evaluation (informational only). Hover over it to see when it was measured and the limits it was judged against.

1. For each audit point, enter your **Measured Value** — for a yes/no attribute, choose **True** or **False** — or tick **Not Measured**. Every row must be one or the other (never both, never neither). Where the recipe can judge the value, the row's **Result** shows **Passed** or **Failed** as you type; where it can't (no limits, no target text, not in the recipe), choose the row's **Result** yourself — a measured row cannot be submitted without one.
2. Set the overall **Result** — **Good Product** for a pass, **Failed Products** for a fail — and add **Notes**.
3. If an **Electronic signature** panel is shown, complete it (see below).
4. Select **Submit Audit**. IntelliTrack closes the record (stamping who and when), stores each measurement, and returns you to the scan station for the next part.

With **GxP Mode** on, closing an audit record is signed by default, by someone who holds **Quality** or **Administrators** as shipped. The **Electronic signature** panel appears on the form with the statement *I record the results of this audit of ...* and the meaning **Performed**; no reason is asked for unless your site's signature policy asks for one. You sign the verdict, the notes and every measurement row together, and they are saved in the same step as your signature — if the signature is refused, nothing is closed. The record's page then lists it in its **Electronic signatures** section. How to sign: [Electronic Signatures & Audit Trail](#).

What this changes in the system:

- The record is **closed** with your verdict: it leaves the backlog and appears in Audit Results and in the **Quality Audits** section of the product's Product Report (Chapter 12). Closing it does **not** change the line's audit cycle — the pull already released the line; this is the verification record.
- Your readings are written into the **process-data stream flagged as audit values** — the only path that produces audit-flagged data — so reporting can show verification values inline, set apart from production data.
- The **target and limits come from the control recipe revision the part was produced under** — the one recorded on its last check-in at the pulling equipment, or, if the equipment logged no data for the part, the revision it was running when the part was pulled — and cannot be edited on the form. Wherever the recipe can judge a value it grades it by the same rules the line uses, and **its** verdict is recorded; where it can't, you judge the row.
- The audit is added to the product's **History** as a **QA Audit** row naming the result and the pulling equipment; a failed audit lists its failed points as that row's reject reasons.
- If the audit **fails** — the overall result is **Failed Products**, or any row fails, even when the overall result is a pass — IntelliTrack raises a **Failed Inspection or Test Recorded** event naming the failed points (the template token), so a publication of that type can alert QA or engineering, optionally only for chosen sites, areas, workcells or equipment ([Notifications](#)). It is raised once per record. Someone receives it only if such a publication exists.

Warning: If another QA technician already closed the record, your submission is not applied — the first verdict stands and nothing is overwritten.

Note: Submitting the form requires the **Quality** or **Administrator** role (see *Before You Begin*). Without it the form shows *Recording an audit needs the Quality role (or Administrators)* and no **Submit Audit**, and the record stays open for QA.

Skip or postpone a due audit (Production Supervisor or Administrator)

Sometimes the line cannot give up a sample in time — the audit bench is down, no QA technician is free, the run is nearly finished. Rather than stop production or work around the control informally, a production supervisor can **skip** the audit or **postpone** it, within limits the process sets, and the override is recorded against the equipment with their name.

Skipping and postponing are **release decisions**: each lets the line run on without the quality check it would otherwise stop for, so they are a supervisor's call rather than an operator's. With GxP Mode on each must be signed by someone who holds **Production Supervisor** — an administrator who does not also hold that role can open the page but cannot sign the override. With GxP Mode off no signature is taken, and administrators make them like any other supervisor task.

1. Open the equipment module's **Audit History** (the link on the **Audit Status** panel of its details page).

2. The **Audit cadence** card shows where the module stands — **Next audit due**, **Last audit**, **Units since the last audit** — and how much of each override has been used against the process's limit: **Skips used**, **Minutes postponed**, **Units postponed** (for example *1 of 2*).
3. Choose one:
 - **Skip this audit** — ends the current cycle as if the samples had been pulled, without them: the next due time is set, and the postponements used are cleared. Allowed while the skips used are below the process's **Max Skips**. That count resets only when real samples are pulled, so repeated skips run out.
 - **Postpone by time** — lengthens the grace period by the step in **Default Audit Override Time** (System Parameter 70; 40 as shipped), in minutes, but never past the process's **Max Override Time**.
 - **Postpone by units** — adds the same step as units, up to the process's **Max Override Units**, to the units the line may run past the audit coming due.
4. With GxP Mode on, each override is signed by default, a **reason is required**, and the signer must hold Production Supervisor; the statement names the equipment — for example *I postpone the audit on LINE-1-OP20 by up to 40 minutes*. See [Electronic Signatures & Audit Trail](#).
5. Select the button. The page confirms *The audit override was recorded*. — or, when that override is already used up, *The process's limit for this override is reached; nothing was changed*. In that case nothing is signed either.

Each override adds an entry to the module's audit history — *Audit Bypassed*, *Audit Postponed by 40 minutes*. or *Audit Postponed by 40 units*. — naming who made it; its signature, when one was required, is kept with the entry, and the override is in the audit trail. The entry has no product; it shows as **Pending QA** in the history, and is not listed as an open audit part.

The card appears only for a module that has an audit configuration and an audit cycle, and the three buttons only for the Production Supervisor and Administrator roles.

Review the QA backlog

The **Pending Audits** worklist, under the scan box on the audit station, shows every open audit part across all processes. It is grouped by process and then by equipment module (each with a count of waiting parts), oldest first within each group, so QA works the backlog in order. Select a process or equipment heading to collapse its parts out of the way, and again to show them; the list always opens fully expanded. Each row shows when the part was pulled (**Pulled At**), how long it has been **Waiting**, its serial number (a link to the product), its material, who pulled it and the pull description, and an **Enter Values** button that opens the completion form.

Review audit results

1. Open **Reports** → **Audit Results**.

2. Filter by **Serial Number** (or alias), **Material**, **Process**, **Equipment** module, and a **Start Time / End Time** window on the completion time, then select **Search**. Up to 500 completed audits return, newest first. Any signed-in user can search.
3. Each row shows when the audit was completed, the process, equipment, product, material, verdict, who completed it, and the measurement count. **Details** opens the full read-only **Audit Record**: the pull, the verdict, the notes, every stored measurement and — when it was signed — its **Electronic signatures**.

A product's own audits also appear in the **Quality Audits** section of its Product Report (Chapter 12), including audits recorded at other sites once they have synchronized. From an equipment module you can open its **Audit History** — every audit record for that equipment, newest first, each showing who pulled the part and, once closed, who completed it (*puller* → *technician*).

Configure audits for a process (Administrators)

A process's audit configuration is what the cycle above runs on: when an audit comes due, how long the line may run past it, how many samples a pull takes, how far an authorized user may skip or postpone it, and whether audited parts may return to production. It belongs to the process, so every equipment module of the process runs the same rules, each on its own cycle. The configuration is kept in the shared configuration database, like the process itself, so it applies at every site that runs the process and reaches each site's line with that site's next configuration synchronization (Chapter 17).

1. Open the process from **Configuration** → **Processes** and select **Audit Configs**. With no configuration yet you are taken to **New Audit Configuration**; otherwise **Audit Configuration Details** shows the settings and schedules, with **Edit** and **Delete** links.
2. Set the fields (see the Field Reference) and add **Schedules** with **New Schedule** — one row per time of day at which an audit comes due. Saving takes you to **Audit Configuration Details**.
3. **Delete** removes the configuration and its schedules; the process is no longer audited.

Note: Which attributes are audited is set by marking process attributes as **Audit Point** (Chapter 2). Turning off **Allow Product Re-Introduction** makes the audit *destructive*: a pulled part is refused at every later check-out.

Note: **Time Interval**, **Interval Tolerance** and **Max Override Time** are kept to the minute. Type each in full as HHH:MM:SS — up to three digits of hours, then two each of minutes and seconds, with the seconds 00 (for example 01:30:00 for an hour and a half). A blank, shorthand such as 90 or 5:30, or a time with seconds is refused on the form against that field, and nothing is saved.

Note: A schedule's **Time of Day** is a time on the 24-hour clock, from 00:00:00 to 23:59:59, and its **Tolerance** is typed like the times above. A blank or malformed one is refused on the form against that row's field, and nothing is saved. A schedule row you delete before saving is not checked.

Field Reference

Audit completion (per row):

Field	Required?	What it does / valid values
Measured Value / Not Measured	Yes (one of)	Your calibrated reading (True/False for a yes/no attribute), or an explicit skip. Exactly one per row.
Result (row)	When the recipe can't judge	Shown as Passed/Failed where the recipe judges the value (its verdict is recorded); otherwise choose Good Product (pass) or Failed Products (fail). A failed row fails the audit.
Result (overall)	Yes	The audit verdict recorded on the record: Good Product or Failed Products . A fail raises the Failed Inspection or Test Recorded event.
Notes	No	Free-text notes stored with the record.

Audit configuration:

Field	Required?	What it does / valid values
Units per Audit	Yes (1 or more)	How many samples a pull takes: the next cycle starts once this many parts have been pulled since the audit came due.
Time Interval	Yes (HHH:MM:SS, whole minutes)	Time from one audit (the pull of its samples, or a skip) to the next coming due. 00:00:00 = not audited by time.
Interval Tolerance	Yes (HHH:MM:SS, whole minutes)	How long the line may keep running after the time interval brings an audit due before check-outs are refused. 00:00:00 = no grace.
Schedules — Time of Day	No (on each row: Yes, 00:00:00 to 23:59:59)	Times of day at which an audit comes due, besides the interval, on the 24-hour clock.
Schedules — Tolerance	Yes, on each row (HHH:MM:SS, whole minutes)	How long the line may keep running after that schedule's time of day before check-outs are refused, unless the part is the audit sample. 00:00:00 = no grace. Of two schedules at the same time, the shorter tolerance applies.
Max Skips	Yes (0 to 10)	How many times the audit may be skipped before samples must be pulled; 0 = no skipping.

Field	Required?	What it does / valid values
Max Override Time	Yes (HHH:MM:SS, whole minutes)	The most the grace period may be postponed by; 00:00:00 = no time postponement.
Max Units Between Audits / Units Tolerance / Max Override Units	Yes	The unit-count equivalents of the interval, the tolerance and the time postponement, counted from the parts checked in (see the note on unit counting). 0 = not audited by units.
Trigger at Work Order Start	No	Makes an audit due when the equipment reports a work order starting, in addition to the other triggers.
Allow Product Re-Introduction	No	Off = destructive audit: a pulled part gets the destructive-audit status and is refused at every later check-out.

Related Topics

- [Production Execution](#) — the OK-to-Work gate audits hold, and how machines pull audit samples
- [Recipes](#) — the snapshot limits audits grade against
- [Work Orders](#) — audits triggered at work-order start
- [Reports](#) — the Product Report's Quality Audits section, and process-data reporting
- [Plant & Equipment Model](#) — audit-point attributes and the equipment details page
- [Notifications](#) — alerts on failed inspections
- [Electronic Signatures & Audit Trail](#) — signing audit closures and overrides

Chapter 10: Quarantine & Disposition

Overview

When a quality problem is found, you need to stop the affected product — fast, and across everything it touches. IntelliTrack does this with **quarantine incidents** (which find the affected population and change its status in one action) on top of **dispositions** — the named status codes whose **OK to work** and **OK to ship** flags are what actually gate production and shipping everywhere in the system. This chapter covers raising and resolving quarantine incidents and maintaining dispositions.

Why this matters: a disposition is not a label, it is an interlock. A product whose status is not *OK to work* is refused at the production gate (Chapter 8); a product not *OK to ship* is held by shipping verification, and only a ship-able-and-not-workable status counts a unit as completed good output on its work order (Chapter 7). A quarantine incident is how you apply that interlock to many products at once — by serial list, by the equipment that made them, by genealogy, or by performance — so a containment that would take hours of manual status changes happens in seconds.

An incident is raised in two steps. A **search** finds the affected parts — every part your criteria match, plus everything built into them or from them — and nothing changes yet. The **confirmation** is the decision: it moves those parts to the incident's status. The floor then **acknowledges** each part as it deals with it physically, and an incident that turns out to be unnecessary is **rolled back**, returning every part it still holds to the status it had before. With GxP Mode on, the confirmation, the dispositioning acknowledgements and the rollback are electronically signed by default; marking a part *Contained* is not ([Electronic Signatures & Audit Trail](#)).

Before You Begin

- **Viewing** incidents and their suspect-product lists, and **exporting** a list, are available to **any signed-in user**.
- **Marking quarantined parts Contained** (on screen or by upload) takes the **Production** role. It records that the floor has physically segregated a suspect part — an act that only stops product — so it is open to the people on the line, and no signature policy can take it away from them.
- **Every other acknowledgement** (*Released – Passed Inspection, Reworked and Verified, Use-As-Is Approved, Scrapped, Destroyed, Returned to Supplier, Customer Notified*, and clearing one back to *Unacknowledged*), **raising** an incident (searching, then confirming or discarding it) and **rolling one back** take the **Production Supervisor** or **Administrator** role. Dispositioning suspect product and rolling back an incident are **release decisions**: they decide what becomes of suspect product, or let it continue, so with GxP Mode on they must be signed by someone who holds Production Supervisor — an administrator without that role reaches the page but cannot sign ([Getting Started](#)).

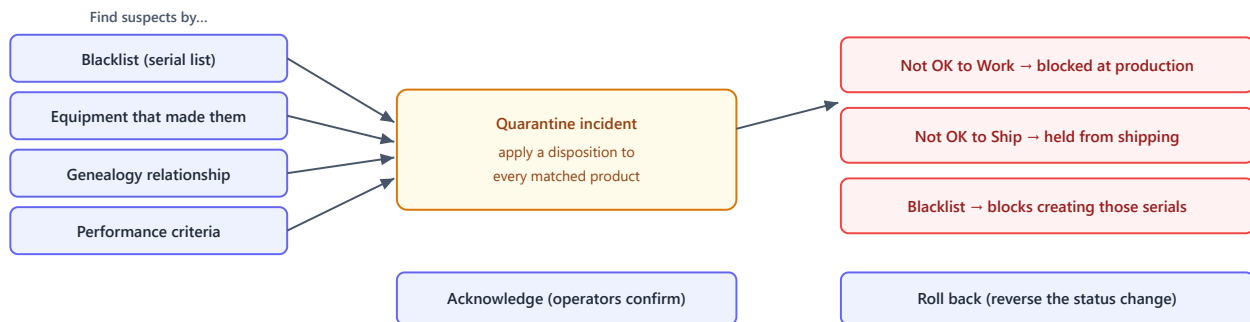
- **Editing** and **deleting** incidents, and **creating, editing and deleting dispositions**, require the **Administrator** role. Anyone signed in can view the dispositions. Which directory groups carry each role is set by an administrator ([Security and Sign-In](#)).
- The status you want to apply must already exist as a product disposition marked **Available To Users** — only those are offered.
- This site must have been assigned its **Site Id** (System Parameter 1), so the search can be scoped to it — see [Quarantine is per-site](#).
- With **GxP Mode** on, some of these acts ask for your electronic signature; each task below says which. How to sign is in [Electronic Signatures & Audit Trail](#).

How to Get Here

- Quarantine incidents: **Quality** → **Quarantine**.
- Dispositions (status codes): **Configuration** → **Additional Configuration** → **Quality & Change Control** → **Disposition Codes**.

How a quarantine incident propagates

One incident → a whole population contained



The effects on the right come entirely from the disposition's OK-to-Work / OK-to-Ship flags — the same flags every gate in the system reads.

Rollback releases the population by reversing the disposition the incident applied.

Every search also takes in each match's genealogy, both ways: the components inside it and the assemblies built from it.

Nothing changes until you confirm the search results; confirming is the decision, and with GxP Mode on it is signed.

Where the quarantine search runs

When you create an incident, IntelliTrack searches for the affected parts in the **reporting database**, not in the live plant database. The disposition you confirm is then applied to the **plant database**, and flows back out to reporting on the next synchronisation.

That split is deliberate, and it buys you two things.

It keeps a recall off the production system. Searching by performance criteria reads every recorded measurement in your time window — easily millions of readings on a busy line. Running that against the

live plant database would compete with the equipment trying to check parts in and out. Running it against the reporting copy means a quality engineer can search as widely as they like, at any hour, without slowing the floor.

It lets a recall reach further back than the plant database can see. The plant database keeps only a working window of recent parts; older ones are pruned to keep it fast. The reporting database keeps them far longer. A supplier notification that arrives months after the fact will name parts the plant database has already forgotten — and those are exactly the parts a recall must not miss. Because the search runs against reporting, they are still found.

Parts found this way fall into two groups:

- **Still in the plant** — these are staged on the incident, and confirming it gives them the incident's status, so they are blocked from being worked or shipped in the usual way.
- **History only** — already shipped, consumed, or aged out of the plant database. There is no live inventory to hold, so nothing is applied to them; the confirmation page tells you how many there are, for traceability and customer notification: *"N affected part(s) were found in the reporting archive but are no longer held at this site, so no status could be applied to them. Review them for customer notification."* That count is shown only there — it is not kept with the incident, and the parts are not listed on it — so record it (for example in the incident's **Notes**, through **Edit**) before you leave the page.

The plant database is always the system of record for parts it still holds. Statuses are never edited directly in reporting for those parts; they are set in the plant and synchronised outward, so the two never disagree.

Parts already held are left out. A part that another open incident — confirmed and not rolled back — already holds is not staged again: it is already contained, and the incident that holds it is the one that will release it.

The search sees what reporting has received. The reporting copy is only as current as the last **Site** → **Reporting** synchronisation ([System Configuration](#)), so a part checked in during the last few minutes may not be found yet. This matters most when the suspect equipment keeps running while you raise the incident: whatever it makes after the search is not on the incident. Once the equipment is stopped and the next synchronisation has run, raise a second incident with the same criteria and a window that runs to when the equipment stopped. Because the parts the first (confirmed) incident holds are left out, the second one picks up only the parts made since the first search.

Quarantine is per-site

The reporting database is shared by every site in the enterprise. A quarantine search is always scoped to the site you are signed in to, and **will only ever put that site's parts on hold**. (Rows the archive received before this site was registered for centralized reporting carry no site at all, and the search counts them as

this site's. On an archive shared by several plants they can add to the history-only count, but they can never be held: only parts in this site's own plant database can be. See [System Configuration](#).)

This matters when an issue spans more than one plant — a bad lot from a supplier who ships to several of your sites, for example. In that case **each affected site raises its own quarantine incident**. There is no single action that quarantines parts across sites, and that is intentional: dispositions are applied in each site's own plant database, each site's operators acknowledge their own parts, and each site can roll back independently without disturbing the others. A cross-site action would mean one site's engineer changing the status of parts on another site's floor.

In practice, send the affected serial numbers or criteria to each site's quality contact and have each of them raise an incident locally. Putting a shared reference — a supplier RMA or corrective-action number — in the **incident name** makes the related incidents easy to find together later.

If this database has not yet been assigned a **Site Id** (System Parameter 1), the search cannot be scoped and refuses to run: *"This database is not yet bound to a site, so the quarantine search cannot be scoped. Set the Site Id system parameter and try again."* An administrator sets this once, when the site is commissioned (see [Set the current site](#)).

Raise a quarantine incident

Raising an incident is a search followed by a decision. **Create New** runs the search and *stages* what it finds; no part changes status. The confirmation page then shows what would change, and **Ok** applies it. Keeping the two apart means a mistyped window or criterion costs nothing — cancel and search again — and it is why a signature, when one is needed, is asked for at the confirmation and not at the search.

Raising an incident — every type, and confirming or discarding what it staged — takes the **Production Supervisor** or **Administrator** role: it changes the status of a whole population at once, and a **Black List** also reaches forward to parts not yet made.

1. Open **Quality** → **Quarantine**. In **Create a new Quarantine Transaction**, choose the type and select **Create New**:
 - **Black List** — a list of serial numbers, typically from a supplier notice.
 - **All Product from Selected Equipment Module(s)** — everything processed on suspect equipment during a time window.
 - **All Products in Genealogy** — one named part and everything related to it.
 - **All Product with Matching Performance Criteria** — every part whose recorded measurements meet the criteria you set.
2. Enter the type's inputs (see [What each type finds](#)), a **Name**, optional **Notes**, and the **Status** to apply. The list shows each status's *OK to Work* and *OK to Ship*, so you can see what it will do as you choose (see [Choosing the status](#)).
3. Select **Create New**. IntelliTrack runs the search, stages the parts this site still holds and opens the confirmation page: the transaction details and **Items Effected**, the number of parts that will

change status. For a **Black List** it also lists **Parts already in system** (serials on your list that this site's plant database holds) and **Parts not yet seen** (the rest).

4. Select **Ok** to confirm, or **Cancel** to discard the incident — see [Confirm or cancel](#).

If the search itself fails, the incident is discarded and the form says so — *"The search for affected parts could not be completed because of a database error, so the quarantine was discarded and nothing was changed. Please try again."* — so a failed search never leaves a half-made incident behind.

Warning: A **Black List** incident also puts its serial numbers on the **Blacklist**, and it does so when you select **Create New**, before you confirm. From then on a new product with one of those serials is refused when it is created or received — unless the incident's status still allows work, in which case it is let in, and the product **Create** page or a machine adding it gives it the incident's status ([Products](#)). The containment reaches back to the point of entry, including parts the plant has not seen yet. Receiving and machines match an entry only for the supplier they name; the **Create** page matches the serial alone. The entries last as long as the incident: a rollback releases the held parts but leaves the serials on the blacklist, and they come off only when the incident is cancelled at confirmation or deleted. No page sets a single entry aside.

What each type finds

Every type widens what it matches through genealogy in **both directions**: the components inside each matched part and the assemblies it was built into, for as many generations as the records go. A raw-material recall therefore reaches every finished good containing the bad lot, and a bad sub-assembly takes its components with it. A matched part with no genealogy record at all is still included, and the search is not cut short by **Genealogy Rows** (System Parameter 41), the limit on genealogy views: a containment list must be complete.

- **Black List** — the **Supplier** the notice came from, and the serial numbers: paste them into **Manual Serial Number List** (separated by comma, space, semicolon, tab or new line), upload a **File** (.xlsx, .csv, .tsv or .txt; **Download spreadsheet template** gives you a ready-made one), or both. A heading cell such as *Serial Number* is ignored, a serial listed twice counts once, and an entry longer than 200 characters is ignored. Each serial must match a part's whole serial number, not just its beginning.
- **All Product from Selected Equipment Module(s)** — the suspect modules, picked in **Equipment Model Definition** (**Enterprise** → **Site** → **Area** → **Work Cell** → **Equipment Module**; several modules may be chosen), and a **Start Time** and **End Time**. A part is caught if its time on any chosen module *overlapped* the window — including one that went onto the module before the window opened, and one still on it now. This is the one type whose suspect list is grouped by assembly (see [Acknowledge the quarantined products](#)).
- **All Products in Genealogy** — the **Serial Number** of the part at the centre of the problem (up to 50 characters). If no part at this site carries that exact serial, IntelliTrack tries it as an alias (when

exactly one part here has that alias), then as a former serial number of a part that has since been renamed.

- **All Product with Matching Performance Criteria** — a **Process** (the search reads measurements taken on that process's equipment), optionally **Materials** (default -- **All** --), a **Start Time** and **End Time**, and one or more **Attribute Criteria** rows of **Attribute**, **Operation** and **Value**; a new blank row appears as soon as you complete the last one.
 - A part must meet **every** complete row.
 - A part is judged on its **latest** reading of each attribute within the window, wherever that reading was taken. If the latest one came from equipment outside the process, or for a material you did not select, the part does not match: a later reading supersedes an earlier one. A part with no reading of the attribute in the window does not match either.
 - The **Value** box follows the attribute's type: **True/False** for a yes/no attribute, a number for a numeric one, text otherwise. Text and yes/no attributes accept only **==** and **!=**; another operator is refused: *"The operator selected for '...' is not valid for that attribute's data type. Text and yes/no attributes support only == and !=."*
 - A row you started but did not finish is refused rather than dropped — *"Criterion N is incomplete. Select an attribute, an operator and a value, or clear the row before searching."* — because a recall run with fewer criteria than you meant would quarantine more parts than you asked for.

Choosing the status

The **Status** is an ordinary product disposition, and its two flags decide what the quarantine does to the parts:

- The seeded **Hold** is *OK to Work* but not *OK to Ship*. Suspect parts keep moving through production — nothing on the line stops — but none can pass shipping verification. It is the usual choice while you investigate, and the only status that raises a **Hold Placed** notification (see [Confirm or cancel](#)).
- A status that is not *OK to Work* also stops each part at its next check-out ([Production Execution](#)). The seeded statuses of that shape (*Closed*, *Scrapped*, *Deleted* and the destructive-audit status) all mean a part is finished with, so a plant that wants a "stop everything" containment usually adds its own status — neither *OK to Work* nor *OK to Ship* — under [Manage dispositions](#).

Confirming changes the parts' status only; their pass/fail results are not touched.

Confirm or cancel

On the confirmation page:

- **Ok** confirms. Every staged part moves to the incident's **Status** in one all-or-nothing step, and each part's previous status is recorded — that record is what a rollback restores. **Items Effected** on the list becomes the number of parts actually held, and the banner reads *"Quarantine confirmed: N part(s) were placed in the selected disposition."* If the database refuses, no part changes: *"The*

quarantine could not be confirmed because of a database error. Nothing was applied - no part's status was changed - and the confirmation can be run again."

- **Cancel** discards the incident and everything entered with it — its criteria, its equipment and material lists and, for a **Black List**, its blacklist entries — and no part changes status: *"The quarantine was cancelled and the incident was discarded."* The audit trail keeps what was discarded.
- Once confirmed, an incident cannot be confirmed or cancelled again: a double-click, or a repost from the browser's Back button, is refused with *"This incident has already been confirmed; it cannot be confirmed or cancelled again."*
- If you leave the page without choosing, the incident stays on the list with nothing held (though a **Black List**'s serials are already on the blacklist), and no page links back to its confirmation. Delete it and raise it again.

Electronic signature. Confirming is the moment parts change disposition, so it is the signed act; the search before it is not. With GxP Mode on, confirming is signed by default, with the meaning *Disposition Decision*; a reason is optional, and the shipped reasons are *Supplier notification*, *Internal nonconformance*, *Customer complaint* and *Other (explain in the comment)*. Cancelling is signed only if your site's signature policy requires it; its meaning is *Rejected* by default, which needs a comment. When either needs a signature, **Ok** and **Cancel** each get their own **Electronic signature** panel — sign in the one above the button you choose. **Show exactly what is being signed** lists every staged serial number (beyond 5,000, their count and a fingerprint of the list), and the signature covers exactly that list: if the staged parts change after the page was displayed, the signature is refused. See [Electronic Signatures & Audit Trail](#).

Notification. When the incident's status is the seeded **Hold**, confirming raises **one Hold Placed notification for the whole incident** — not one per part — by default titled *"Hold placed: (parts)"*. It counts only the parts that were not already on Hold, so it stays silent when every part was held already, and a quarantine to any other status raises none. A publication limited to part of the plant hears about the incident when any held part was last at equipment in that part of the plant. The notification is sent after the quarantine is saved, so a delivery problem can never undo it. To receive it, subscribe to a **Hold Placed** publication ([Notifications](#)).

Work the incident list

The **Quarantine** page lists incidents newest first — **Name, Transaction Type, Status, Date Created, Items Effected, Supplier** (a blacklist's supplier, or the supplier of a genealogy incident's named part) and whether it has been **Rolled Back**. Production supervisors and administrators also see the **Create a new Quarantine Transaction** card and a search box that matches incident names, statuses, a genealogy incident's serial number and supplier names. From an incident you can:

- **Details** — the incident's inputs and history, and an **Electronic signatures** section listing every signature given on it — confirmation, acknowledgements, rollback, edits — with who signed, when, the meaning and the reason ([Electronic Signatures & Audit Trail](#)).

- **Acknowledge Items** (**Affected Items** once rolled back) — the suspect-product list; see [Acknowledge the quarantined products](#).
- **Roll Back** (Production Supervisor or Administrator) — see [Roll back an incident](#).
- **Edit** (Administrators) — changes only the **Name** and **Notes**. The type, status, window and criteria are what the search ran on, so they are fixed: to change them, roll the incident back and raise a new one. Editing is signed if your site's signature policy requires it.
- **Delete** (Administrators) — removes the incident with its suspect list, criteria, equipment and material lists and, for a **Black List**, its blacklist entries; the audit trail keeps a record of what was deleted. Deleting is signed if your site's signature policy requires it (with a reason, by default). It is refused while the incident still holds parts — *"This incident has been confirmed and is holding parts in quarantine. Roll it back before deleting it."* — because its suspect list is the only record of each part's previous status, which is what a rollback restores. A part that a rollback left as it was (see below) still counts as held, so an incident whose rollback left any part as it was cannot currently be deleted.

Acknowledge the quarantined products

Acknowledge Items opens the **Quarantine Suspect Product Index**: every part the incident holds, with the equipment it was **Last Processed At**, when it was **Last Updated**, and its acknowledgement — **Acknowledged Time**, **Acknowledged By** and **Acknowledgement Type**. The header counts how many have been acknowledged; search filters by serial number or equipment name, and each serial number opens that part's Product Report. Anyone signed in can open the list; who may acknowledge depends on the type (below).

An acknowledgement records what the floor actually did with each part — segregated it, told the customer, scrapped it, reworked it, returned it — so the incident carries evidence that the containment was physically acted on, not just flagged in software.

Two kinds of acknowledgement are kept apart. **Contained** says the part has been found and segregated: it only stops product, so the operators who do the segregating record it (**Production** role). Every other type is a **disposition** — a decision about what happens to suspect product, or about letting it go — and is a **release decision** for a **Production Supervisor** or **Administrator**. The **Acknowledgement Type** list offers only the types you may choose, so a Production user sees **Contained** alone.

For an **equipment** incident the list is grouped: each group is headed by the outermost assembly containing a suspect part ("*... — contains N suspect part(s)*"), and clicking the heading expands it. The part that was actually on the suspect equipment carries an **Issue product** badge. You contain the assembly as a unit, and the badge shows which part inside it is the reason. Other incident types list their parts one per row.

Acknowledgement Type	Number (for uploads)	Who may choose it	What it does to the part
Unacknowledged	0	Production Supervisor, Administrator	Clears the acknowledgement. It does not undo the status change made by <i>Scrapped</i> or <i>Destroyed</i> .
Scrapped	1	Production Supervisor, Administrator	Records the acknowledgement and changes the part's status to Scrapped , with an entry in its history.
Customer Notified	2	Production Supervisor, Administrator	Records the acknowledgement only.
Contained	3	Production, Production Supervisor, Administrator	Records the acknowledgement only.
Released – Passed Inspection	4	Production Supervisor, Administrator	Records the acknowledgement only — the part stays in the quarantine status.
Reworked and Verified	5	Production Supervisor, Administrator	Records the acknowledgement only.
Returned to Supplier	6	Production Supervisor, Administrator	Records the acknowledgement only.
Destroyed	7	Production Supervisor, Administrator	Records the acknowledgement and changes the part's status to Deleted , with an entry in its history.
Use-As-Is Approved	8	Production Supervisor, Administrator	Records the acknowledgement only.

Important: Only **Scrapped** and **Destroyed** change a part's status. *Released – Passed Inspection* and *Use-As-Is Approved* record a quality decision, but the part stays in the quarantine status. To release parts, roll the incident back — everything still in the quarantine status returns to its previous status — or change a single part's status on its **Edit** page ([Products](#)). Both are release decisions for a Production Supervisor.

On screen. In **Manual Product Acknowledge**, click the rows to select them, choose an **Acknowledgement Type**, and select **Update**.

By file. For a large incident, work offline. On the **File Upload** card, select the download icon (**Export Results**) to save every part matching the current search as a spreadsheet named after the incident, with the columns **Item**, **Serial Number**, **Alias**, **Last Processed At**, **Last Updated**, **Acknowledged Time**, **Acknowledged By** and **Acknowledgement Type**. Fill in the **Acknowledgement Type** column — the type's name as the export writes it, the name followed by its description as the on-screen list offers it, or

the number from the table above — and upload the file under **Acknowledgement upload (xlsx/csv/txt)** with **Upload**.

- Rows are matched by **Serial Number** alone. The **Alias** column is there for your reference and is ignored on upload; the file carries no internal ids, so it can be worked offline or sent to a supplier and brought back.
- A row whose **Acknowledgement Type** is blank is left alone, and so is one that asks for the acknowledgement the part already has.
- A hand-made .csv or .txt file works too (comma, semicolon, tab or vertical bar between columns): with the headings **Serial Number** and **Acknowledgement Type**, or, without a heading row, the serial number first and the acknowledgement second.
- The banner reports "*N product(s) updated from M row(s) in the uploaded file.*" Rows that could not be applied are listed by row number (the first 20): a serial number that is not on this incident, an acknowledgement with the serial number left blank, or an acknowledgement that is not one of the types above.
- **A file is checked against your role first, as a whole.** If any row asks for a type you may not choose — a Production user's file with one *Released – Passed Inspection* among a hundred *Contained*, say — the whole upload is refused and nothing in it is applied, so a file never goes half in. Split the file, or have a production supervisor upload it.

Electronic signature. Dispositioning is a release decision: with GxP Mode on, every type except *Contained* is signed by default (*Disposition quarantined products*, with the meaning *Disposition Decision*), and the signer must hold Production Supervisor. Marking parts *Contained* is a separate act, *Contain quarantined products*, which ships unsigned; your site's policy may turn a signature on for it, but can never require a particular role or a second signer, so configuration cannot take it away from the floor. When a signature is needed, one **Electronic signature** panel, below the two cards, serves both **Upload** and **Update**, and one signature covers everything that submission changes — each part and the acknowledgement it is given. IntelliTrack decides which act a submission is from what it asks for, not from the page: a submission that only contains parts is *Contain quarantined products*; one with any other type in it — a supervisor's file that marks some parts *Contained* and releases others, say — is *Disposition quarantined products* as a whole, and is signed as the release decision it is.

Once an incident has been rolled back its list is read-only: the acknowledgement cards are gone, and **Export Results** moves to the **Suspect Products** card.

Roll back an incident

Roll back when a quarantine turns out to be unnecessary — the equipment checks out, the supplier withdraws its notice, the criteria were wrong — or when the parts that needed a decision have had one and the rest can go back to where they were. **Roll Back** (Production Supervisor or Administrator) opens **Quarantine Transaction Rollback**; hovering over its title explains the rule. Select **Roll Back** to confirm. A rollback releases suspect product in bulk, so it is a **release decision**.

- Every part still in the quarantine status returns to the status it had before the incident. Pass/fail results are not changed.
- A part whose status has moved on since — scrapped or destroyed through its acknowledgement, or changed on its **Edit** page — is left as it is: someone has already decided about it, and a rollback must not overrule them. The banner gives both numbers — *"Rollback complete: N part(s) were restored to their original disposition; M part(s) were left as-is because their disposition changed after the quarantine."* — and shows as a warning when any were left.
- It is all-or-nothing: on a database error nothing is released (*"The quarantine could not be rolled back because of a database error. Nothing was released - no part's status was changed - and the rollback can be run again."*).
- It is one-shot, and only for a confirmed incident: *"This incident cannot be rolled back: it was never confirmed or has already been rolled back."*
- A **Black List** incident's serials stay on the blacklist after a rollback (see [Raise a quarantine incident](#)).

With GxP Mode on, a rollback is signed by default, with the meaning *Disposition Decision*, a **Reason** is required, and the signer must hold Production Supervisor — an administrator without that role cannot sign it.

Warning: Rolling back releases every part the incident still holds, in one step. Confirm you intend to release the whole population; to release only some parts, change those parts' status individually instead.

Manage dispositions (product status codes)

Dispositions are the status codes applied to products — and, in groups of their own, to suppliers, work orders and customers — and their flags are read by every gate in the system. They live in the shared configuration database, so one set serves every site: a change is saved there at once, and each site's production and shipping checks use it from that site's next **Configuration** → **Site** synchronisation ([System Configuration](#)).

1. Open **Disposition Codes** (from Additional Configuration → Quality & Change Control). Anyone signed in can view the list and each code's **Details**; search filters by name, label or description, and results are paged.
2. **Administrators** can select **Create New**, and **Edit** or **Delete** the codes their plant added. The codes IntelliTrack ships with — *Released*, *Hold*, *Scrapped*, the supplier and work-order statuses and the rest — show no **Edit** or **Delete**.
3. **Create New** always creates a **Product** status, numbered automatically. Give it a **Name**, an optional **Label** and **Description**, and its switches (**OK To Work**, **OK To Ship**, **Include In Count**, **Available To Users** — see the [Field Reference](#)). Tick **Available To Users**, or the new status is offered in no status list — not for a quarantine, a product's status or a receipt.

4. **Edit** changes the **Label**, **Description** and the four switches; the **Name** and **Type** are fixed. **Display Name** previews how the status reads in lists. Saving with nothing changed writes nothing.
5. **Delete** retires the code: it leaves this list and every status list, and each product that already carries it keeps it.

With GxP Mode on, creating, editing and deleting a disposition are signed if your site's signature policy requires it (meaning *Approved*, by default); what is signed is each setting before and after. A disposition's **Details** page shows its **Electronic signatures**.

Warning: Because these flags drive production and shipping — at every site — changing a disposition's rules changes how *every* product with that status behaves. For example, turning **OK To Ship** off for a status starts holding every product in it at shipping verification, at each site as soon as the change has synchronised there.

Shipping verification

Confirming a finished product is **OK to Ship** (and getting its shipping code) is part of production; see [Production Execution → Verify a product is OK to Ship](#). It reads the same OK-to-ship flag these dispositions set — for the product and for everything in its genealogy, so a component held by a quarantine also stops the assembly it was built into.

Field Reference

Quarantine incident:

Field	Required?	What it does / valid values
Type	Yes	Chosen on Quarantine before Create New: Black List, All Product from Selected Equipment Module(s), All Products in Genealogy or All Product with Matching Performance Criteria — how suspects are found. Fixed once created.
Name	Yes	Up to 200 characters; editable later. A shared reference (supplier RMA, corrective-action number) here ties related incidents together across sites.
Status	Yes	The product disposition applied to every matched part; the list shows each one's <i>OK to Work</i> / <i>OK to Ship</i> . Fixed once created.
Notes	No	Editable later — a good place to record the history-only count.
Supplier	Black List	The supplier the blacklist entries belong to.
File / Manual Serial Number List	Black List (at least one serial)	.xlsx, .csv, .tsv or .txt, and/or pasted serials separated by comma, space, semicolon, tab or new line; entries over 200 characters are ignored.

Field	Required?	What it does / valid values
Equipment Module	Equipment (at least one)	Chosen through Enterprise → Site → Area → Work Cell .
Start Time / End Time	Equipment, Performance	The window searched; the start must be before the end.
Serial Number	Genealogy	Up to 50 characters: the exact serial, else a unique alias at this site, else a former serial. A blank serial finds nothing.
Process	Performance	Only measurements taken on this process's equipment count.
Materials	No	Performance only; default -- All --.
Attribute / Operation / Value	Performance (at least one complete row)	Every complete row must match; text and yes/no attributes accept only == and !=.

Disposition (create/edit):

Field	Required?	What it does / valid values
Name	Yes	Up to 82 characters. Set when the status is created; fixed afterwards.
Label	No	Up to 100 characters. Shown instead of the name wherever the status is listed.
Description	No	Up to 200 characters. Shown after the label in most status lists.
Type	—	Shown as Object on the list: <i>Product</i> , <i>Supplier</i> , <i>WorkOrder</i> or <i>Customer</i> . A status created here is always a <i>Product</i> status.
OK To Work	No	Whether products with this status may be processed (the production gate).
OK To Ship	No	Whether they pass shipping verification. With OK To Work off, it also counts them as completed good output on their work order.
Include In Count	No	Whether products with this status are counted in the plant database's inventory totals (by location and by equipment). The web Inventory Reports do not use it.
Available To Users	No	Whether the status is offered in status lists — a quarantine's Status , a product's status, receiving. Off keeps it off new choices without changing products that already carry it.

Related Topics

- [Products](#) — status on a product, and the blacklist at creation
- [Production Execution](#) — the OK-to-Work / OK-to-Ship gates
- [Work Orders](#) — how "good" status counts toward fulfillment
- [Reports](#) — the reporting database the quarantine search reads, and the **Quarantine Aging** report of open incidents

- [Where-Used / Recall Impact](#) — every downstream product built from a suspect serial or lot, to size a recall before you raise an incident
- [Materials & Suppliers](#) — supplier default dispositions
- [Notifications](#) — the **Hold Placed** notification a confirmed quarantine raises
- [System Configuration](#) — the Site Id and the synchronisation jobs the search depends on
- [Electronic Signatures & Audit Trail](#) — signing a quarantine decision, and reading an incident's signatures

Chapter 11: OEE (Overall Equipment Effectiveness)

Overview

IntelliTrack watches equipment **states** over time, groups them into **events**, and turns them into OEE metrics — **Availability × Performance × Quality** — plus downtime, fault, reject, and reliability (MTBF/MTTR) reporting. This chapter covers reviewing OEE reports, managing machine alarms, reviewing and classifying OEE events, and the configuration (modes, states, equipment timing, and user-defined fields), maintained by engineers and administrators, that makes the numbers meaningful.

Why this matters: OEE is only as honest as its **event classification**. A five-minute stop attributed to "planned changeover" affects availability very differently than one attributed to "unplanned fault." The whole point of the event-review workflow is to get that attribution right, because it flows straight into every availability number and every downtime Pareto a manager uses to decide where to spend improvement effort.

Before You Begin

- **Viewing OEE reports, the alarm pages and OEE events** is available to **any signed-in user**.
- **Acknowledging an alarm** or **requesting work** on the **Active Alarms** board, and **assigning a reason** to an OEE event nobody has classified yet, require the **Production** role (production supervisors and administrators can do them too): they are the operator's own record of what happened. The alarm buttons are shown to everyone; pressing one without the role is refused. Recording the start and end of a downtime by hand, through IntelliTrack's downtime requests (no page makes them in this release), needs the Production role as well.
- **Changing a reason another person already assigned** requires the **Production Supervisor** or **Administrator** role. The first coding of a stop is the operator's; re-coding someone else's moves that downtime between OEE loss buckets after the fact, which is a supervisor's correction. The **Edit** link is not shown on an event you may not change.
- **Configuring** modes, states, equipment OEE timing, and UDFs, and maintaining a machine's **alarm definitions**, requires the **Engineering** or **Administrator** role. The four OEE configuration pages open only for those roles; alarm definitions can be viewed and exported by anyone. Engineering is the role for the controls and process engineers who commission machines: it lets them set up OEE and load a machine's alarm list without the rest of an administrator's rights — security, system parameters, licensing and signature policy.
- OEE data comes from the reporting database, populated from equipment activity; event detection uses each equipment's **event recovery** and **event start delay** times, set on the equipment's own page ([Configure Equipment](#)).
- Dates and times on the OEE pages and charts are written in the site's date and time format, which an administrator chooses for everyone ([Date and time format](#)); the **Alarm Journal** keeps a fixed

year-month-day form.

How to Get Here

- OEE reports and the alarm pages: **Reports** → **OEE Reports**, which opens a page of report tiles.
- OEE event review: **Production** → **OEE Events** (the same page is the **OEE Event Review** tile on the OEE configuration page).
- OEE configuration: **Configuration** → **Additional Configuration** → **Equipment & Processes** → **OEE Configuration** ([System Configuration](#)).

From machine activity to OEE

From machine activity to OEE



A mode's "scheduled" flag decides whether downtime is planned or unplanned; a state's flags decide what counts as running.

Reviewing an event and assigning its reason is what makes the downtime Pareto and availability numbers trustworthy.

Run OEE reports

Reports → **OEE Reports** opens a page of tiles, one per report. Most reports let you pick equipment from the plant model, a **time range**, and (where applicable) **user-defined field (UDF)** filters; the fault and alarm reports also filter by **Severity** and alarm **Category**:

Report	What it tells you
OEE Scorecard	Availability × Performance × Quality per equipment, ranked side by side, with an overall trend — the headline number.
OEE by Hour	OEE over time, by hour — where in the shift it dipped.
Product Produced per Hour	For one machine, how much good, in-process and rejected product it made in each hour.
OEE Loss Analysis	How the lost time splits between Availability, Performance and Quality loss over the window — which of the three to work on first.

Report	What it tells you
Downtime Pie Report	Downtime by machine state (stopped, faulted, held...) — where the time went.
Fault Pareto Report / Fault Detail Report	Faults ranked by how often they happened — or, with Duration ticked, by minutes lost — for 1 to 20 codes (10 unless you change Fault Items), and the detail behind them grouped by machine and code. One line per alarm, however long it ran: a fault that lasts a shift is one alarm, not one per hour. The Pareto can also be narrowed by Product Impact .
Reject Detail Report	Each rejected part — when, its serial number, material and machine, and the attributes that failed it — the Quality drag. It also opens, in a new tab, when you click a bar of the Reject Pareto on the Performance reports (Reports).
State Transition Diagram	Every state each machine was in, on a zoomable timeline — what happened, when, and in what order.
State Detail Report	The same timeline for one chosen state, with the operator reasons recorded against its events.
Cycle Time Report	A histogram of measured cycle times, optionally without outliers — the Performance drag.
MTBF / MTTR	Reliability: mean time between failures / to repair. A failure is an entry into a <i>faulted</i> state. The same report shows stops beside it — every interruption, including idle, suspended and held — because those are interruptions to plan, not failures of the machine, and mixing them understates MTBF.
Active Alarms	What is alarming right now, with acknowledge and request-work. The only report here that does not look backwards.
Alarm Health	Whether the alarm system is readable at all: rate, floods, chattering, stale alarms and the top-ten share, against the ISA-18.2 figures.
Alarm Heat Map	When the plant alarms -- day of week against hour of day, coloured by rate rather than by raw count.
Alarm Journal	Every raise, acknowledgement, clear, shelve and work request in time order -- the sequence of events a shift review reads.
Fault Reliability	MTBF and MTTR per fault code, with the worst single occurrence beside the mean.
Downtime by Reason	Downtime by the reason an operator gave, including how much of it nobody explained.
OEE Event Log	Every OEE downtime event across the chosen machines and window, with the reason and notes an operator recorded; tick Annotated events only to see just the explained ones, and Export Event Log CSV to take them away.

See [Reports](#) for the shared run/export/history controls. Every run is saved to report history, so each has a share link and can be put on a [dashboard](#).

The events behind the numbers. The OEE Scorecard, OEE Loss Analysis, Downtime Pie Report, Fault Pareto Report and State Detail Report end with a folded **OEE Events** section: the downtime events inside the report, each with the reason and notes an operator gave and who gave them. Open it when a number surprises you — an explained stop and an unexplained one look the same in a total.

Drilling down. Some charts open the report that explains them, for the same machines and window, as an ordinary run with its own share link:

- On **OEE Loss Analysis**, click Availability to open the Downtime Pie Report, Performance to open the Cycle Time Report, or Quality to open the Reject Detail Report (each in a new tab).
- On the **Downtime Pie Report**, click a faulted slice to open the Fault Pareto Report ranked by minutes lost, or any other slice to open the State Detail Report for that state.
- On **MTBF / MTTR**, the icons on a machine's row open the Fault Pareto Report or the State Transition Diagram for that machine alone.

How OEE counts production

OEE counts each processed event by the **quantity it carried**, not as one unit per row: a 480 kg lot counts as 480 of its unit, a single piece as one. On a discrete line, where every event carries a quantity of one, nothing changes; on a batch or process line, Performance and Quality now reflect how much was actually made.

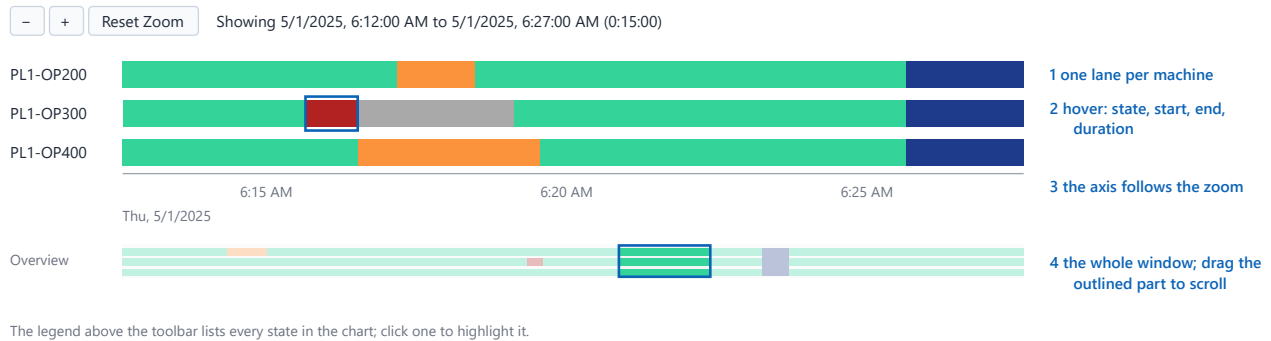
A machine that cannot be scored is not averaged in silently. The **OEE Scorecard** lists such machines above its table — *These equipment modules were not scored, so they are absent from the figures below* — each with the reason: its ideal rate (the **Ideal Cycle Time** on the equipment, Chapter 2) is not set, or its quantities are recorded in a unit that cannot be converted to the unit its ideal rate is stated in. Fix the machine's setup and re-run; until then its absence is deliberate, not a machine with no output.

Read the state timeline

The **State Transition Diagram** and **State Detail** reports draw the raw state history as a timeline: one lane per equipment module, and one coloured bar for every period the machine spent in a state, in the colours set on **Manage OEE States**.

Why this matters: the other OEE reports add time up — how much downtime, how many faults. The timeline keeps the *order*. It answers "what happened just before the fault", "did the whole line stop together or one machine at a time", and "is this machine flickering between executing and held all shift" — the patterns a total cannot show. Choose a few related machines (a line, or a workcell) so their lanes sit together and a stoppage that travels down the line is visible as it happens.

Parts of the state timeline



Zoom in and move through time

The chart opens showing the whole report window. To look closer:

To...	Do this
Zoom in or out around the pointer	Hold Shift and turn the mouse wheel. On a touch screen, pinch.
Zoom in on one spot	Double-click it (Shift + double-click zooms back out).
Move earlier or later	Drag the chart sideways, or swipe sideways on a trackpad.
Jump to any part of the window	Click the overview strip under the chart — that moment moves to the middle.
Choose an exact range	Drag across the overview; drag the outlined window's edges to widen or narrow it.
Use the keyboard	Click the chart, then ←/→ to move (Shift for bigger steps), +/- to zoom, 0 to see the whole window again.
Start over	Reset Zoom.

This timeline is drawn on its own canvas rather than by the reporting charts, so its gestures are its own: the other report charts zoom by dragging a rectangle and reset from their toolbar ([Reading a chart](#)).

A plain turn of the mouse wheel still scrolls the page, so the chart never traps you while you read the rest of the report. The line above the chart always says exactly which stretch of time is in view.

When a bar is thinner than a pixel

A busy machine records a new state every few seconds — ten thousand periods in a day — so until you zoom in, most periods are narrower than one pixel on screen. Each pixel is then painted in the state that held **most of that pixel's time**, and hovering it lists everything inside it (for example "5 state periods from 12:24:06 PM to 12:26:24 PM — Executing 73.5%, Held 26.5%"). Zoom in and the periods separate. This is why the colours of a crowded stretch can change as you zoom: you are seeing more detail, not different data.

Find every occurrence of one state

Click a state in the legend to **highlight** it. Every other state fades, and the highlighted one is drawn in every pixel where it occurs at all — at least a few pixels wide — so a two-second fault in a week-long report stays visible without zooming. The overview strip shows the same highlight, which makes it a map of *where* the state happened. Click the state again (or press **Esc** in the chart) to show all states.

Tip: Two states with the same colour cannot be told apart on the chart. Give the states you care about distinct colours on **Manage OEE States** (see below); the change applies to every report run afterwards.

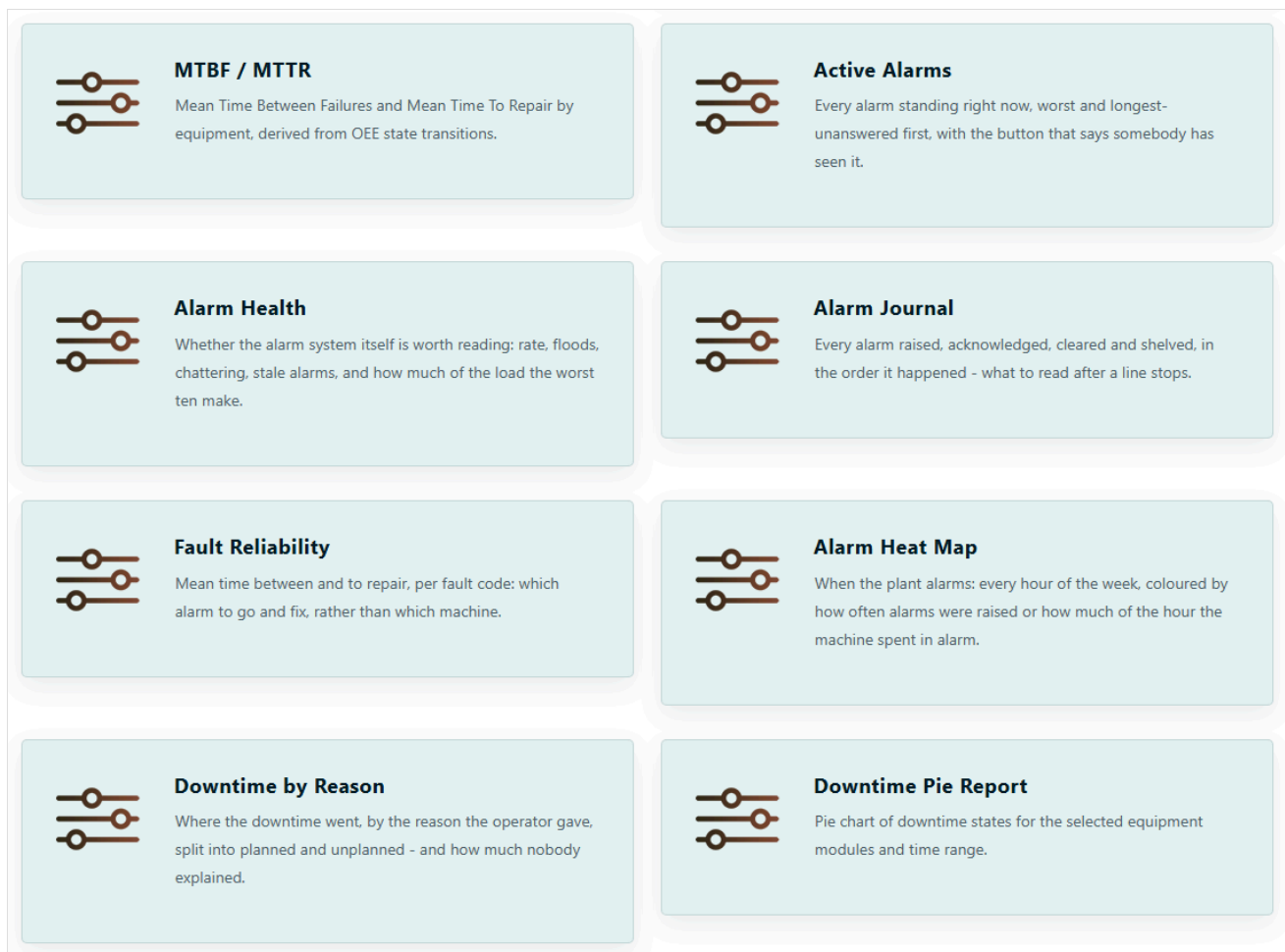
Times, share links, and dashboards

- Times on the axis and in the pop-ups are in the plant's time zone — the zone the report's **Start/End** fields are entered in — whichever time zone your own computer is in. Dates on the axis, in the pop-ups and in the line above the chart follow the site's date format, like the other dates on IntelliTrack's pages.
- The report's **share link** reopens the same timeline, fully zoomable, for whoever you send it to.
- On a **dashboard**, the timeline fills the screen under the item's name and shows the whole window, its lanes and text sized to be read from across the room. The zoom controls and overview strip are left off there; hovering still shows each state's times.
- On the State Transition Diagram, **Show Data Table** lists every period with its exact start, end, and minutes, for export.

Manage alarms

An OEE report tells you the machine stopped. The alarm pages tell you **which alarm stopped it, whether anybody answered, and whether the alarm was worth raising at all.**

That last question is the one most plants never get to ask. An alarm system fills up over years: every commissioning engineer adds the alarms they needed that week, nobody removes any, and the operator ends up reading three hundred messages a shift and acting on two. The international standard for this (ISA-18.2 / IEC 62682) puts numbers on it — about six alarms an hour is comfortable, twelve is manageable, and the ten most frequent alarms should be a few per cent of the load, not a third of it. These pages measure your plant against those numbers and give you the short list of fixes that would move them.



Reports → OEE Reports: the alarm pages in the tile list — Active Alarms, Alarm Health, Alarm Journal, Fault Reliability, Alarm Heat Map and Downtime by Reason.

What is wrong right now

Reports → OEE Reports → Active Alarms is the only page here that does not look backwards. It opens already showing every alarm standing at this moment across the plant; narrow it with the plant picker, **Severity** and **Category**, then **View Report**. Unanswered alarms come first, the most severe at the top and, within a severity, the longest-standing first — so the top line is the alarm that has stood longest with nobody touching it. Each row shows:

- **how long it has been standing (Standing)** — and a red **Overdue** mark, with the row highlighted, when it has gone past the response time the alarm's own documentation allows and nobody has acknowledged it;
- **what to do (What to do)** — the operator action recorded against the fault code. **Not documented** in this column is a finding in itself: it means nobody has ever written down what this alarm means. Hover the cell to read the consequence, where one is written;
- whether it was the **First Out** alarm — the one that started a cascade;

- **who acknowledged it (Acknowledged)**, if anyone.

Alarms that have been **shelved** — deliberately set aside for a stated time, with a reason — are left off the board unless you tick **Include shelved alarms**, and then carry a **Shelved** badge (hover it for the reason). Shelving is recorded when it happens, but no page in this release shelves an alarm.

Two buttons act on a row. Both need the **Production** role (or Production Supervisor or Administrator):

Button	What it does	What it does <i>not</i> do
Acknowledge	Records that somebody has seen the alarm, with your user name, and moves it below the unanswered ones.	Clear the alarm. The machine is still in it.
Request work	Raises a maintenance work request from the alarm, records it in the alarm's history (the Alarm Journal), and — where an Alarm Work Request Raised publication covers that category of fault — notifies its subscribers.	Create a job in your CMMS. IntelliTrack keeps the request, but no page in this release records the CMMS ticket number or closes the request.

Once a request is open, the row shows **Work requested** in place of the button (hover it for who asked and when). A second request for the same alarm — from a screen that still showed the button — joins the request that already exists: a line with four people watching it raises one request, not four.

Running the board with **View Report** saves it to report history like any other report, so it can be put on a [dashboard](#). The tile does not replay the rows of that run: each time it comes round it shows what is standing at that moment.

Equipment	Fault Code	Description	Severity	Category	Standing	What to do	First Out	Acknowledged
DEMO_OP100	101	Feeder jam	Error	Equipment	15:20 Overdue	Stop the infeed, clear the fallen container, restart. If it recurs within the hour, call mechanical: the guide rail is out of adjustment.	First Out	Acknowledge Request work
DEMO_OP300	303	Ribbon low	Information	Material	50:01:20 Overdue	Fit a fresh reel at the next planned stop.	First Out	Acknowledge Request work
DEMO_OP300	301	Label web break	Error	Equipment	38:20	Rethread the web, check the splice, and pull back to the last labelled bottle.	First Out	m.delgado Work requested
DEMO_OP200	201	Guard door open	Error	Safety	07:20	Close the guard and reset at the local panel. Report any guard that opens on its own -- that is a fault, not an operator action.	First Out	j.okafor Request work

Machine time is shown for reference only: it is what the controller said. Every duration, order and window on this page is measured from the times this database recorded, because a controller clock drifts and IntelliTrack cannot tell when it has.

The Active Alarms board. The two unanswered alarms come first, both overdue; below them, one already turned into a work request and one acknowledged but still standing.

Why it matters: before this page, the only way to know what was wrong with a machine two buildings away was to walk to it and read its HMI. A supervisor could not see that a line had been standing on one alarm for forty minutes.

Is the alarm system worth reading?

Alarm Health measures the alarms themselves rather than the faults behind them, machine by machine, against the ISA-18.2 figures:

Measure	What it means	What good looks like
Alarms per hour	How often the operator is interrupted	About 6; 12 is the practical limit
Flood periods	Ten-minute windows with more than ten alarms	Under 1% of periods
Chattering alarms	Alarms that re-raise within a minute of clearing	Near zero
Stale alarms	Standing more than 24 hours	Zero — a permanently standing alarm is furniture
Fleeting alarms	Gone again within five seconds — before anybody could act on them	—
Top-ten share	What share of all alarms the worst ten are	A few per cent

The ISA-18.2 figures are stated per operator position, and a machine is not an operator, so the page says to read them as a yardstick rather than a pass mark. Beneath the table, **Where the noise comes from** names the ten worst alarms across your selection, with each one's share of the load. This is the page to open before an alarm rationalisation meeting: it turns "the alarms are bad" into ten named fixes and a number that says what fixing them is worth.

When does the plant alarm?

Alarm Heat Map is a seven-by-twenty-four grid — day of the week against hour of the day — coloured by how often alarms were raised, or by how much of each hour the machines spent in alarm.

It answers a question no Pareto can. An alarm that is evenly spread is a machine problem; one that piles into Monday 06:00 is a **procedure** problem — a shift handover, a cold start, a cleaning cycle, one crew — and nothing in the alarm record names it.

The cells show a **rate**, not a raw count, because a window rarely covers each weekday the same number of times: a ten-day window holds two Mondays and one Thursday, and raw counts would make whichever day came round twice look like the problem. A cell the window never reached is left blank rather than shown as quiet.

Choose what the colour shows with **Colour by** — **Alarms raised per hour** or **Share of the hour in alarm** — and hover a cell for its numbers and its most frequent code. Beneath the grid, **Worst hours** lists the

hottest cells with their shift and top code, and **By shift** totals the window per shift. The shifts are fixed — Day 00:00-07:59, Swing 08:00-15:59, Night 16:00-23:59, the same as the Shift Performance report — whatever your plant's own shift times are.

Cells show a RATE, not a raw count: a window rarely covers each weekday the same number of times, and a raw count would make whichever day came round twice look like the problem.
Hover a cell for its numbers and its worst code.

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	Total
Sunday	0.4	0.2	0.6	0.2	0.2	0.6	0.2	0.2	0.6		0.6	0.2	0.2	0.6	0.2	0.6		0.2	0.6	0.2	0.4	0.4	0.2	0.6	41
Monday	0.2		0.6	0.2	0.6	0.2	5	0.6		0.8		0.2	1.7	0.3	0.5	1	0.3	1	0.3	0.3	1	0.3	0.8	0.3	73
Tuesday	0.3	1	0.3	0.3	1	0.3	1	0.3		1	0.3	1	0.3	0.3	1		0.8	0.5	0.3	1	0.3	0.3	1	0.3	49
Wednesday	0.8	0.3	0.3	1	0.3	1	0.3		1.3		0.3	1	0.3	1	0.3	0.3	0.8	0.3	1	0.3	0.3	1	0.3	0.5	49
Thursday	0.8		1	0.3	0.3	1	0.3	1		0.3	1	0.3	4.3	0.5	0.3	1	0.3		1	0.3	1	0.3	0.3	1	64
Friday		1	0.3	0.3	1	0.3	0.5	0.8	0.3	0.8	0.3	0.3	0.9	0.2	0.8	0.2		0.8	0.2	0.8	0.2	0.2	0.8	0.2	48
Saturday	0.2	0.6	0.2	0.8	0.2	0.2	0.8	0.2	0.6	0.2	0.6	0.8	0.2	0.4	0.6	0.2	0.8		0.2	0.8	0.2	0.6	0.2	0.2	49
Total	11	13	14	13	15	15	38	13	12	13	13	16	33	14	16	14	12	12	15	16	14	13	15	13	

Worst hours

When	Shift	Occurrences	Per hour	Minutes	Most frequent code
Monday 06:00	Day	25	5.00	145.5	202 — Cap misfeed (24)
Thursday 12:00	Swing	17	4.25	27.8	102 — Low air pressure (7)
Monday 12:00	Swing	8	1.72	114.3	102 — Low air pressure (3)
Wednesday 08:00	Swing	5	1.25	7.1	102 — Low air pressure (3)
Monday 15:00	Swing	4	1.00	9.7	102 — Low air pressure (3)

By shift

Shift	Occurrences	Per hour	Minutes
Day	132	0.53	1,438.5
Night	110	0.44	1,341.8
Swing	131	0.53	1,563.2

Shifts use the same fixed boundaries as the Shift Performance report: Day 00:00-07:59, Swing 08:00-15:59, Night 16:00-23:59.

An Alarm Heat Map over 30 days. The alarms pile into Monday 06:00, and Worst hours ranks that hour first with its most frequent code, 202 Cap misfeed: a procedure to look at, not a machine.

What happened, and in what order

Alarm Journal is the sequence of events: every raise, acknowledgement, clear, shelve and work request in time order, with who did it (**By**) and any **Note**. Narrow it by **Transition**, or tick **Only the first alarm of each cascade**. It is what a shift review or an incident investigation reads. It lists up to 5,000 transitions from the start of the window, and says so when it stops there: the latest transitions are the ones left out, so narrow the window or the filters if a busy period is cut short.

The list shows a page at a time. Move with **First**, **Previous**, **Next** and **Last**, type a page number, or choose how many rows a page holds under **Items Per Page** (it opens at your own list-page size). Paging never re-runs the report, and sorting a column sorts the whole list, not just the page on screen, then returns you to the first page. **Export** downloads the whole list — every page — as a CSV file with the same columns, for a spreadsheet or an investigation record.

Where the machine reported its own timestamp, that is shown as **Machine time** beside the time IntelliTrack received the message — for reference only. Every duration, order and window on the alarm pages is measured from the times IntelliTrack recorded, because a controller's clock drifts, can be set by

hand, and IntelliTrack cannot tell when it is wrong. Use **Machine time** to line an event up against the controller's own log, not to time it.

Fault Reliability gives MTBF and MTTR **per fault code**, not just per machine, with the worst single occurrence (**Longest**) beside the mean — because a mean hides the four-hour one. It lists the 25 codes that cost the most time in the window.

Downtime by Reason splits downtime by the reason an operator gave, not by machine state. The first number to look at is **Unexplained downtime** — the minutes with no reason recorded: every other number on the page is only as good as that one is small. Each reason shows whether it is **Planned**, which of the six big losses it belongs to (equipment failure, setup and changeover, minor stops and idling, reduced speed, process defects, start-up and yield), and its events, minutes and share. The reasons are the ones chosen on **OEE Events** (see [Review and classify OEE events](#)).

Have the alarms delivered to you

Four publications send alarm information without anybody opening a page. Configure them under **Configuration** → **Notifications** → **Publications** and subscribe the people who need them (see [Notifications](#)); each can be limited to part of the plant:

Publication	When	What it says
Machine Alarm Raised	The moment a machine raises an alarm	Which machine raised which alarm. Out of the box it watches only Error alarms, reports only the first alarm of a cascade (First-out alarms only), and stays quiet for 15 minutes about the same alarm on the same machine (Cooldown Minutes), because a publication that mails every alarm is switched off within a day. Narrow it further by Alarm categories and Minimum product impact .
Alarm Work Request Raised	When somebody presses Request work on the Active Alarms board	The request, to the people who look after that kind of fault — choose them with Alarm categories .
Shift Alarm Summary	Every 8 hours with the Recurrence it is created with (<i>Shift</i>); <i>Hourly</i> , <i>Daily</i> and <i>Weekly</i> are the other choices	One message: how many alarms, how long, the worst machine and code, and what is still standing unacknowledged as the next crew arrives. Lookback (hours) — 8 by default — sets how far back it reads.
Alarm Bad Actors Digest	Weekly (Recurrence Weekly), reading back Lookback (days) , 7 by default	The worst alarms (10 unless you change Top N) and what share of the whole load they were.

The two digests stay silent when there is nothing to report — a shift with fewer alarms than **Minimum alarms**, or a period with no alarms, or one where the worst alarms are a smaller share of the load than

Minimum share of the load (%). A message that says the shift was quiet is how a publication teaches people to filter it away unread.

A scheduled publication first runs within about a minute of being created and then once per period from that moment — there is no shift calendar behind it. To have the summary arrive at a shift change, create it at a shift change.

Document what an alarm means

An alarm is worth raising only if an operator can do something about it. Recording what that something is — **rationalisation** — is what separates an alarm system from a list of messages.

Each fault code carries four fields beyond its description:

Field	What to write
Cause	Why the machine raises it, in the words of somebody who knows the process
Consequence	What happens if it is left — what the operator is being protected from
Operator action	What to do . The field that decides whether the alarm was worth raising
Response time	How long the operator has, in seconds, before the consequence lands

They reach two places: the **Active Alarms** board, and the operator at the machine — an HMI can ask IntelliTrack what one alarm means and show the cause, the consequence and the action on the screen in front of them.

Editing them: each machine's alarm definitions have their own pages — **Alarm Definitions**, reached from the equipment's Details or Edit page. One definition is edited there directly (Engineering or Administrator role); a whole list is loaded from a spreadsheet with **Import**, which previews what it would add, change and retire before anything is saved. See [Manage alarm definitions](#) for both, step by step. The file carries up to ten columns:

```
Code,Description,Severity,Category,ProductImpact,Cause,Consequence,OperatorAction,Response  
Seconds,ResponsibleGroup
```

Everything after the third column is optional, and a file written for an older version still imports. A narrower file, or a blank cell, never erases a rationalisation somebody wrote by hand — but it does set **Category** and **Product Impact** back to 0 for the codes it lists, so keep those two columns in any file you re-import.

Copying a rationalised list to another machine: on the machine you documented, choose **Export (Excel)**, then **Import** that workbook on the next machine of the same make. The work is done once, and the export is always the current list.

There is no page for the lists themselves: the alarm **categories** (shipped as 0 Unclassified, 1 Safety, 2 Equipment, 3 Process, 4 Quality, 5 Material, each with a default owning group) and the downtime reasons

(see [Review and classify OEE events](#)) are set up in the reporting database when it is installed or upgraded. An upgrade only adds entries that are missing; it never changes one that exists.

Reasons for classification changes

A fault code's **Product Impact** says whether its alarm belongs on the record of the **product** being worked when it was raised: *Not on the product record* (0), *On the product record* (1), or *On the record, and holds product* (2). It is copied onto every alarm the code raises, and the Fault Pareto Report and the **Machine Alarm Raised** publication can filter on it. It also decides which alarms appear on a product's record: the Product Report's **Machine Alarms During Processing** section and the **Product-Impacting Alarms** quality report ([Reports](#)) list the alarms of 1 and 2 that were active while each unit was being worked — as each alarm was classified **when it was raised**, so changing a code later never rewrites a record already made. Nothing holds product automatically: a *holds product* alarm is shown in red on those reports, and containment stays a decision made on the Quarantine page. Because it is the one classification that speaks for the product, a change to it is treated as a controlled change.

So any change that would move a code's Product Impact to or from 1 or 2 — editing the definition, retiring or reinstating it, or importing a list over it — asks for a **Reason for the change**, and is refused without one: nothing is saved. An import's preview names the codes that need it. The reason is kept in the definition's **Change History** in the reporting database, beside what changed, when, and who did it. It is a recorded reason, not an electronic signature: no signature policy covers alarm definitions in this release, whether an engineer or an administrator makes the change.

Keeping alarm records past the purge

IntelliTrack prunes old alarm records to keep the database a working size: the reporting database's maintenance purge deletes every alarm record older than the **Report History Retention Time** (System Parameter 12; 30 days as shipped). The same purge removes the machine state history — the periods the state timeline draws — on the same window, and that is never archived. A site with a records-retention obligation needs the alarm records to survive the purge.

An administrator sets **Alarm Archive Retention Time** (**System Parameter 17**, on the **System Parameters** page — Configuration → Additional Configuration → Security & Administration; see [System Configuration](#)) to the number of days your policy requires. Every alarm record is then copied to the alarm archive before the purge deletes it — its code, machine, severity, category, times and who acknowledged or shelved it — and the copy is kept until it is that many days older than the alarm's own start. Left at **0**, as shipped, nothing is archived and records are deleted as before. The value can be 0 to 36,500 days.

Set it **longer than the Report History Retention Time**: the archive keeps a copy only until the alarm is that many days old, so an archive window no longer than the purge window deletes each copy as soon as it is made. Changing a system parameter is signed if your site's signature policy requires it (see [Electronic Signatures & Audit Trail](#)).

Effect to expect: the archive grows roughly in proportion to your alarm rate and the retention you choose. Turn it on deliberately, not by default. The archive is kept in the reporting database for records

retention; the alarm pages report only on the live records. The product record is the exception: the Product Report and the Product-Impacting Alarms report read the archive as well, and a product is kept for the **Minimum Product Retention Time** (System Parameter 10; 180 days as shipped). **If your quality system relies on the alarms on a product's record, set System Parameter 17 to at least System Parameter 10** — otherwise a product's record loses its alarms months before the product itself, and the report can only say that they may be missing.

Review and classify OEE events

This is where you make the numbers honest.

1. Open **Production** → **OEE Events (OEE Event Review)**.
2. Select one **equipment** module and **Show Events** (your last selection is remembered and loads by itself next time).
3. The list shows the machine's downtime events, newest first: the state each began in (**Initial State**), its **Duration**, and whether it is still **Open** or **Closed**. Expand a row for its **Start Time**, **Stop Time** and **Reason Code** (hover the reason to read the note). **Details** shows the whole event, including its mode and who last classified it.
4. With the **Production** role, **Edit** an event and choose its reason from the list — a sub-reason reads as its parent and itself, such as *Setup / Changeover - Tool Change* — and add **notes** where they help. A reason is required to save, and some reasons (such as *Other*) also require a note. Saving re-attributes that downtime in every report and records you as the person who classified it.

A Production user can code an event that has no reason yet, and change a reason they gave themselves. An event that already carries a reason recorded under someone else's name can be re-coded only by a **Production Supervisor** or an **Administrator**; for anyone else its **Edit** link is not shown. That keeps each classification the considered answer of the person who gave it: an operator's explanation of a stop cannot be quietly overwritten by the next shift, and a supervisor who corrects it is recorded as the person who classified it.

The reasons are a configured list rather than free text, and each one carries whether the stop was **planned** and which of the six big losses it belongs to — which is what **Downtime by Reason** reports on. As shipped the list is: *Equipment Failure* (with *Unplanned Maintenance*), *Setup / Changeover* (with *Tool Change*), *Material Shortage*, *Waiting For Operator*, *Operator Break*, *Quality Issue*, *Planned Maintenance*, *Scheduled Downtime* and *Other*; of these, setup and changeover, tool change, operator break, planned maintenance and scheduled downtime count as planned. The event keeps both the reason chosen and the words it had when it was chosen, so renaming a reason later does not rewrite last year's stops.

Tip: Unclassified or mis-classified events are the usual cause of an OEE number nobody trusts. Triage the event list regularly so the Pareto reflects reality.

Understand modes and states

- A **State** classifies what the equipment was doing (executing, faulted, held, starved...). Its flags decide how every report counts it: **Running** counts toward run time (the Performance and Availability numerator), **Available** toward available time, **Faulted** marks a *failure* (what MTBF, MTTR and the fault reports count), **Blocked** and **Starved** mark waiting on the rest of the line rather than a fault of this machine, and **Track Events** opts the state into downtime **events** — the ones reviewed on OEE Events and the ones that start a *Machine Down Too Long* notification.
- A **Mode** classifies the operating context; its **Scheduled** flag is what splits downtime into **planned** vs. **unplanned** — the single biggest lever on the availability number. Read it this way round: **Scheduled** means the machine was supposed to be producing, so a stop *inside* a scheduled mode is **unplanned** downtime, and a stop in an unscheduled mode (maintenance, no orders) is the planned kind. That is also what a *Machine Down Too Long* notification goes by: it ignores planned downtime, so it is raised for a stop in a scheduled mode and not for one in an unscheduled mode.

Because these definitions drive every downtime and availability figure, they are maintained centrally, by the Engineering role or an administrator.

Configure OEE (Engineering or Administrators)

Configuration → **Additional Configuration** → **Equipment & Processes** → **OEE Configuration** opens a page of tiles. Anyone can open it, and **OEE Event Review** behind it; the three configuration pages open only for the **Engineering** or **Administrator** role:

Tile	What you maintain	Why it matters
Manage OEE Modes	Operating modes, their colour, and the Scheduled flag	Splits downtime into planned and unplanned.
Manage OEE States	Equipment states, their colour, and the flags described above	Defines uptime, failures, and what raises events; distinct colours keep states apart on the charts.
OEE UDF Definitions	The labels of the five user-defined fields (UDF00 to UDF04); Edit a label, or remove one that is no longer used with Clear or Delete	Lets you tag and filter OEE data in reports.
OEE Event Review	—	The same page as Production → OEE Events .

A new reporting database comes with the standard (PackML) states, each in its own colour, and seven modes (Undefined, Production, Maintenance, Development, Dry Run, Pass-Through and Idle). An upgrade adds any that are missing and never changes one that exists, so your own colours and flags survive it.

Note: The **event start delay** and **event recovery** times — which decide how raw activity is grouped into events — are set on each equipment's own page ([Configure Equipment](#)), not here. They are saved to the configuration database, so a change reaches OEE event detection only after the scheduled synchronization jobs have carried it to the reporting database ([System Configuration](#)), not the moment you save. Tuning them changes how short blips are merged or split.

Field Reference

OEE report controls (typical):

Field	Required?	What it does / valid values
Equipment selection	Yes	One or more equipment from the plant model.
Start / End time	Yes	The reporting time range.
UDF selections	No	Filter by user-defined fields.

OEE event (classification: Production; re-coding a reason someone else gave: Production Supervisor or Administrator):

Field	Required?	What it does / valid values
Reason	Yes	One of the configured downtime reasons. Re-attributes the downtime in every report; the reason carries planned/unplanned and its loss category.
Notes	Conditional	Free text. Required when the chosen reason asks for it (as shipped, <i>Other</i>).

Alarm definition (per machine, on its **Alarm Definitions** pages, or imported from a spreadsheet; Engineering or Administrator):

Field	Required?	What it does / valid values
Code	Yes	The integer the machine reports.
Description	No	What the alarm says. Up to 200 characters.
Severity	Yes (column)	0 Information, 1 Warning, 2 Error; blank means 0. How bad it is.
Category	No	The number of one of the configured alarm categories (as shipped, 0 to 5); blank means 0, Unclassified. What KIND of alarm it is, and who is called.
Product Impact	No	0 not on the product record, 1 on the record, 2 on the record and holds product; blank means 0. Recorded with each alarm when it is raised; decides which alarms appear on the Product Report and the Product-Impacting Alarms report, and is used as a filter. Nothing is held automatically.
Cause	No	Why the machine raises it. Up to 500 characters.

Field	Required?	What it does / valid values
Consequence	No	What happens if it is left. Up to 500 characters.
Operator Action	No	What to do about it. Up to 500 characters.
Response Seconds	No	How long the operator has. A whole number greater than zero, or blank. Drives the Overdue mark on the alarm board.
Responsible Group	No	The maintenance group that owns the fix, when it is not the category's group. Up to 50 characters.
Reason for the change	Conditional	Up to 500 characters. Required when the change moves a code's Product Impact to or from 1 or 2 — by editing, retiring, reinstating or importing — and nothing is saved without it. Kept in the definition's Change History.

Related Topics

- [Plant & Equipment Model](#) — event timing & alarm definitions
- [Production Execution](#)
- [Reports](#)
- [Dashboards](#) — OEE reports and the alarm board on a wall screen
- [Notifications](#) — alarm, downtime and digest publications
- [System Configuration](#) — system parameters 12 and 17, synchronization jobs

Chapter 12: Reports

Overview

IntelliTrack's **Reports** menu turns the data captured during production into answers: where a product has been, what a suspect lot went into, how a process is performing, what quality looks like, what inventory you hold, which carriers need service, and how suppliers are doing. Most reports share one pattern — pick what to report on, run it, then export it or revisit it from history. This chapter describes each area and the shared mechanics. (OEE reports, including the alarm reports, have their own chapter — see [OEE](#).)

Why this matters: reporting is where the data you captured pays off. Each report exists to support a specific decision — *is this lot safe to ship, which step is dragging yield, which supplier to put on probation, do we have enough inventory?* Knowing what each report answers (below) is the difference between "running a report" and getting a decision out of it.

Before You Begin

- Running and viewing reports is available to **any signed-in user** (the **General** role — see [Security and Sign-In](#)).
- Two maintenance actions — **Rebuild Inventory Aggregates** and **Rebuild WIP aggregate** — require the **Administrator** role, and only administrators see their buttons. Both recompute the same work-in-process totals (see [Inventory Reports](#)).

How to Get Here

All report areas are on the **Reports** menu:

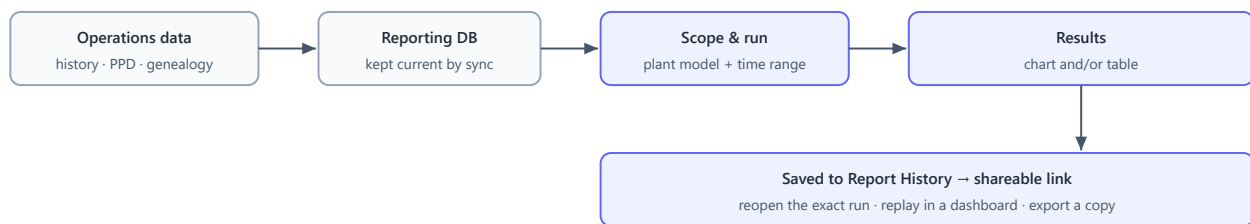
Menu item	Opens
Reports	The Reporting Index — a tile for each report area, and the only way in to three reports with no menu item of their own: Where-Used / Recall Impact , BoM Consumption / Variance and WIP Aging .
Product Report	The Product History Report — one unit's complete record.
Performance Reports	Product Performance Reports — process-data (PPD) exports, trends, SPC, Pareto, histogram, scatter plot and curves.
Quality	Quality Reports — yield, shift performance, deviation/override trend, quarantine aging, product-impacting alarms.
Audit Results	Product audit results (Product Audits).
OEE Reports	OEE and alarm reports (OEE).

Menu item	Opens
Inventory Reports	Inventory levels, the production chart, and work-in-process status.
Supplier Reports	Products, material usage and quality by supplier.
Carrier Reports	Cycle life, activity, yield and measurements for reusable carriers.
AI Insights, Dashboards	Their own chapters (AI Insights , Dashboards).

The **Reporting Index** tiles are **Product Performance Reports**, **OEE Reports**, **Quality Reports**, **Inventory Reports**, **Supplier Reports**, **Work In Process Status**, **Where-Used / Recall Impact**, **BoM Consumption / Variance** and **WIP Aging**.

The shared report mechanic

Running a report — and reusing the run



Reports read the reporting database, kept current by sync — so a momentary lag after production is normal, not a missing record.

The shared steps:

1. **Scope** — choose equipment/process from the plant model and/or a **time range**; some reports add their own filters. The range is either a **Custom Selection** (a **Start Date** and **End Date**) or a **Report Time Span** — **Last Hour**, **Last 4 Hours**, **Last 8 Hours**, **Last 12 Hours**, **Last 24 Hours**, **Last 7 Days** or **Last 30 Days** — counted back from the moment you run the report. Most report pages open with the selections you last ran them with, so a report you run every shift is one click.
2. **Run** — select **View Report**; a wait indicator appears while it generates.
3. **Review** the chart and/or table. Every chart has a toolbar of its own — zoom by dragging, **Reset** to go back, **Image** to save the picture, **Data** to see the figures behind it — and a report that draws several charts shows them as panels that follow each other. Bar charts always start their value axis at zero, so bar heights compare directly. Where a chart plots individual products (Trend, Scatter Plot, SPC), clicking a point opens that product's Product Report in a new tab. See [Reading a chart](#).
4. **Export** — tick **Show Data Table** on a chart report to list the charted values beneath it, with a download icon that saves them as a CSV file. Some reports have an export of their own (Export Report Values to Excel, curves to CSV).

5. Most runs are **saved to Report History** with a shareable link (`/Reports/ReportHistory/{id}`), shown under the results as **Share this report!** with a **Copy Link** button. The link reopens that run exactly as it was — the same figures, even after more production — which is what makes it worth pasting into an e-mail, a shift log or an [AI Insights](#) question; it is also the mechanism [Dashboards](#) use to replay reports. **Not saved** (run them again instead): Export Report Values, the Supplier Reports, the Carrier Reports, WIP Aging, and the Product Report's print view.

Dates and times on report pages, chart axes and export headings follow the site-wide date and time format an administrator chooses on **Additional Configuration** (see [Date and time format](#)); the windows your report pages remember keep working when that format changes. A few columns keep a fixed year-month-day form (for example 2026-03-14) whatever the setting: the dates in Shift Performance, Deviation / Override Trend and Quarantine Aging, the Where-Used **Timestamp**, and the WIP status page's calculation times.

Why a saved link can stop opening. Saved runs are kept while people use them, not forever. Opening a link restarts its clock, and the Reporting database's maintenance purge removes a run that nobody has opened within the **Report History Retention Time** (System Parameter 12; 30 days if it is not set) — unless an AI Insights conversation still refers to it ([AI Insights](#)). A run saved by an earlier version of the software can also outlive the report it replays: when a report has since been renamed or reshaped, its old runs cannot be reopened, and they are not converted. Either way the link shows **Report Link No Longer Available** — run the report again for a new link. Opening a dead link changes nothing: it neither deletes the saved run nor keeps it alive, so the purge still retires it on schedule.

A link saved before this version of the software was installed opens its tables but not its chart: a link draws the picture stored with the run, and this version stores a different kind of picture. Run the report again for a link that carries one.

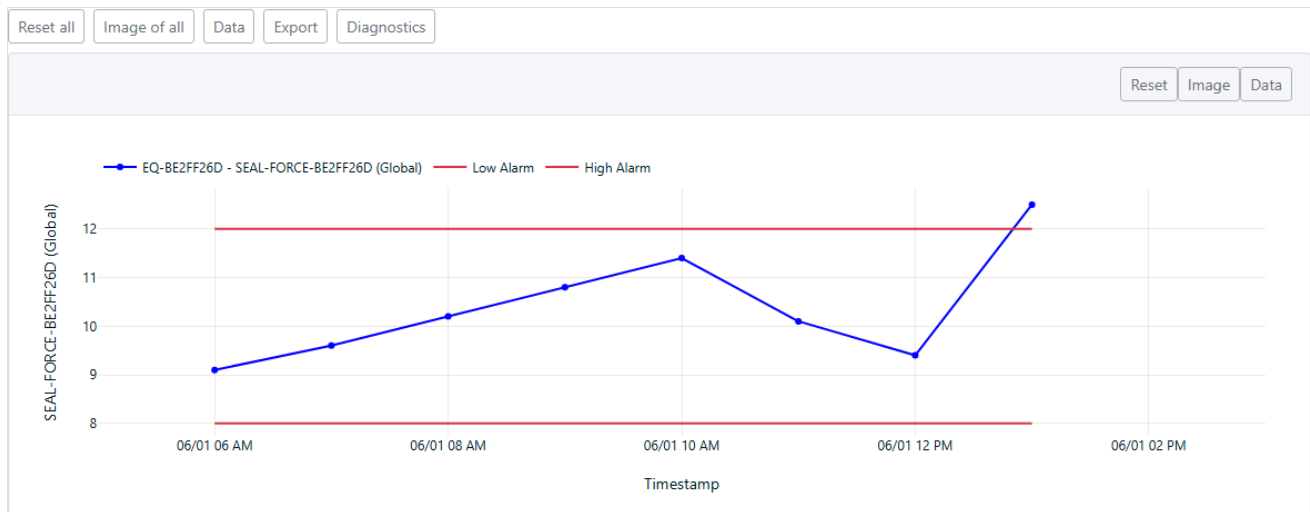
Reading a chart

Every chart in IntelliTrack is drawn the same way and answers to the same controls, whether it is one plot on a report page, a panel of several, a tile playing on a wall screen or the equipment drawer on the WIP page.

Looking closer. Drag a rectangle inside a chart to zoom into it, or drag along an axis to slide the view along that axis. Hovering a point names it — the serial number, the value, the time — and on a chart with several series, all of their values at that position. The chart's toolbar carries:

Control	What it does
Reset	Puts this chart back to the whole range.
Image	Saves this chart as a picture (PNG) to download.
Data	Shows the report's data table beneath the chart — the same table as Show Data Table .

Control	What it does
3D / Flat	On a chart that has both, draws it as a live 3D view or as its flat equivalent. See Curve reports .
Linked / Unlinked	Stops this panel following the others, or makes it follow again.



Two toolbars: the report's, top left, and the panel's, top right. The limits are layers like any other, so the legend can hide them.

The **legend** above the plot names every series — the measurement, and each limit drawn beside it. Selecting one hides or shows it, in every panel that draws it.

Above all the panels sits the report's own toolbar: **Reset all**, **Image of all** (the whole report as one picture), **Data** (the table for the whole report), **Export** (its figures as a CSV file) and **Diagnostics** (what the renderer is doing on this machine, for when a chart does not look right).

The exported file holds the rows and columns of the **Data** table, and is written to open straight into a spreadsheet without anything moving or running. A value that holds a comma, a quotation mark or a line break — a fault description, a material name — is written in quotes, so it stays in its own column. A name that begins with **=**, **+**, **-** or **@** — an equipment, a fault, a serial number — is written with an apostrophe in front of it, so the spreadsheet shows the name rather than working it out as a formula; a file passed around a plant cannot carry a formula someone typed into a name. Counts, measurements and percentages are written as plain numbers, negative ones included, so they can be summed and charted. The **Export Raw Data** icon on the **OEE State Chart** writes its file the same way.

A report with several charts shows them as panels, one above another, and its toolbar gains **Split** and **Merge** (separate the series onto a panel each, or bring them back onto one where their units allow it) and — where a report offers more than one way to see the same figures — one button per view, with the current one pressed. Panels that measure the same thing are **linked**: zoom one and the rest follow, hover a point and the matching point is marked on the others. That is what makes a comparison readable, and **Linked** on any panel opts it out.

Why a dense chart shows fewer points than the report holds. A curve can arrive with hundreds of thousands of samples while the panel drawing it is a few hundred pixels tall. Above a configured count (System Parameter 142, 5000 as shipped) the *plot* is thinned, and the panel says so: *"Showing 5,000 of 190,000 points, keeping the extremes and the limit crossings."* The reduction keeps the first and last sample, the highest and the lowest, every crossing of a limit, and the start of every gap, then spends what is left on the shape of the rest — so the peak you are looking for is on the plot, and a gap in the data stays a gap instead of the line closing over it. **The data table, the CSV export and every stored curve metric are never reduced:** they are computed from, and carry, every sample. Zoom in and the plot redraws from more of it.

Dates on a time axis follow the site's date and time format like everything else ([Chapter 17](#)). An axis has less room than a table, so it shows the part of the format that changes across the window — the day and month while you look at a week, the time of day once you have zoomed into an hour — and it pads single digits (a site on the shipped US format reads `09/20` on an axis and `9/20` in the table beside it).

If a chart does not appear, the page will say why rather than leave a blank space: a report with nothing to draw says so, a comparison of more panels than the site allows (System Parameter 140) says that instead of drawing part of it, and a 3D view on a machine whose graphics cannot render one opens flat with **Drawn flat. Use the 3D button to draw it as a live 3D view.** A chart that is missing altogether, with no message, is a sign the web files did not all reach the server — tell whoever installed it.

Product Report

Answers: the full story of one unit. Open **Reports** → **Product Report** (the page is titled **Product History Report**), type the unit's identifier under **Search by Serial Number**, and select **View Report** to see its details, genealogy, processing history, process data and audits — the report you pull for a customer return, a recall or an auditor's question. (The home-page search opens the product's **Details** page instead; its **Full Report** link leads here — see [Products](#).)

What you can type. The current **serial number**, the **alias** or the **shipping code** all find the unit. If none of them matches, IntelliTrack also tries **former serial numbers** — the name a part carried before a process step renamed it — so an old label still leads to the part.

When more than one product answers. A serial number, alias or shipping code is unique within one site, but several sites can report into one Reporting database, and one product's alias can be another's serial number. So when the identifier matches more than one product, IntelliTrack does not guess: it asks **Which product do you mean?** and shows each product as a card with what tells them apart — the **Site** it came from (on an installation that records one), its disposition, which identifier matched when it was not the serial number, its material, quantity, where it was last seen, when it was created and last updated, and how many curves it has. **Open Report** on the right one opens its report; **Include Extended Data** comes across from the search and can still be changed there. Choosing carefully matters: a report on the wrong plant's part answers a customer return or a recall with somebody else's history. Nothing is remembered

— the next search for the same identifier asks again, because next time you may mean the other product. The curve report asks with the same page (see [Curve reports](#)).

Include Extended Data adds to the Process Data section the measurements of every product in the unit's genealogy — the components it consumed and the assemblies it went into — so one report shows the data of the whole build. When the section then holds more than one product's measurements, a **Serial Number** column at its left names the product each row was measured on; the unit's own report does not have the column. Point at the ⓘ beside the option for a reminder of what it adds.

Curves (top right) opens the unit's curves in the curve report, in a new tab: the press force against distance, the torque against angle — whatever its machines recorded as a curve rather than as a single number. The number on the button is how many curves were recorded; with Extended Data it counts the curves of the whole build, and the curve report lists exactly those. Up to four curves of one kind open already drawn; more than that, or a mix of 2D curves, 3D curves and surfaces, opens as a list to choose from (see [Curve reports](#)). When the unit has no curves the button is greyed out and its tooltip says **No curves were recorded for this product**. If the curves could not be read, it says that instead — so a unit whose curves failed to load is never mistaken for one that has none.

Printing. The printer icon opens a print-ready copy in a new tab (every history row laid open, no links) and raises the browser's print dialog, for a paper or PDF record; the print copy is not saved to Report History. A unit with curves first asks which of them to include — **All curves**, **No curves**, or **Choose curves** from the list — because paper cannot follow a link: each curve you include prints as a numbered figure (**Figure C1**, **Figure C2** ...) in a **Curves** section at the end of the copy, drawn exactly as the curve report draws it, with its attribute, equipment module, time, operator and sample count under the title, and its nominal reference dashed beside it where it has one. A 3D curve or a surface prints as its flat views, which are the ones you measure from. A unit with no curves prints straight away, as before.

Print product report

×

This product has 1 curve(s). Choose which to include in the printout.

☐ All curves
 ☐ No curves
 ☒ Choose curves

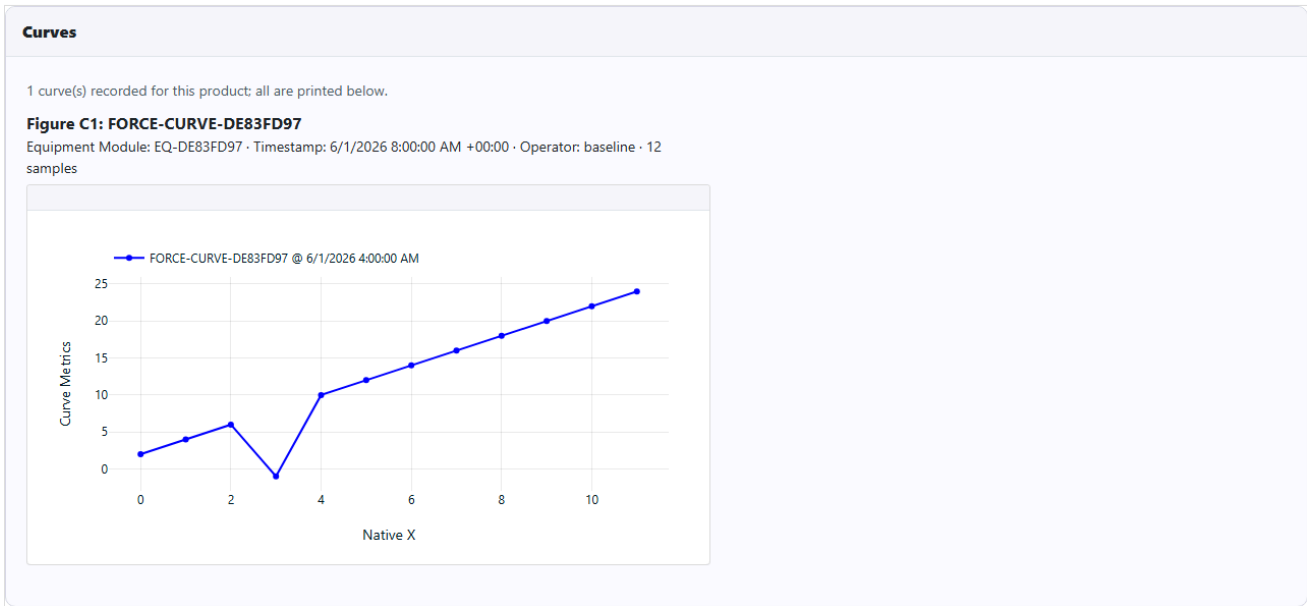
☒ **C1** FORCE-CURVE-CBCCC111 EQ-CBCCC111 · 6/1/2026 8:00:00 AM +00:00

Each printed curve appears as a figure at the end of the report, under the number shown here.

Print

Cancel

The print options for a unit with one curve. Each curve is listed under the figure number it will print as, with its equipment module and the time it was measured.



The end of the printed copy. The first line says how many curves were recorded and which were printed; each figure's caption names the attribute, equipment module, time, operator and sample count.

- The numbers do not move.** Curves are numbered in the order they were measured, so **C3** is the same curve on every copy, whichever curves you print. The Process Data table's curve rows name their figure, or say the curve was not printed.

- **The copy says what it left out.** The Curves section always states how many curves were recorded and which were printed, so a curve left out on purpose never reads as one that is missing.
- **A very long curve is drawn from a sample of its points.** A curve with more samples than a chart can show (System Parameter 142, 5,000 as shipped) is drawn from that many, keeping the first, last, highest and lowest of each stretch so no peak is lost; its caption says so. The recorded samples are untouched — **Export Raw Curve Data** on the curve report still gives every one.

Why the print waits. The figures draw after the page has loaded, so the print dialog waits until every figure has finished: a half-drawn chart on paper is a record that is wrong without looking wrong. If a curve cannot be drawn, or the figures take longer than 30 seconds, the copy is **not** sent to the printer and a notice at the top says why. Reload to try again, or use the browser's own Print command to print the copy as it stands.

The report reads, top to bottom:

- **Details** — the unit's current state (material and its image, status, amount, location, carrier and so on), read-only.
- **Genealogy** — every produced/consumed link in the unit's family, walked both ways (the components inside it and the assemblies built from it), generation by generation: **Generation**, **Produced Serial Number**, **Consumed Serial Number**, **Transaction Type**, **Equipment Module**, **User** and **Date Created**. Each serial opens that product's own report in a new tab. The walk stops at the plant's limits — **Max Generations** (System Parameter 42, 99 by default) and **Genealogy Rows** (System Parameter 41, 1,000 rows by default) — so a very large family is shown only up to that many rows.
- **Product History** and **Process Data** — below.
- **Machine Alarms During Processing** — the machine alarms that were active while the unit was being worked; see [Machine alarms during processing](#).
- **Quality Audits** — only when the unit was pulled for a QA audit ([Product Audits](#)): who pulled it and when, who completed it and when, the result (**Pending QA** until QA finishes), notes, and each measurement — including audits recorded at other sites once they have synchronized. Signatures are not shown here: a signed audit's **Electronic signatures** are on its **Audit Record** page (**Reports** → **Audit Results**, then **Details**).

Product History is the process-by-process account: one row per processing event, showing when it started and ended, the equipment module that ran it, the material it went in and came out as (with revision), and whether it passed (**Pass**, **In-Process** or **Reject**). Use the arrow at the left of a row to open that event's specifics — **Processed As Serial No.** (the serial it was worked under, shown as *old* --> *new* on an event that renamed the part), **Checked Out By** and **Checked In By**, the **Control Instance Recipe** it ran under, and the **Failed Attributes** if it rejected — including *Incomplete BOM* when the check-in was rejected because the station's bill-of-material items were not satisfied ([Bills of Material](#)). Rows that rejected are tinted red, and two kinds of row also carry a badge:

- **Failed Check-Out** (red row) — the equipment presented the part but the check-out was refused; the reasons are in the row's details. This is how a supervisor sees why a part kept being turned away at a station ([Production Execution](#)).
- **User Edit** (blue row) — a change made on the web rather than by equipment. Its details list the **Changes**, field by field, as *field: old → new*, with the values as they read at the time.

Why the control instance is per row. Each process a part visits runs under its own control instance, so the recipe that governed OP200 is generally not the one that governed OP400. Reading it off the row tells you which recipe — and which revision of it — was in force for *that* step; the link opens the control instance report as of the values in force at the time, not its latest revision. A row shows a dash when the event recorded no process data, so there is nothing tying it to a recipe run — or when IntelliTrack cannot tell which run it was: measurements from a whole-tray cycle are matched to the event by module and time, and if that window holds data from more than one control instance the report leaves the field blank rather than guess.

Reading the Process Data table

Process Data lists every measurement recorded against the unit, newest first: **Equipment Module**, **Operator** (who submitted that measurement set), **Attribute**, **Low Alarm**, **Target**, **Actual**, **High Alarm**, **Timestamp** and **Control Instance Recipe**.

The colour of the Actual cell is the verdict IntelliTrack stored when the value was recorded — the same verdict that decided the part — not a judgement the page makes again:

Colour	Legend	Meaning
Green	Within Limits	Inside the recipe limits.
Red	Out of Limit - Evaluation	Outside the limits on an attribute used for evaluation — this is what rejects a part.
Orange	Out of Limit - Non-Evaluation	Outside the limits on a data-collection attribute — informational; it never rejects the part.
None	—	Nothing to compare: the recipe sets no limits for this attribute (the limit cells are blank too), or the attribute is not part of the control instance (marked \$). Values recorded before IntelliTrack stored a verdict for each value also show no colour.

For a multi-value attribute, **Target** lists the allowed values. Markers beside a value say where it came from:

- **Audit** badge on a light-blue row — recorded by the line when the unit was pulled as an audit sample.
- ***** (**Externally Loaded**) — loaded from an external process rather than an IntelliTrack event; the equipment reads **Imported**.

- † (**Skipped**) — not measured at this check-in; a prior recorded value was carried forward and evaluated.
- ‡ (**Not recorded**) — a required attribute got no value at this check-in.
- § (**Not in control instance**) — the attribute is not part of the control instance the part ran under (it was mapped to the module after the instance was built), so the value was recorded but not evaluated and has no colour. It is judged from the next control instance on.
- **Tray** badge (**Tray:** and the carrier's name, when known) — recorded as part of a whole-tray group cycle, so the value is shared with every part on the tray (see [Carrier Reports](#)).

A curve attribute shows **See Curve**, which opens that curve, already plotted, in the curve report. On the printed copy it names the curve's figure instead (**Figure C2**), or says **not printed**. A curve attribute whose samples never arrived shows **No samples recorded**: there is no curve to open. **Control Instance Recipe** is printed only where it changes down the table — at the first value of each recipe run — and links to the control instance as of the revision those values were judged against, so you can see exactly where the process moved from one recipe run to the next.

Machine alarms during processing

Answers: was anything wrong with a machine while this unit was on it? A machine that alarms while it works a part can leave that part out of specification — a fill weight out of range, a seal temperature low, a guard opened mid-cycle — and the unit's own measurements do not always show it. This section puts those alarms on the unit's record, so a batch-record review, a deviation investigation or a customer complaint starts from one page, instead of from the unit's check-out times laid against the machine's alarm journal by hand.

What decides whether an alarm is on the record. Every fault code carries a **Product Impact**, set on the equipment's fault-code list ([OEE](#)): *Not on the product record*, *On the product record*, or *On the record, and holds product*. Each alarm keeps the Product Impact its code had **when the alarm was raised**. Reclassifying a code afterwards changes the next alarm it raises, never a record already made — so the record you review today says what it said last month. An alarm raised with a code that had no classification counts as not on the record.

How an alarm is matched to the unit. An alarm is listed against a processing step when it was active at any moment the unit was being worked there: on the same equipment module, between the step's check-out and its check-in. An alarm that cleared the instant the step began, or was raised the instant it ended, is not listed. One alarm that stood over several steps — or over every position of a carrier — is listed against each of them, and counted once in the summary. Both times are the moments IntelliTrack recorded them, not the machine's own clock, so the match is as fine as the rate at which the machine reports its alarms.

The section shows:

- **A summary** — how many alarms that belong on the product record were active while the unit was being worked, and how many of them hold product.

- **One row per step and alarm** — **Equipment Module; Step**, its check-out and check-in; **Alarm**, the fault code and what it said (**First Out** marks the alarm that started a cascade); **Severity; Category; Product Impact; Alarm Active**, when it was raised and cleared (or *Still active*); and **Time During Step**, how long it was active while the unit was being worked in that step. A row whose alarm holds product is tinted red. (*current wording*) after a description means the alarm recorded none when it was raised, so this is the fault code's description today.
- **The other alarms** — the alarms active at the same time that do not belong on the record. They are kept for investigation behind **Show the ... other alarm(s)**, up to 250 of them (the Alarm Journal and Fault Detail reports list every alarm for a machine and time span). The print copy leaves them out and says how many there were: a printed record carries what the record carries.

Steps that were never checked in. A unit can be checked out at a machine and never checked in there — it was added at that machine and moved straight on, or its cycle was abandoned. Such a step has no recorded end, so it is taken to end at the unit's **next check-out** on any machine, or at the **machine's next check-out** of another unit, whichever came first. A step with neither is still checked out, and alarms count up to now. The **Step** column says which applied, and a note under the section counts such steps. (On a machine that works several units independently, an unfinished step ends at the next unit's check-out.)

Include Extended Data widens this section the way it widens Process Data: the alarms on the steps of every product in the unit's genealogy are listed too, each row naming its serial number.

When alarms may be missing. Alarm records are deleted after the **Report History Retention Time** (System Parameter 12, 30 days as shipped) unless the site keeps an archive (**Alarm Archive Retention Time**, System Parameter 17, off as shipped), while this report keeps a product for the **Minimum Product Retention Time** (System Parameter 10, 180 days as shipped). When some of a unit's steps are older than the alarm records still kept, the section says so, rather than showing an empty list that would read as a clean record. **If your quality system relies on this section, set System Parameter 17 to at least System Parameter 10** ([System Configuration](#)).

When the alarms could not be read, the section says so in red. Do not read it as a record without alarms.

What this changes in the system: nothing. The section reads the alarm log and the processing history; it writes nothing and stores no link between them, so the record is always worked out from the two records themselves. It needs the alarms to have been recorded: the equipment must report them to IntelliTrack (the `aoi.LogAlarms_v1` or `aoi.LogOEE_v1` interface — see [OEE](#)).

Product History

	Processing Start Time	Processing End Time	Equipment Module	Start Material	End Material	Pass/Fail Result
▶	9/19/2026 10:46:00 AM -04:00	9/19/2026 10:50:00 AM -04:00	DEMO_OP300	DEMO_Bottle (Revision 0)	DEMO_Bottle (Revision 0)	Pass
▶	9/19/2026 10:40:00 AM -04:00	9/19/2026 10:44:00 AM -04:00	DEMO_OP200	DEMO_Bottle (Revision 0)	DEMO_Bottle (Revision 0)	Pass
▶	9/19/2026 10:30:00 AM -04:00	9/19/2026 10:38:00 AM -04:00	DEMO_OP100	DEMO_Bottle (Revision 0)	DEMO_Bottle (Revision 0)	Pass

Machine Alarms During Processing

1 alarm(s) that belong on the product record were active while this part was being worked; 1 of them hold product.

Equipment Module	Step	Alarm	Severity	Category	Product Impact	Alarm Active	Time During Step
DEMO_OP100	9/19/2026 10:30:00 AM -04:00	103 Fill weight out of range	Error	Quality	On the record, and holds product	9/19/2026 10:05:00 AM -04:00	00:05:00
	9/19/2026 10:38:00 AM -04:00	First Out				9/19/2026 10:35:00 AM -04:00	

[Show the 1 other alarm\(s\) that were active but do not belong on the product record](#)

- Each alarm is shown as it was classified when it was raised. Changing a fault code's classification later does not change this record.

The Product Report for bottle DEMO_260919-1030. The filler, DEMO_OP100, was in a Hold alarm (103 Fill weight out of range) for the first five minutes of this part's step, so the row is red. The one alarm that does not belong on the record stays behind the link.

Where-Used / Recall Impact

Answers: a component or lot is suspect — which finished products contain it? Open **Reports** → **Reports** and the **Where-Used / Recall Impact** tile. Enter the **Source Serial Number or Lot** (an alias or shipping code works too), optionally change **Generation** — how many levels of assembly to follow upward, 1 to 50 (10 by default) — and select **View Report**.

Every downstream product whose genealogy contains the source is listed, level by level: **Generation**, **Downstream Serial Number** (opens its Product Report), **Material**, its current **Disposition**, the **Equipment Module** where the consumption was recorded, and the **Timestamp**. This is the scoping step of a recall: it shows how far a bad lot travelled and, through **Disposition**, which of those products are still in your hands and which have already shipped. To hold them, raise a quarantine incident ([Quarantine & Disposition](#)), whose genealogy search follows the same links.

If more than one product answers to what you typed — possible when several sites report into one Reporting database — this report follows the most recently updated of them; check the source with the Product Report first, which lets you choose. **The supplied serial number or lot could not be located.** means no product answers to the entry at all; **No downstream products were found...** means it exists but nothing was built from it.

Performance Reports (process data / PPD)

Answers: how is the process behaving over time? Reports → **Performance Reports** opens **Product Performance Reports**:

Report	Answers
Export Report Values	The raw measured values, product by product, for a spreadsheet or statistical tool.
Trend Report	How attributes moved over time — spot drift before it becomes scrap.
Statistical Process Control	Is the process stable, and capable of meeting its limits?
Reject Pareto Report	Which causes produce most of the rejects — where to aim improvement.
Attribute Histogram	How one attribute's values are distributed — centred, and how wide?
Attribute Scatter Plot	Does one measurement move with another?
PPD Curves by Serial Number	The X-Y curves a machine captured for a part.

Trend Report. Pick the equipment and one or more attributes from the plant model, the window, and optionally **Eliminate Zeros**. Each equipment/attribute pair is its own line, and each attribute gets its own value axis, so attributes with different units can share one chart.

Attribute Histogram. One equipment module and one attribute over the window. **Exclude Outliers** leaves out values more than 3 standard deviations from the median, and **Eliminate Zeros** leaves out zero readings. Use it to see whether a process is centred between its limits and how wide it runs, not just whether parts passed.

Attribute Scatter Plot. Pick an **X-Axis Attribute**, a **Y-Axis Attribute**, the equipment and the window. Each point is one product: its latest value of each attribute in the window, taken from the product itself or from a product in its genealogy — so a measurement taken on a component can be plotted against one taken on the assembly it went into. A long window over busy equipment can take too long; the report then says **This report covered too much production to finish...** — shorten the time span or pick fewer equipment modules.

Reject Pareto Report. Counts the reject reasons recorded at check-in on processing events that ended in the window, for the equipment, attributes and **Material (post processing)** you choose, and draws the top **Reject Items** (1 to 20) as bars — each labelled *equipment: process: attribute* — with the **Cumulative Percentage** as a line against the right-hand axis. Click a bar to open the OEE **Reject Detail** report for the same equipment and window in a new tab ([OEE](#)). Only rejects raised against an attribute are counted: an *Incomplete BOM* reject (the station's bill-of-material items unsatisfied at check-in) has no attribute, so it does not appear here — find it in the Product Report's **Failed Attributes**.

Export Report Values. Choose the equipment and attributes, **Material (post processing)**, one or more **Pass/Fail Result** values, and the **Reference Process** — the process whose measurements locate each

product in the window (set for you when the equipment you chose all performs one process). Optionally tick **Include Re-Run Values** (every value from repeated runs, comma-separated, instead of only the latest) and **Include Extended Data** (values from the product's genealogy relatives, each with its serial number). The result is one row per product, one column per equipment and attribute, with each value coloured by its stored verdict as on the Product Report (a value with nothing to compare is pale yellow here). **Items per Page** pages the table; the download icon exports **every** row, not just the page, to an Excel workbook (**Export Report.xlsx**) — for example for statistical analysis, or to attach to an AI Insights question.

Statistical Process Control (SPC)

Answers: is the process stable, and can it hold its limits? Pick one equipment module and one attribute and a **Count** of samples (2 to 10,000); the chart uses the most recent that many values, whatever their date, in the order they were taken.

It is an individuals control chart. Its centre is the samples' mean, and the **Upper Control Limit** and **Lower Control Limit** lines are the mean plus and minus 3 sigma, where sigma is estimated from the average change between consecutive samples — so the control limits describe what *this* process actually does. The recipe's **Low Alarm** and **High Alarm** are drawn as reference lines and feed the capability figures shown under the chart: **Cp (Process Potential)** and **CpK** use the within-subgroup sigma (what the process could do), **Pp (Overall Potential)** and **Ppk (Overall Capability)** the overall **Standard Deviation (Population)** (what it is doing), beside the **Average (Mean)** and **Sigma (Within)**.

Western Electric Rule Violations lists every sample that breaks one of four out-of-control rules: **Rule 1**, a point beyond 3 sigma; **Rule 2**, two of three consecutive points beyond 2 sigma on the same side; **Rule 3**, four of five beyond 1 sigma on the same side; **Rule 4**, eight in a row on one side of the centre line. A violation means the process has shifted even if every part is still within its limits — the time to act is before the scrap. To hear about a violation as it happens instead of finding it on the chart, subscribe to the **SPC Out-of-Control Signal Detected** notification ([What you can be notified about](#)): it applies the same four rules as values arrive, but against limits built from every sample since the last recipe change, so its figures can differ slightly from a chart of the last **Count** samples.

No chart is drawn — a message says why — when the samples were not all collected under the same control recipe revision (control limits mean nothing across a recipe change: lower the **Count**, or wait for more samples under the current recipe), when they were collected under no recipe at all, or when that recipe gives the attribute no alarm limits. The alarm (specification) limits come from the recipe revision in force when the data was collected ([Recipes](#)); the control limits never do.

Curve reports

PPD Curves by Serial Number shows the curves a machine captured — a press force against distance, for example. **Find Curves** for a serial number (its alias, shipping code or a former serial number work too) opens that unit's curves straight away, as the Product Report's **Curves** button does: up to four curves of one kind already drawn, otherwise the list to choose from. Select curves under **Available Curves**: one to

see it on its own, or up to four to compare them. **Add Serial Number** brings another part's curves into the list to compare against; the curves you had already selected stay selected, and nothing is drawn until **View Report**. With two or more selected, **X-Axis Mode** (**Native X**, **Sample Number** or **Normalized X (0-100%)**) and **Alignment Mode** (**Align at Start**, **Align at Peak** or **Align at Threshold**) decide how they line up. **Stored Curve Metrics** lists the metrics logged with the curve against their limits, and **Export Raw Curve Data** downloads the samples as a CSV file.

When more than one part answers. Several sites can report into one Reporting database, and two of them may have used the same serial number. Then **Find Curves** — and **Add Serial Number** — show the Product Report's **Which product do you mean?** page, so you choose the plant's part instead of being shown the curves of whichever part came first. A part with no curves is still listed, so you can see the serial matched it, but it cannot be opened (**No curves were recorded for this product.**), and a part already in your comparison says so. Choosing a part for **Add Serial Number** brings you back to your comparison exactly as you left it, with the part added; **Back To Curve Report** returns without one. A serial that matches no part at all says so under the box you typed it in.

Opened from a Product Report — its **Curves** button, or **See Curve** on a curve row — the curve report opens on that unit by its identity, not by its serial number, and keeps to it as you change the selection, so it never needs to ask which part you mean. A part you add, or choose on the picker, is kept the same way.

A curve can have a third component. What a curve attribute captures is fixed when the attribute is created, and the report follows it. A **2D curve** is the X-Y plot above. A **3D curve** is a path through space — a tool's position, or a value against two conditions. A **surface** is one value measured over a grid: a thickness map, a temperature field across a plate.

A 3D curve opens as **Three-dimensional path** with the three flat views beside it — **X against Y**, **X against Z**, **Y against Z**, each labelled with the components' own names and units. All four share one cursor: hover a sample on any of them and the same sample is marked on the rest, which is how you read a feature off a path without having to turn it to exactly the right angle. **With 3D** and **Flat only** switch between the pair and the flat views on their own.

A surface opens as the surface itself, with its **Colour map** and **Contours** beneath. A surface captured as scattered points rather than a grid is drawn as the points it is, and says so: nothing is interpolated between measurements that were never taken.

Comparing several 3D curves gives each one its own view, turned together — rotate one and the rest follow, so you are always comparing the same angle. Surfaces are not compared on one plot; the page says how many of your selections were surfaces and leaves them out, so open a surface on its own.

Why a 3D view sometimes opens flat. A live 3D view needs the browser's graphics hardware. On a thin client, a remote desktop or any machine whose browser renders 3D in software, a 3D panel opens flat and says so — **3D** still draws it, at whatever speed that machine manages. A page keeps a limited number of live views (System Parameter 143, 8 as shipped); ask for one more and the view you used least recently goes flat to make room, which it also says. And if a view loses its graphics context outright, it falls back to a flat plot whose colour carries the third component: the rotation is what you lose, never a component.

Export Raw Curve Data gains a **Z** column for a 3D curve or a surface. For a 2D curve the file is exactly what it always was.

Use as nominal reference

A **nominal reference** is the shape a good part makes — the curve a qualified part actually produced, kept on a recipe revision so that every later curve of that attribute can be judged against it.

Once a revision holds one, every curve measured under it is **drawn with the reference beside it**, dashed, so you compare a shape against a shape; the **Max Deviation From Nominal** metric is stored with the curve; and a comparison of such curves can be switched to **% of target**, which expresses each curve against its own reference at the same sample. A curve measured before the reference was captured keeps the reference that was in force when it was measured — which is usually none, and reads as none.

You capture one from the curve in front of you. Below a single curve, **Use as nominal reference** asks which **recipe revision** to keep it on, takes an optional **description**, and **Capture reference** stores that curve's samples there. Four things are worth knowing before you press it:

- **Only revisions that still accept edits are offered** — in a standard configuration, a Draft. A reference is part of a recipe's definition, so it is authored where the rest of the recipe is and an approved or active revision is closed to it exactly as it is to any other edit ([Chapter 5](#)).
- **A revision holds one reference per curve attribute.** The picker says what each revision already holds — how many samples, and which part they came from. Capturing again replaces it whole, rather than adding to it.
- **Each control instance freezes the reference** it was created with, so a curve is always judged against the reference that was in force when it was measured. Capturing a new one does not re-judge yesterday's parts.
- **It is recorded in the audit trail** under your name, with the curve, the part and the number of samples it was taken from. Only **Recipe** and **Administrators** are offered the control; everyone else reads the report without it.

Production Report

Answers: how much did each machine finish in a window, and how much of it was good? The Production Report counts the processing runs that *ended* between a start and an end time, one row per

equipment module and **end material**, split into **Good**, **In-Process** and **Reject** counts — the figures behind a shift hand-over or a daily output review. It has no menu entry at present; open it at [/Reports/ProductionReport](#) (**Production Report Query**), choose the equipment from the plant model, the **Material (post processing)** (or **All**) and the window, and select **View Report**. (The **Production Report** tile on Inventory Reports is a different report: a chart of amounts by plant-model level — see [Inventory Reports](#).)

Enter the start and end as local clock times. Each is read for its own date, so a window on a winter morning covers that morning even when you run the report in summer, after the clocks have changed. The heading repeats the window exactly as you entered it, so what it says is what the counts cover.

Each run is saved to Report History like any other report, and the **Share this report!** link beneath the results reopens **that run**: the same window, the same rows and the same counts, even after more parts have been made. That is what makes it worth pasting into a shift log, an e-mail or an [AI Insights](#) question — everyone who opens it sees the figures you saw. To see the current figures, run the report again.

Quality reports

Reports → **Quality** opens **Quality Reports**:

Report	Answers
First Pass Yield / RTY	What fraction passed first time — the core quality KPI. Per equipment module in the window: Attempts (products), First Pass , Reworked (failed the first attempt and were run again), Scrap (failed the first attempt and were not run again in the window) and FPY % , plus the Rolling Throughput Yield (RTY) across the selected modules — the chance that a product gets through all of them first time.
Shift Performance	How output and yield differ by shift: per date, shift and module, Total , Good Count , Reject Count and FPY % , with Subtotal , Date Total and Grand Total rows. Shifts are fixed eight-hour blocks by start time: Day 00:00–07:59, Swing 08:00–15:59, Night 16:00–23:59.
Deviation / Override Trend	Whether recipe deviations and overrides are trending up (process control slipping): for each day in the window, how many Deviations and Overrides were requested, across all equipment.
Quarantine Aging	How long held product has been sitting (disposition backlog): every open quarantine incident with when it was Opened , its Age (days) , its age bucket (0–7, 8–30, 31+ days), its number of Suspect Products and who raised it. It runs as soon as you open it.
Product-Impacting Alarms	Which units were worked while a machine alarm that belongs on their record was active — see below .

How products are counted. Yield reports follow the pass/fail convention (1 = Good, 0 = In Process, negative = Reject) and count each product once, not once per processing event, so a yield can never exceed 100%. **First Pass Yield** judges each product by its *first* attempt at a module in the window, so rework counts against yield — that is the point of the measure. **Shift Performance** takes each product's *last* result within a shift: a part rejected and reworked to good in the same shift counts once, as good, while a part rejected in one shift and fixed in the next shows as a reject in the first and good in the second.

Under **Related quality reports (from other categories)** the same page links to **Statistical Process Control** and the **Reject Pareto Report** (above), the OEE **Reject Detail Report** ([OEE](#)) and the **Supplier Quality Summary** (below).

Product-Impacting Alarms

Answers: which units carry a machine alarm on their record — and which were worked under one that asks for a hold? The Product Report answers the question for one unit you already have in hand. This report answers it for everything made in a time span: the list a quality engineer needs before releasing a batch, or when a fault code is found to matter and every unit worked while it stood has to be reviewed.

Choose the equipment modules and the time span as on any report, and the **Product Impact** to look for — *On the product record* and *On the record, and holds product* are chosen when the page opens; add *Not on the product record*, or choose -- **All** --, to see every alarm. Then select **View Report**.

A unit is listed when one of its **check-outs in the time span** overlaps an alarm of the chosen Product Impact on the same machine — matched exactly as its Product Report matches it ([above](#)), including how a step that was never checked in is ended, and with each alarm counted as it was classified when it was raised. One row per unit:

Column	Meaning
Serial Number	Opens the unit's Product Report in a new tab, where the alarms are listed one by one.
Material, Disposition	What the unit is, and its disposition now — so a unit already shipped or already on hold stands out.
Steps	How many of its processing steps were worked under such an alarm.
Alarms	How many alarms. One alarm that stood over several steps counts once.
Hold Alarms	Of those, how many were raised as <i>On the record, and holds product</i> . A unit with any is tinted red.
Highest Product Impact, Highest Severity	The worst of its alarms.

Column	Meaning
Time Worked Under Alarm	How long the unit was being worked while at least one such alarm was active. Two alarms standing at the same time count once.
First Alarm	The first moment the unit was worked under one.
Machines	Where.

A run with no such unit says so. The run is saved to **Report History** like the other quality reports, so its link can be shared.

What this changes in the system: nothing — it reads, and it does not hold anything. A Hold alarm asks for containment; quarantining the units it lists stays a decision made on the Quarantine page ([Quarantine and Disposition](#)). Like the Product Report section, it can only report the alarms still kept (System Parameters 12 and 17, above).

Product-Impacting Alarms

Serial Number	Material	Disposition	Steps	Alarms	Hold Alarms	Highest Product Impact	Highest Severity	Time Worked Under Alarm	First Alarm	Machines
DEMO_260918-0630	DEMO_Bottle (Revision 0)	Released	1	1	0	On the product record	Error	00:04:00	9/18/2026 6:46:00 AM -04:00	DEMO_OP300
DEMO_260918-1800	DEMO_Bottle (Revision 0)	Released	1	1	0	On the product record	Error	00:02:57	9/18/2026 6:16:00 PM -04:00	DEMO_OP300
DEMO_260918-2230	DEMO_Bottle (Revision 0)	Released	1	1	0	On the product record	Error	00:00:03	9/18/2026 10:37:57 PM -04:00	DEMO_OP100
DEMO_260919-0300	DEMO_Bottle (Revision 0)	Released	1	1	0	On the product record	Error	00:02:03	9/19/2026 3:11:57 AM -04:00	DEMO_OP200
DEMO_260919-1000	DEMO_Bottle (Revision 0)	Released	1	2	1	On the record, and holds product	Error	00:05:03	9/19/2026 10:02:57 AM -04:00	DEMO_OP100
DEMO_260919-1030	DEMO_Bottle (Revision 0)	Released	1	1	1	On the record, and holds product	Error	00:05:00	9/19/2026 10:30:00 AM -04:00	DEMO_OP100
DEMO_260919-1430	DEMO_Bottle (Revision 0)	Released	1	1	0	On the product record	Error	00:01:57	9/19/2026 2:40:00 PM -04:00	DEMO_OP200
DEMO_260919-2130	DEMO_Bottle (Revision 0)	Released	1	1	0	On the product record	Error	00:04:57	9/19/2026 9:30:00 PM -04:00	DEMO_OP100
DEMO_260920-0400	DEMO_Bottle (Revision 0)	Released	1	1	0	On the product record	Error	00:01:03	9/20/2026 4:18:57 AM -04:00	DEMO_OP300
DEMO_260920-1530	DEMO_Bottle (Revision 0)	Released	1	1	0	On the product record	Error	00:04:00	9/20/2026 3:46:00 PM -04:00	DEMO_OP300

A part is listed when one of its check-outs in the time span overlaps an alarm of the chosen Product Impact on the same machine. Each alarm counts as it was classified when it was raised. The Product Report lists the alarms one by one.

Product-Impacting Alarms over three days. The two bottles the filler worked during the Hold alarm are red; the others were worked under alarms raised as Record.

Inventory Reports

Answers: what do we hold, and where? Reports → Inventory Reports:

Tile	Answers
Inventory Levels by Process Model	How much work is in each process, by material — pick processes and Materials (or All). Tick Eliminate Zeros to leave out the process/material pairs with no amount in any status, which otherwise appear as empty columns.
Inventory Levels by Location	How much is at each location, by material. A product with no location of its own counts against the equipment module holding it, shown as "@ module". Filter by plant model, Location and Material (or All , which also counts materials deleted since).

Tile	Answers
Product History Report	The Product Report .
Production Report	What finished in a window: processing events that ended in it, totalled at the Group Level you choose (enterprise down to equipment module) and by material. (For counts per equipment module, see the Production Report above.)
Inventory by Product Serial Number	Everything in one serial number's genealogy — what it was built from and what it went into — counted by material and current pass/fail status.
Inventory by Lot Number	The same for a lot: what it went into, and what state those products are in now.
Work In Process Status	The live WIP page .

The charts are stacked **Good / In Process / Rejected** bars; hover a bar for the count and amount (and unit) behind it. The two **Inventory Levels** reports count products that are currently inside a process step — checked out and not yet checked back in — from the work-in-process totals the WIP sync job keeps ([System Configuration](#)); a product sitting between steps is not in them.

Administrators see **Rebuild Inventory Aggregates** at the top of the page. It recomputes those work-in-process totals from scratch and has this site pull the per-equipment counts from them — the same counts the WIP limit gate reads at check-out ([Production Execution](#)), so a module wrongly blocked by, or wrongly clear of, its WIP limit is corrected at once. Normally the totals refresh automatically as data syncs, so a manual rebuild is for after a correction or a suspected drift, not routine use. It does the same work as **Rebuild WIP aggregate** on the WIP page.

Supplier Reports

Answers: how is each supplier performing? Reports → **Supplier Reports** has a card per report: pick the supplier on the card and select **Get Started** ».

- **Products by Supplier** — what you received from a supplier: each product's **Serial Number**, **Supplier Lot Number**, **Material**, **Amount/UoM**, **Product Status**, **Current Location** and **Date Created**.
- **Supplier Material Usage** — per supplier lot, the **Quantity Received**, **Quantity Consumed**, **Quantity Remaining** and **Quantity Quarantined** — how much of each delivery you have used and how much is on hold.
- **Supplier Quality Summary** — per material over the **Report Time Span** you pick (**Last 30 Days**, **Last 3 Months**, **Last 6 Months**, **Last 12 Months** or **All**): **Products Received**, **Products Quarantined**, **Products Rejected**, **Products Failed**, the **Quarantine Incident Count** and the

Failure/Quarantine Rate — the evidence behind a probation or de-listing decision ([Materials & Suppliers](#)).

Each list is paged and has a search box; these reports are not saved to Report History.

Carrier Reports

Answers: are my reusable carriers healthy, and is any pallet making bad parts? Reports → Carrier Reports (see Chapter 8 for how trays are operated):

- **Carrier Activity** — how many times each carrier was seen at each equipment module. **This is the report to use when deciding what to service.** Pick the carriers by plant-model scope (or one specific carrier), pick the window, and run it. Rows group by carrier and collapse; a filter box narrows by carrier or equipment name. See *Reading Carrier Activity* below.
- **Cycle Life** — for each carrier, the group cycles run since its last maintenance, its lifetime cycle count, and the last maintenance date. **Only whole-tray check-outs produce cycles** (Chapter 8), so if your plant checks parts out individually every figure here reads zero — that is correct, not a fault, and Carrier Activity is the report you want instead.
- **Yield by Pallet** — runs and reject rate for each pallet. Read it as a *funnel plot*: judge the reject rate against the run count, so a pallet with a handful of runs is not flagged on noise while a genuinely bad pallet with many runs stands out.
- **Attribute Analysis by Pallet** — the mean and spread of each measurement, per pallet, grouped by attribute. A defective pallet (a bent nest, a worn locating pin) shows up as a **shifted mean or a wider spread** than its peers on the specific measurement it distorts. A note of caution: pallet cycles also correlate with time-of-day, operator, and upstream lot, so treat a standout as a lead to investigate, not a verdict. It counts measurements taken when parts on a carrier are checked in one at a time; values recorded at a whole-tray group check-in belong to the tray cycle and are not included.

Where a measurement was captured as part of a whole-tray cycle, the Product Report's process data tags that row with a **Tray** badge — naming the tray when it is known — so you can tell tray-shared data from an individual measurement at a glance (see [Reading the Process Data table](#)).

Reading Carrier Activity

Answers: which carriers have done the most work, and where? Use it to decide which trays are due a wash or an inspection. It reports, it does not enforce — nothing is blocked on these numbers.

Choosing what to include. Set as much of the plant model as you want to narrow by and leave the rest on -- **All** --; a carrier scoped to the whole enterprise appears under any site you pick, because that is what its scope means. Optionally pick a **Single Carrier** to look at one tray on its own.

Choosing the window. Three kinds:

- **Since Last Maintenance** — each carrier is counted from **its own** last service event, so carriers washed on different days are still compared fairly. A carrier with no maintenance on record shows its whole retained history.
- A **relative span** — Last 24 Hours, Last 7 Days, Last 30 Days.
- **Custom Selection** — an explicit start and end.

Reading the numbers. Each carrier is one collapsed group; expand it to see the equipment modules it visited.

Column	What it means
Events	Parts handled — not trips. A full 8-position carrier contributes 8 events per pass through the station.
Est. Passes	Events ÷ the carrier's declared Positions , as a rough trip count. Blank when the carrier declares no capacity.
First Seen / Last Seen	The first and last event in the window.

Treat Est. Passes as an estimate, not a count. It assumes carriers run full, so a tray habitually run half-loaded shows fewer passes than it actually made. Where the distinction matters, judge on **Events** — it is a direct count — and use Est. Passes only for a sense of scale.

A carrier is worn by the station that wears it. A tray visiting six modules accumulates six times the events plant-wide, but only the furnace trips age it for furnace purposes. Expand the group and read the row for the module that does the damage, rather than the carrier's total.

When you service a carrier, record it with **Record Maintenance** on the carrier's **Details** page (Chapter 8) — that is what the *Since Last Maintenance* window resets against.

Material and WIP reports

(Curve reports are under [Performance Reports](#).)

BoM Consumption / Variance (Reports → Reports, tile of the same name) — **answers: where are we over- or under-consuming components?** For the finished products created in the window (optionally only the materials you pick), each component's **Actual Qty** consumed, taken from the genealogy, is set against its **Planned Qty** — the midpoint of the bill of material's minimum and maximum — with the **Variance** and **Variance %**. A component consumed but not on the bill, and one on the bill but never consumed, both appear, because both are variances ([Bills of Material](#)).

Work In Process Status (the **Reports → Reports** tile, or Inventory Reports) — **answers: where is work piling up right now?** One row per equipment module — **Status** (**OK** or **Over**), **Workcell**, **Equipment**, **Total Count**, **In Process**, **Reject Count**, **Amount** and **Reject Amount** — measured against the WIP limits set on the equipment ([Plant & Equipment Model](#)), with a **Workcell rollup** beneath. Type in the filter box or tick **Violations only** to narrow it. Select a row for its detail: each count and amount against its limit,

and a **24-hour processing trend** of parts **Started** and **Rejected** per hour. A module marked **Over** is refusing check-outs until its counts drop below the limits — the WIP gate ([Production Execution](#)). The figures are those this site last pulled from the Reporting database, and **Last calculation** shows when a module's figures last changed; **Administrators** can **Rebuild WIP aggregate** to recompute them from scratch after an out-of-band change (the same rebuild as on Inventory Reports). Each visit is saved to Report History, so the page can be shared or played on a dashboard; a saved run shows the figures as they were when it ran.

WIP Aging (Reports → Reports, tile of the same name) — **answers: what has stalled?** Open work in process — products whose latest step is still in progress — grouped by **Process**, **Equipment Module** and **Age Bucket** (0–1d, 1–3d, 3–7d, 7d+, measured from when the product entered its current step), with the **Count** and the oldest **Age**. It finds what the live status grid cannot: a part left at a station for a week. It runs as soon as you open it.

Field Reference

Typical report inputs:

Field	Required?	What it does / valid values
Serial Number (or lot)	Yes (Product Report, Where-Used, inventory by serial or lot, curves)	The product to report on — serial number, alias or shipping code; the Product Report also finds former serial numbers.
Include Extended Data	No	Product Report: adds the process data of every product in the unit's genealogy. Export Report Values: adds values from genealogy relatives.
Equipment / Process selection	Usually	What to report on, from the plant model; choosing a higher level takes in everything below it.
Report Time Span / Start Date / End Date	Usually	Custom Selection uses the dates you enter; the relative spans count back from the moment you run the report.
Attribute(s)	Performance reports	The measured attribute(s) to chart.
Count / Reject Items	SPC / Reject Pareto	SPC: how many of the most recent samples to chart (2–10,000). Reject Pareto: how many causes to show (1–20).
Eliminate Zeros / Exclude Outliers	No	Leave out zero readings / values more than 3 standard deviations from the median before charting.
Show Data Table	No	Lists the charted values under the chart, with a CSV download.
Report-specific filters	Varies	Supplier, material, reference process, pass/fail result, group level, month span, UDFs, etc.

Related Topics

- [Products](#) — finding a product; the Product Report destination
- [Recipes](#) — control instances and the limits process data and SPC are judged against
- [Bills of Material](#) — what BoM Consumption / Variance compares against
- [Production Execution](#) — check-out gates (WIP limits), carriers, tray operations, and recording carrier maintenance
- [Product Audits](#) — Audit Results
- [Quarantine & Disposition](#) — holding what a Where-Used search finds; Quarantine Aging
- [OEE](#) — OEE and alarm reporting
- [Dashboards](#) — replaying reports
- [AI Insights](#)
- [System Configuration](#) — the date and time format; the synchronization jobs that keep reports current
- [Reporting Database Data Dictionary](#) — building your own reports on the reporting views

Chapter 13: Dashboards

Overview

A **dashboard** is a saved, auto-rotating slideshow of reports. You assemble an ordered list of report items (the form calls them **pages**) — or, if you need one, another web page such as a report from your report server — give each a display duration, then play the dashboard full-screen so it cycles through them and loops. This chapter covers creating, editing, archiving, and playing dashboards.

Why this matters: the reports in Chapter 12 answer questions when *you* go look. A dashboard answers them when *nobody* is looking — on a monitor over the line, cycling the OEE scorecard, first-pass yield, and WIP so the floor sees its own numbers in real time without anyone running a thing. It turns reporting from a pull into an always-on push.

Before You Begin

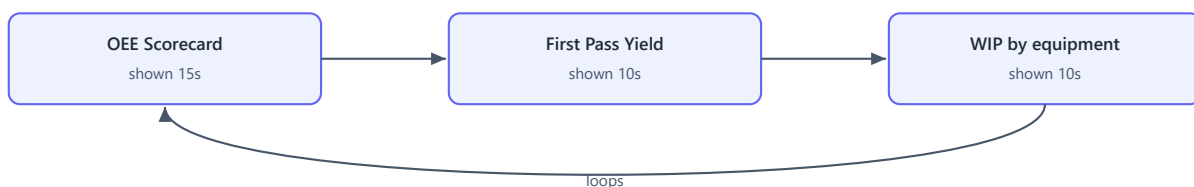
- Dashboards are available to **any signed-in user** — viewing, creating, editing, and playing.
- The reports you include should already run successfully, because a dashboard **replays saved reports** (Chapter 12). Set the reports up first.
- Those saved runs expire if nothing opens them — see [How long a tile keeps working](#).

How to Get Here

Reports → **Dashboards** opens the dashboard list.

How playback works

Playback — an unattended rotation



Full-screen and unattended: each item shows for its own seconds, then the next; an item that cannot be shown is skipped.

Browse dashboards

The list shows active dashboards — each with its **Name**, **Description**, whether it is **Enabled**, and how many **Pages** it has; tick **Show Archived** to include archived ones. From here you **Play**, **Edit**, or **Archive** a

dashboard. **Play** is greyed out for a dashboard that is not enabled or has been archived.

Create a dashboard

1. From **Reports** → **Dashboards**, select **Create New**.
2. Give it a **Name** (this is how it appears in the dashboard list, up to 200 characters) and, if you like, a **Description**. Leave **Enabled** ticked to be able to play it.
3. Under **Dashboard Pages**, select **Add Page** for each item and fill in:
 - **Display Name** — shown above the item while it plays.
 - **URL** — what to show. Paste a report run's **share link** (or just the run's id) to show that report; anything else must be the full address of a web page (`http://` or `https://` , or an address on this site starting with `/`), which plays framed inside the dashboard.
 - **Display Seconds** — how long it stays on screen: 10 to 1000 seconds, 30 unless you change it.
 - **Enabled** — untick to keep the item but leave it out of the rotation.
4. Drag the rows by their handle (≡) into the order the rotation should follow.
5. Save. The dashboard is refused, with the reason shown, if an item has no usable **URL** (*Item '...' requires a valid URL.*) or if it is enabled but has no enabled item.

Tip: Order the rotation by audience attention — lead with the number people most need to see, and keep the loop short enough that it comes around again before they walk away.

Edit a dashboard

Open a dashboard and **Edit** to rename it, add items (**Add Page**) or take them out (**Remove**), change each item's URL or duration, switch an item or the whole dashboard off with **Enabled**, or drag the rows into a new order. Save to apply — the next playback reflects the change.

Play a dashboard

Playback starts the full-screen rotation:

- Each item displays for its configured seconds, then advances automatically and loops.
- Every item fills the screen under its **display name**, with a countdown to the next item and a **Skip** link. **Back to Index** (or the logo) leaves playback.
- Charts — and the OEE state timelines — are drawn to fit the screen. The OEE Scorecard also shows its table — each machine's availability, performance, quality and OEE — under its trend chart, because the table is the scorecard. The table takes up to half the screen; a longer one scrolls, as below.
- Table reports show **only their results**. The report's heading, criteria, share link and buttons are left out, because nobody watching a wall screen can use them, and sections the report page folds away

(such as the OEE event log) are opened.

- A table longer than the screen **scrolls itself** while the item is shown: it holds at the top for a moment, scrolls steadily down, and holds at the bottom before the next item. The floor sees every row, not just the first screenful — so give a long table a longer display duration to slow the scroll down.
- An item re-runs its report each time it comes round, so a relative time span such as *Last 8 Hours* always covers the latest hours. If the report cannot be re-run, the item shows the run as it was saved, and the application log records a warning naming the dashboard and the item.
- An item that cannot be shown at all — usually because the report run it points at has been removed — is **skipped**, and the rotation carries on with the next item instead of stopping on an error page. The application log records which item was skipped and why. To bring the item back, open its report, run it, and paste the run's new share link into the item.
- Items whose **Enabled** box is cleared are left out of the rotation. A dashboard that is not enabled, or has been archived, cannot be played.
- An item whose URL is another web page shows that page, framed under the item's display name, for its display seconds.

Watching a dashboard changes nothing of yours. Although each item re-runs its report, that run is **not** added to report history, and it does **not** change the criteria your own report pages remember — the tile's criteria are the ones saved with its run, not yours. A wall screen can be left signed in without filling anyone's history.

Note — wall screens and the inactivity lock. Moving from one item to the next loads a new page, and the [inactivity lock](#) treats a page load as activity. So while a dashboard is playing, the lock does not engage unless a single item stays on screen for longer than the lock time. Leave playback with **Back to Index** and the dashboard list locks like any other page.

How long a tile keeps working

A tile over a report shows a **saved run** of that report, and saved runs do not live forever: the nightly maintenance job prunes any run that has not been opened for the **Report History Retention Time** (System Parameter 12; 30 days if it has not been set).

Playing the dashboard counts as opening them. Each time a tile comes around in the rotation, its saved run is marked as used and its clock restarts — so a dashboard on a wall screen keeps its own tiles alive indefinitely, with no housekeeping from you.

The case to watch is the dashboard **nobody plays**. Leave one unplayed for longer than the retention window and its saved runs are pruned like any other unused run; the next time you play it, those tiles are **skipped** (see [Play a dashboard](#)), and a dashboard with no tile left to show ends at Not Found. Two ways out:

- Play the dashboard at least once inside the retention window — a single pass through the rotation is enough.
- Or re-run the report (Chapter 12), then edit the dashboard item to point at the new run.

Note: A tile is also skipped — rather than stopping the rotation on an error page — if the report behind it has since been renamed or retired. Re-run the report under its current name and repoint the item.

Archive and restore

- **Archive** takes a dashboard out of the default list (after you confirm on the **Archive Dashboard** page); it is retained, not deleted, and cannot be played while archived.
- Tick **Show Archived** to find it. An archived row is marked **Deleted** (hover for when and by whom) and offers **Details** — the dashboard read-only — and **Restore**, which returns it to the active list, in place of **Play**, **Edit** and **Archive**.

Field Reference

Dashboard (create/edit):

Field	Required?	What it does / valid values
Name	Yes	Shown in the dashboard list. Up to 200 characters.
Description	No	Shown in the dashboard list. Up to 500 characters.
Enabled	—	Only an enabled dashboard can be played; it must then have at least one enabled item.
Page: Display Name	Yes (per item)	Shown above the item while it plays. Up to 200 characters.
Page: URL	Yes (per item)	A report run's share link or id, or the address of a web page (<code>http://</code> , <code>https://</code> , or starting with <code>/</code>).
Page: Display Seconds	Yes (per item)	Seconds shown before advancing: 10 to 1000; 30 by default.
Page: Enabled	—	Whether the item is in the rotation.
Order	—	Drag the rows; the rotation follows the list from top to bottom.

Related Topics

- [Reports](#) — the reports a dashboard replays
- [OEE](#) — including the live **Active Alarms** board

- [AI Insights](#)
- [Electronic Signatures & Audit Trail](#) — the inactivity lock

Chapter 14: AI Insights

Overview

IntelliTrack AI can analyze your manufacturing data and answer questions in natural language. You submit an **insight request** to a configured AI provider, read the response (including any generated files), and continue with follow-ups. The assistant reads IntelliTrack's own reports to answer — the same reports you can open, with your permissions — and links each answer to the report runs it used. Supporting it are the **AI memory** (what the assistant retains, with a human review gate), **AI connections** (your own AI provider accounts, and which one each kind of AI work uses) and **budgets** (limits on estimated AI spending). This chapter covers them all.

Why this matters: an AI assistant is only useful if it is **grounded and governed**. Connections use your own provider accounts, keep the keys encrypted and never show them again, and send each kind of work only to the connections an administrator chose for it; the memory review gate means a human decides what the assistant is allowed to remember and reuse. Together they let you get AI analysis of plant data without surrendering control of your credentials, where your data goes, or what the model treats as fact.

Before You Begin

- Anyone who signs in can use **AI Insights**: ask questions, hold conversations, and use **Ask AI** on a report. You see and act on your own conversations; administrators also see everyone's in the list.
- **AI Settings** (switches, egress, glossary, connections, routing, chat availability, budgets) and **AI Memory** require the **Administrator** role.
- Before anyone can run insights, an administrator must **add a connection** and assign it to the **Chat** workload (see below). The setup checklist on AI Settings walks through it.
- **What you need from your AI provider.** IntelliTrack uses your own provider accounts and never supplies one. For each provider you connect you need an account with **OpenAI** or **Anthropic**, access to the models you want to use, and an **API key** for IntelliTrack. A key of its own, with its own spending limit at the provider, lets you see and cap what IntelliTrack uses. Your IT administrator must allow the web servers to reach the provider over HTTPS.

How to Get Here

- AI Insights: **Reports** → **AI Insights**. The menu shows it only while a connection can take chat.
- **Ask AI**: on any report that shows **Copy link**, **Ask AI** beside it opens a new request bound to that report run.
- AI Settings: **Configuration** → **AI Settings**, or, for an administrator, the **AI Settings** link at the foot of the AI Insights list.

- At the foot of the AI Insights list, **Manage Memory Items** opens **AI Memory Management**. When Chat has no usable connection, the new-request page says so and offers **AI Settings** to an administrator.

Turn AI on or off — AI Settings

AI comes in two layers, each switched on or off for this site on **AI Settings**:

- **Insights** — the deterministic layer: detectors and findings, containment, curve similarity, recipe replay, prior occurrences and resolution search. It needs no AI provider, and no data leaves the plant.
- **Assist** — the model layer: AI chat (the insight requests described below), report tools, narration, drafts, image review, guidance and summaries. It takes effect only while **Insights** is on, and each kind of work needs a connection assigned to it (see *Routing* below).

Both are on until an administrator changes them. **Backfill window (days)** says how far back Insights looks when it is first turned on or turned back on. **Save** records the change with who made it, in the audit trail.

The **Status** card says whether each layer can run now, and if not, why: *Switched off*, *No AI provider configured*, *Not available: IntelliTrack is not licensed*, or *Off: the AI settings could not be read* (AI stays off until they can be read again; nothing else in IntelliTrack is affected). A change made on another server of a multi-server installation takes up to a minute to show.

When Assist cannot run:

- **New AI Request** is not offered on the AI Insights list, which says why; the list and every existing conversation stay readable, and so do downloads.
- Opening a new request, or asking a follow-up, shows **AI is not available** with the reason instead of sending anything to a provider. *No AI provider is configured* (nothing is assigned to Chat) and *switched off* are different messages.
- Memory compaction stops: jobs already waiting are cancelled, and no new ones are queued. The conversations they were for are queued again once Assist is back on. The same happens when no usable connection is assigned to **Summaries**, the workload memory compaction runs on.

If the installation is not bound to a site, AI Settings says so and saves nothing.

Set up AI — the checklist

Until setup is complete, AI Settings opens with a **Setup checklist**: review the Insights and Assist switches, add a provider connection, assign every workload, and review what may be sent to AI providers (egress; saving AI Settings once completes it), and switch on at least one notification channel (Notification Channels, [Chapter 15](#)). Each step links to where it is done. The checklist disappears once every step is done.

Decide what may leave the plant — egress

The **What may be sent to AI providers** card on AI Settings applies to every connection and every kind of AI work:

- **Images** — pictures attached to a request, such as inspection images. **Off** as shipped. While off, a request with an image attached is refused before anything is sent, and says why; other files are still sent.
- **Free text** — what people typed into IntelliTrack, such as notes, comments and reasons, where it appears in report data the assistant reads. **On** as shipped. While off, those columns are left out of what the assistant receives, and it is told which were withheld. Your own question is always sent.

Save records the change with who made it, in the audit trail.

Help the assistant with your plant's words — site glossary

The **Site glossary** on AI Settings holds the site's own terms — line names, local words and what they mean, one per line (up to 4,000 characters). It is given to the assistant with every chat. **Save** records it with the switches.

The **Reports the AI can use** card counts the reports the assistant can read. A report whose card has no description is listed there: the assistant uses it only when you paste its link or use **Ask AI** on it.

Connect an AI provider — connections

A **connection** is one of your AI provider accounts: a name, the provider, an endpoint when the provider needs one, the API key, and the models to use with it. Add as many as you need, for example one per provider, or separate keys for separate budgets or departments. The providers offered are those IntelliTrack has an adapter for: **OpenAI (ChatGPT)** and **Anthropic (Claude)**.

1. On **AI Settings**, under **Connections**, select **Add connection**.
2. Enter a **Name**, choose the **Provider**, and paste the **API key** from your provider account. Leave **Endpoint** blank unless your provider told you to use a particular one.
3. Under **Models**, open a row's list and pick a model (for example `claude-sonnet-4-5` or `gpt-4o`). The list is the provider's own: the models your key can use. To use one it does not list, type the name and press **Enter**. The **x** on a row clears it, which removes that model when you save.
4. **Save and test**. IntelliTrack stores the key encrypted and makes one short call to each enabled model. Each model's **Last test** and **Capabilities** (tool use, structured output, vision, streaming, context size) are recorded with who ran the test.
5. If some workloads have no connection yet, IntelliTrack then offers to assign the new connection to the ones it can handle. Tick the ones you want and **Assign selected**; nothing is assigned until you confirm.

The key is write-only. After saving, only its last four characters are shown (*ends in 1a2b*). To replace it, type a new key on **Edit**; leave the field blank to keep the current one. Replacing the key, renaming, or changing the models keeps the connection and every place it is used.

Test connection on the edit page tests every enabled model again. A failed test says why (for example *Authentication*: the provider refused the key) and keeps the capabilities from the last successful test.

Delete is refused while routing or chat availability uses the connection, and the page lists where, so you can reassign it first. A deleted connection keeps its history; **Show deleted** on AI Settings lists it with **Restore**.

If the page says credentials cannot be stored safely on this server. Keys are encrypted with keys the server keeps in a folder. When the application cannot write to that folder, a key stored now could not be read after the next restart, so IntelliTrack refuses to store one. Your IT administrator must allow the application to write to the folder (the installer normally does this) and restart the application.

If a connection's key cannot be read on this server (typically after the database was restored onto a different server, or the server's key folder was deleted or moved), that connection is skipped and a test says so. Enter the key again on **Edit**.

Choose where each kind of work goes — routing

Every AI call belongs to one **workload**. Each workload needs certain model capabilities:

Workload	Used for	Requires
Chat	AI Insights conversations, their rolling summaries, report tools	tool use, structured output, streaming
Narration	Explaining findings	tool use, structured output
Drafting	Drafting reports such as 8D	structured output, large context
Image review	Reviewing images	vision, structured output
Guidance	Troubleshooting guidance	structured output
Summaries	Shift and recipe summaries, AI memory compaction	structured output
Validation Package Generator	Assessing a URS for a validation package (Chapter 21)	tool use, structured output, large context

The **Routing** card on AI Settings shows, for each workload, its **Primary** connection and model and its ordered **Fallbacks**. **Edit** a workload to choose them; only models whose last successful test shows every required capability are offered.

How a request is sent:

- It goes to the primary. On a provider outage (a timeout, a server error or a rate limit) it is tried once more, then goes to the fallbacks in order.
- It never goes to a connection that is not listed for that workload, so your data reaches only the providers you chose for that kind of work.
- An authentication, permission or request error is shown, not retried on another connection: it is something to fix, and another connection would hide it.
- An entry that cannot be used is skipped and marked on the Routing card: its connection or model is disabled, its key is missing, or a later test no longer shows a capability the workload needs (*flagged*).
- A workload with no usable entry shows *no usable connection*, and its features say *No AI provider configured*.
- An entry whose budget has no room is passed over for the next one (see budgets, below).

Every call is recorded with its workload, connection, model, whether a fallback was used and why, the tokens it used and its estimated cost.

Offer connections in chat — chat availability

By default a new AI Insights request goes to the Chat primary (with its fallbacks). Under **Chat availability**, an administrator can also offer other models for users to choose, each to everyone or only to some roles. Only models that can handle Chat can be offered. A connection a user chooses is used on its own: if it fails, the user sees why and can retry on the default; it never switches to another connection silently.

Limit AI spending — budgets and rates

IntelliTrack meters every AI call — its workload, connection, model, tokens and an **estimated cost** — and an administrator can set **budgets** that cap that estimated spending. Nothing is invoiced: the costs are estimates from a rate table, so you can see and manage what your provider accounts are likely to be charged.

Budgets. On AI Settings, the **Budgets and usage** card lists each budget with its **Limit**, what has been **Spent (estimated)**, what calls in progress have **Reserved**, what is **Remaining**, and when it **Resets**. **Add budget** to create one:

- **Applies to:** the **Whole site**, **One connection**, or **One workload**.
- **Period: Daily (UTC) or Monthly (UTC).** Days and calendar months are UTC; the reset time is shown in site time.
- **Limit:** at least 0.01, in the rate table's currency (US dollars).

A site can have one budget per scope and period — for example a monthly budget for the whole site and a daily one for a single connection. **Edit** changes the limit only; to change what a budget applies to or its period, delete it and add another. A new or changed budget takes effect within a minute. A budget added part-way through a day or month counts what that period has already spent.

How a budget works:

- Before each call, the most the call could cost — its prompt and data, plus the largest answer the connection allows — is reserved against every budget that covers it. When the call returns, the reservation is replaced by what the call actually cost.
- If a budget has no room, the call goes to the next connection listed for its workload, and the call is recorded as a fallback for the reason *Budget*.
- If no listed connection has room, the user sees which budget stopped the request and when it resets. A connection the user chose in chat is not replaced by another.
- Background work (AI memory compaction) pauses without using up its retries, and resumes when there is room again.
- A model that has **no rate** cannot be priced, so while a budget covers it, calls to it are refused. The Budgets and usage card lists such models, each linked to where its rate is set.
- With no budgets, nothing is reserved; every call is still metered.
- AI Insights (the deterministic layer) is never limited by a budget.

Estimated usage. The same card shows, by connection and by workload, the calls and estimated cost today and this month (UTC), and the tokens used this month. Calls to a model without a rate are counted, but their cost is not in the estimate.

Export usage. Choose a **From** and **To** date (UTC) and select **Export usage** to download a CSV of every metered call in the range: when it started, workload, feature, connection, provider, model, attempt, whether it was a fallback and why, whether it succeeded, tokens (input, cache read, cache write, output), images sent, estimated cost, duration, user and conversation. It includes no prompt, answer or error text. A range of more than 100,000 calls is refused; choose a shorter one.

Rates. **Rates** on the Budgets and usage card opens **AI rates**, priced per million tokens:

- **Models in use** lists each connection's models and the rate each is priced at: **Yours**, **Shipped**, or **No rate**. **Set your rate** opens the rate form prefilled with the rate in force.
- **Your rates** lists the rates you have set. **Use the shipped rate** removes yours, so the shipped rate (if there is one) applies again.
- **Shipped rates** is the table delivered with this release, taken from the providers' list prices on the date shown.

Set your own rate when your agreement with a provider differs from its list price, or for a model the shipped table does not list. A rate applies to the model named and to its dated versions (for example -20250929 or -2025-08-07), never to a different model. A saved rate prices calls within a minute; the cost already recorded for earlier calls does not change.

Run an AI insight

1. Open **Reports** → **AI Insights**. Previous requests are listed newest first — when, the **Provider** and **Model**, the **Original Prompt**, the **User** who asked, and **Do Not Prune**; search to filter (paged).
2. Select **New AI Request**. **Connection** offers **Default** (the Chat primary, with its fallbacks) and any connections your administrator made available in chat to your role. Your last choice is preselected next time.
3. Optionally give it data to work from (**Data Source**): paste a **Report History Link** — a report run's share link — and/or **Upload a File** (CSV, TXT, JSON, PDF or Excel; each model has its own size limit). A pasted link runs that report again with the same settings, for you and with your permissions, and the assistant receives a summary of the result: row counts, totals and ranges over every row, and the first rows. A run of a report the assistant cannot run itself (such as a product history) is summarized as it was saved. With neither, the assistant can still look reports up itself.
4. Choose a **Prompt Template** — *Summary & Insights*, *Quality Analysis*, *Throughput & Bottlenecks* or *Variance & Anomalies* — or *Custom Prompt* and write your own **Prompt**. **Insight Type** guides the kind of analysis (*Auto* picks one from your data and prompt); **Advanced AI Guidance** and **Conversation Defaults** are optional refinements.
5. **Run Analysis**. IntelliTrack sends the request, stores the answer, and opens **AI Conversation Details**: the response, its data source, and the tokens it used. To answer, the assistant may search the reports, run the ones it needs with the settings the question calls for, and read their summaries — only reports you may open, over at most a year at a time, and only a few per question. Each report it runs is saved as a run of your own; the answer links to them in the text and lists them under **Sources**, and a link opens exactly the data the answer used (for as long as report runs are kept). If no answer comes back, nothing is stored and the page says why; when you had chosen a connection, **Retry on the default connection** sends the same request to the default instead.
6. **Ask a follow-up** to continue the conversation (you can attach another report link or file), and download any files an answer produced. A conversation stays on the connection that first answered it, because its files and history live with that provider; a follow-up that gets no answer is not stored. If that connection has since been deleted, start a new conversation.

The list shows your own conversations; administrators also see everyone's. A conversation can be opened, continued, and deleted only by the person who started it.

Manage insight retention

Insights are pruned over time: the reporting database's maintenance purge deletes a conversation once its latest answer is older than the **Report History Retention Time** (System Parameter 12; 30 days as shipped). To keep an important one, tick **Do Not Prune** (on the list or on its details page). Delete ones you no longer need individually — **Delete** removes the conversation and all its messages permanently — or select several on the list and delete them together. Do Not Prune and Delete act only on your own

conversations. Your AI **Preferences** (**Audience Level**, **Confidence Level**, **Recommendation Style**, **Tone**) set the defaults for your new requests.

An insight keeps the report run it works from. A saved report run is normally removed once nobody has opened it within the **Report History Retention Time** (System Parameter 12 — see [Reports](#)). The run an insight works from is the exception: for as long as the insight exists, the Reporting database's maintenance purge leaves that run in Report History, so the link under the insight's **Data Source** keeps opening. Pinning an insight with **Do Not Prune** therefore keeps its run as well. A follow-up that attaches a different report, or only files, replaces what the insight works from, and the earlier run falls back under the ordinary retention rule. A plain question with nothing attached holds no run, and never stops the purge from retiring the runs nobody is using.

Manage AI memory — the human review gate

AI Memory is what the assistant retains to give context-aware answers; **AI Memory Management** (**Manage Memory Items** on the AI Insights list) lets a human govern it. Its cards count the **Active Scopes**, **Active Memory Items**, items **Pending Approval** and **Failed Jobs**, each linking to the list behind the number.

- Browse memory **scopes** and their **items** (filter by **Type**, or search). Deleted items are left out unless you tick **Show deleted**; then each is marked **Deleted** and offers **Reactivate** beside **View**.
- Open an item to act on it — this is the gate over what the assistant is allowed to treat as known:
 - **Approve** or **Reject** an item that is waiting for approval (a rejection can carry a reason);
 - **Delete** an active item, with an optional reason — from then on it is left out of what the assistant is given;
 - **Reactivate** a deleted item.
- Each item's **Audit History** lists every action on it — who performed it (**Performed By**), the **Date** and the **Reason** — including the ones IntelliTrack takes itself, shown as performed by **System**: items **Expired** at their expiry date, and items **Pruned** to keep a scope within its limit or because they were old and of low importance.
- Review **Compaction History**, where compaction consolidates a conversation's memory. **Enqueue Manual Compaction** asks for the conversation's ID (the one at the end of the conversation's address on AI Insights). A compaction that keeps failing is recorded as **Failed** and counted under **Failed Jobs**; that conversation is then not compacted again until it gets a new message or somebody enqueues a manual compaction for it.

Warning: Deleting or rejecting memory items changes what the assistant knows going forward. Review item details before acting — a wrongly approved "fact" propagates into future answers.

Field Reference

AI insight request:

Field	Required?	What it does / valid values
Connection	Yes	<i>Default</i> (the Chat primary, with fallbacks) or a connection made available in chat to your role.
Report History Link	No	A report run's share link: that report is run again with the same settings, for you, and summarized for the analysis. Ask AI on a report fills it in.
Upload a File	No	CSV, TXT, JSON, PDF, or Excel files; each model has its own size limit.
Prompt Template	One of these two	A ready-made request, or <i>Custom Prompt</i> .
Prompt	Yes, with <i>Custom Prompt</i>	What you want analyzed or answered.
Insight Type	No	The kind of analysis; <i>Auto</i> chooses one from your data and prompt.

AI Settings:

Field	Required?	What it does / valid values
Insights	No (on by default)	Turns the deterministic layer on or off for this site.
Assist	No (on by default)	Turns the model layer on or off; effective only while Insights is on.
Backfill window (days)	Yes	1 to 3650; 90 as shipped.
Images	No (off by default)	Whether images may be sent to AI providers, for every connection.
Free text	No (on by default)	Whether free text people typed (notes, comments, reasons) may be sent in report data.
Site glossary	No	Up to 4,000 characters of the site's own terms, given to the assistant with every chat.

AI connection:

Field	Required?	What it does / valid values
Name	Yes	Up to 200 characters, unique among this site's connections.
Provider	Yes	OpenAI (ChatGPT) or Anthropic (Claude); fixed once saved.
Endpoint	No	An absolute http or https address, only when the provider requires one.

Field	Required?	What it does / valid values
API key	Yes (to add)	Stored encrypted; only its last four characters are shown after saving. Blank on edit keeps it.
Enabled	No (on by default)	A disabled connection is skipped; one in use by routing or chat cannot be disabled.
Models	Yes	Up to 50 model names; each can be enabled or disabled, and a name cleared to remove it.

Routing and chat availability:

Field	Required?	What it does / valid values
Primary, Fallback 1, 2, ...	No	A connection model with every capability the workload requires; up to 10 in all.
Offered	No	Makes a model a choice in AI Insights.
Roles	No	Who may choose it; none ticked means everyone who may use chat.

Budget:

Field	Required?	What it does / valid values
Applies to	Yes	Whole site, one connection, or one workload; fixed once saved.
Connection	With <i>One connection</i>	A connection of this site.
Workload	With <i>One workload</i>	Chat, Narration, Drafting, Image review, Guidance, Summaries or Validation Package Generator.
Period	Yes	Daily (UTC) or Monthly (UTC); fixed once saved.
Limit	Yes	At least 0.01 and at most 1,000,000,000, in US dollars; an estimate.

AI rate (per million tokens, US dollars):

Field	Required?	What it does / valid values
Provider	Yes	OpenAI (ChatGPT) or Anthropic (Claude).
Model	Yes	The model name, up to 150 characters; also covers its dated versions.
Input, Output	Yes	0 to 100,000.
Cache write, Cache read	Yes	0 to 100,000; input served from or written to the provider's prompt cache. For OpenAI, set cache write to the input rate.
Image	Yes	0 to 100,000; the rate for image tokens a provider reports separately.

Related Topics

- [Reports](#)
- [Dashboards](#)
- [Notifications](#)
- [System Configuration](#)

Chapter 15: Notifications

Overview

IntelliTrack can tell people when something happens — by **email**, **text message (SMS)**, in **Microsoft Teams** or **Slack**, and inside IntelliTrack, in their **Inbox**. You define **publications** (what to be told about, and where in the plant), set up **user profiles** with email addresses and phone numbers, and create **subscriptions** that connect a person to a publication. An administrator can also send a publication to **everyone holding a role** and post it to **Teams and Slack channels**. Messages leave IntelliTrack through the site's **notification channels**: its own mail server (set on **Email Server (SMTP)**, [System Configuration](#)), its own Twilio account for text messages, and the Teams and Slack webhooks an administrator adds. This chapter covers all of these.

Why this matters: the value of track-and-trace is acting on a problem quickly. A notification turns a condition the system already detects — a product put on hold, a machine down too long, a work order completed — into a message that reaches the right person on the channel they actually watch, without anyone monitoring a screen. The publication/subscription split is what lets one event reach many people, each on their own terms.

Before You Begin

- **Publications, subscriptions, profiles and phone numbers** are available to **any signed-in user** for their own; **Administrators** can manage anyone's, and see every private publication.
- **Notification channels**, a publication's **Audience** and the **Delivery Log** require the **Administrator** role.
- For anything to leave IntelliTrack, an administrator must set up the site's channels (see [Set up notification channels](#)).
- You need a notification profile of your own before you can subscribe: the Subscriptions page sends you to create one first.
- A phone number is sent text messages only after it **agrees to receive them** (see [Agree to receive text messages](#)).

How to Get Here

Configuration → **Notifications** opens the **Notifications** page, with a tile for each notification page your role can open:

- **Inbox** — the notifications sent to you inside IntelliTrack. It is also the **Inbox** button under your name, at the right of the menu bar on every page.
- **Subscriptions** — who receives what.

- **Publications** — what can be notified. A publication's **Audience** (administrators) is part of its Create, Edit and Details pages.
- **User Profile** — notification profiles, their email addresses and phone numbers.
- **Delivery Log** and **Notification Channels** — administrators only. The channels are also a tile under **Configuration** → **Additional Configuration** → **Connections & Notifications**, beside the email server.

These pages used to be separate items of the Configuration menu.

- **Configuration** → **Additional Configuration** → **Email Server (SMTP)** — the mail server notification email goes through (Administrators).
- The link in an email, text or Teams/Slack post opens the notification in IntelliTrack, for its recipients and administrators.

The links at the foot of each list lead back to the **Notifications** page and on to the pages that go with it: **Inbox** links to **Subscriptions**; **Publications** and **Subscriptions** link to each other; **Subscriptions** also opens your own **Profile**; **User Profile** links to **Subscriptions**; **Delivery Log** and **Notification Channels** link to each other. A notification opened from an email or text links to the **Notifications** page and to your **Subscriptions**.

How a notification reaches someone

Event → publication → subscription → delivery



Message text can include tokens such as the event's name and description, filled in from the published event's data at send time.

A send that fails is tried again; one that cannot succeed is recorded as not sent, and every send is in the Delivery Log.

A message goes out when a publication fires, to:

- each **subscriber** whose subscription is enabled, whose **Minimum Severity** the message meets, and who may view the publication's type;
- each member of a **role group** on the publication's Audience — the people IntelliTrack recorded in that role when they last signed in at this site — who may view the publication's type;
- each **Teams or Slack channel** on the publication's Audience.

Then, per send:

- the site's channel for it must exist and be switched on, and the message must meet the channel's own **Minimum Severity** (text messages default to **Alert** only);
- an hourly cap per recipient and per channel holds back a flood; what it holds back is recorded, not queued;
- a text message goes only to a number that has agreed to receive them, and carries the **whole notification** — severity, title, the text and a link. When that fits one text message it goes as SMS; when it is longer it goes as **one MMS** carrying the whole text (no picture), to North American (+1) and Australian (+61) numbers. Elsewhere, where MMS does not reach, a long notification goes as one short text — severity, title and the link — and opens in IntelliTrack. Summaries and digests go by text the same way;
- a send that fails for a reason that can clear (a server down, a time-out) is tried again after 1, 2, 4, 8 and 16 minutes; after the sixth try, or at once when it cannot succeed (an address that does not exist, a number that replied STOP), it is recorded as **not sent**;
- a publication that fires again for the same thing within its **Cooldown Minutes** is not sent again; the repeat is counted on the first message.

Set up a user profile and addresses

A **user profile** is a notification recipient: a person, the channels they accept, and the addresses to reach them.

1. **Configuration** → **Notifications** → **User Profile** opens **Notification Profiles**; select **+ Create Profile**. Your profile is created under your IntelliTrack user name (**Username**), which is how the Subscriptions page recognises you. An administrator can instead pick any existing user or type a name.
2. Under **Channel Permissions**, tick the channels this person accepts: **Allow In-App Notifications**, **Allow Email Notifications**, **Allow SMS Notifications**. They start unticked, and a channel that is not allowed here is never delivered, whatever a subscription asks for. There is no separate MMS choice: a long text goes as MMS by itself where MMS reaches (see above).
3. **Create Profile**, then add **Email Addresses** and **Phone Numbers**, each with an optional **Label** (*Work, Cell...*) and a **Primary** mark.
4. Enter a phone number **with its country code**, a plus sign first — for example `+1 555 010 2030`. IntelliTrack stores it as `+15550102030`; a number marked for text messages without a country code is refused, because it could not be texted.

A profile with no addresses can be subscribed but has nowhere to send email or text messages — so add the addresses first.

Note: In-App notifications go to the person's **Inbox** inside IntelliTrack (see [Check your notifications \(Inbox\)](#)), where they see them the next time they open IntelliTrack. Give anyone who must hear at once, away from a screen, an email address or phone number as well.

Agree to receive text messages

A number is sent notifications by text only after it agrees to. On **Configuration → Notifications → User Profile → Phone Numbers**, the **Texts** column shows each number's state — **Receives texts, Not agreed**, or **Replied STOP** — and what you can do:

1. **Send a code** texts a six-digit code to the number through the site's SMS channel.
2. **Enter the code** on the page that opens. It lasts **10 minutes** and survives five wrong tries; **Send another code** sends a fresh one (at most one a minute, and five an hour, per number).
3. The number then **Receives texts. Stop texts** ends it at any time, and so does replying **STOP** to any text — IntelliTrack records that when the carrier reports it, and shows it to administrators.

Changing a number ends its agreement: the new number must agree again.

The page and the code text name your company, taken from **Company Name** on **Configuration → Additional Configuration → Security & Administration → Branding**. They also say that how often texts come varies, that message and data rates may apply, and that replying **HELP** gets help and **STOP** stops the texts. US carriers require all of this before they deliver texts from a business. If you text US numbers, set the company name to the business name Twilio approved for your sending number: the business on its toll-free verification, or the Brand registered for A2P 10DLC.

Administrators can also **Record their agreement** for a number whose owner agreed another way — on a signed form, for example — and cannot enter a code. Say how and when the person agreed; the words are kept with the record. Every agreement and every stop is recorded with who recorded it and when, and is never edited afterwards.

Define a publication

A **publication** says what to be told about: one kind of event or condition, where in the plant, and how the message reads. Everyone subscribed to it gets the same message.

1. **Configuration → Notifications → Publications** lists the publications you can see — their **Name**, **Description**, **Type**, who they were **Created By**, whether they are **Enabled**, and when they were **Last Updated**; search to filter (paged).
2. **Create New**, then choose the **Publication Type Code** — the kind of notification (see [What you can be notified about](#)). Its settings appear under **Publication Settings**, and most types add a plant picker to limit the publication to part of the plant (leave a level at **-- All --** to include everything beneath it). The type cannot be changed once the publication is saved.
3. Give it a **Name** and **Description**, and set:
 - **Is Active** — only an active publication is evaluated and sent.
 - **Is Private** — hides the publication from everyone except the person who created it and administrators: it is left off their list, its pages do not open for them, and they cannot

subscribe to it. Create your own notification profile before creating a private publication: the creator is recorded through it.

- Adjust the type's settings — its thresholds and filters, the **Severity** of its messages, **Cooldown Minutes** where offered (how long to stay quiet about the same item after notifying), and the **Title Template** and **Body Template**. The templates start with the type's default wording, and the hint beneath them lists the tokens you can use, such as `{Name}`, `{EquipmentName}` or `{OccurredAt}`.
- Save**, **Edit** and **Delete** are on the list.

A publication belongs to the site it was created at, and its messages go out through that site's channels.

Note: Tokens are filled in from the event's data when the message is sent, so a single template produces a specific, readable message per occurrence.

What you can be notified about

The **Publication Type Code** list offers the types below, by name. A type shown as *(not yet available)* cannot be chosen.

- Types that react to an event fire as soon as IntelliTrack sees it. Some events are seen in the reporting database — first and last piece, work order status, SPC signals, expired lots — so those follow the synchronization that carries production data there.
- The "too long" types wait for their threshold, and can send an all-clear (**Notify On Resolution**) when the condition ends.
- Scheduled types run on their **Recurrence** — the first run within about a minute of creating the publication, then once per period from then on.
- Some types reach only people in a given role, whoever subscribes: **License Expiring Soon**, **Database Job Failure**, **Notification Delivery Failure Detected** and **Reporting Data Stale Detected** reach **Administrators**; the two electronic-signature alerts reach **Audit** and **Administrators**. The subscription page offers them only to those people.
- The summaries and digests (**Shift Alarm Summary**, **Alarm Bad Actors Digest**) go by every channel. By text they are usually longer than one SMS, so they arrive as one MMS carrying the whole summary.

When something happens

Publication type	What raises it
Hold Placed	A product put on Hold from its Edit page, or a confirmed quarantine that holds products — one notification per quarantine, naming it and how many parts it held (Quarantine & Disposition).
Manual Product Advancement	Somebody changes a product's disposition or operational status by hand on its Edit page, outside the normal sequence.

Publication type	What raises it
Negative Inventory Adjustment Posted	A product's amount is lowered on its Edit page.
Failed Inspection or Test Recorded	An audit is completed with any entered value out of specification — even when the audit's overall result is a pass. Can be limited to where the product was last processed (Product Audits).
First Piece Produced	The first good piece at the start of a new work order, after a recipe or material change, or on restarting after extended downtime (you choose which).
Last Piece Produced	A work order reaching its planned quantity, or the last good piece of a recipe before the machine changes material (you choose which).
Work Order Status Changed	A work order's status changes.
Material Approval State Changed, Recipe Approval State Changed, BOM Approval State Changed	The approval status of an existing material, recipe or bill of materials changes — not its creation, and not rejecting or revoking a pending request.
SPC Out-of-Control Signal Detected	A numeric process attribute breaks a Western Electric rule, judged against control limits worked out from that attribute's own readings on that machine since its last recipe change. One notification when the stream goes out of control, naming the attribute, the rule and the chart statistics.
Attribute Recorded Outside Control Instance	A value is recorded for an attribute that is not part of the control instance the part ran under — one mapped to the module after the instance was built — so it was saved but not evaluated; raised once per control instance and attribute, one reporting synchronization after the check-in (Recipes).
Machine Alarm Raised	A machine raises an alarm, filtered by class, category and product impact (OEE).
Alarm Work Request Raised	Somebody presses Request work on the Active Alarms board (OEE).
License Expiring Soon	Checked daily: a license expiring within Expiring Within Days , optionally only licenses allocated to chosen sites.
Database Job Failure	An unattended job in the reporting database fails — the Site-to-Reporting synchronization, the work-in-process aggregate rebuild, the OEE roll-ups, or the maintenance purge. Nobody watches those jobs run, so subscribe the person who looks after the databases.
Notification Delivery Failure Detected	A delivery is given up on — after its last retry, or refused outright (an address that does not exist, a number that replied STOP) — one notification per failing channel.

Publication type	What raises it
Electronic Signature Security Alert	Attempted misuse of electronic signatures: a person blocked after repeated failed signing attempts, failures across the system above the alert threshold, a user id that is not the signed-in person, or a directory check that did not use Kerberos (signature attempts). Subscribe your security or quality contact.
Electronic Signature Preflight Failed	The signature readiness preflight starts failing, so domain signatures are refused until it passes (signature readiness).

When a condition lasts too long

Publication type	What raises it
Machine Down Too Long	A machine's downtime event stays open past Threshold Minutes (60 unless you change it), with a reminder every Reminder Interval Minutes . Only states that track events start it, and planned downtime — a stop in an unscheduled mode — is ignored (OEE).
Lot Expired in Active Inventory	A lot passes its expiry date, plus Grace Period Days , while its disposition still allows work. A lot held, rejected or used up before then is never reported.

On a schedule

Publication type	What it sends
Shift Alarm Summary	One summary of the alarms the chosen machines raised over the last Lookback (hours) (OEE).
Alarm Bad Actors Digest	The alarms that made most of the noise over the last Lookback (days) , and their share of the load (OEE).
Reporting Data Stale Detected	No reporting synchronization has completed within Max Sync Lag Minutes — the reports and dashboards are going stale.

Types not yet available

These appear in the list as *(not yet available)* and cannot be chosen: **Integration Failure**, **IntelliTrack Application Error**, **Nonconformance Created**, **OEE Below Target**, **OEE Report Distribution**, **Scheduled Report Distribution**, **Scrap Rate Threshold Exceeded**, **Shift Production Summary**, **Top Scrap Reasons Digest** and **Transactional Reporting Sync Lag Exceeded**.

Create a subscription

A **subscription** connects a profile to a publication so that person is notified when it fires, on the channels and at the addresses they choose.

1. **Configuration** → **Notifications** → **Subscriptions** lists your subscriptions — the publication, the channels, the minimum severity and whether each is enabled — with **Edit**, **Details** and **Delete**. Administrators can tick **Show all subscribers** to see every person's, with a **Subscriber** column. If you have no notification profile yet, it sends you to create one first.
2. **Create**, and fill in:
 - **Subscriber** — you; an administrator can choose any profile.
 - **Publication** — the publications the subscriber may view: not another person's private ones, not inactive ones, and not those of a type their roles do not reach (see above). Each is listed by name with its description beneath, and typing finds a word in either, so two similarly named publications can be told apart; the chosen one's description stays under the box. IntelliTrack checks again when you save.
 - **Minimum Severity** — **Information**, **Warning** or **Alert**; messages below it are not sent to the subscriber.
 - **Suppression Window Minutes** — your personal rate cap: after a message from this publication reaches you, further ones are skipped (not queued) until this many minutes have passed. 0 receives every one.
 - **Enabled** — clear it to pause the subscription without deleting it.
 - The channels — **In-app**, **Email**, **Text message (SMS)** — and, under **Delivery Addresses**, which of the profile's email addresses and phone numbers to use. Email and text messages go **only** to the addresses chosen here, so the page keeps each channel and its addresses together:
 - Ticking **Email** or **Text message (SMS)** enables its addresses and, when none is chosen, chooses the profile's **Primary** one (for text messages, the primary number that can receive texts). A new subscription starts with **Email** ticked and the primary email address chosen, when the profile has one.
 - Clearing a channel clears and disables its addresses, so none is kept for a channel that is off.
 - **Email** and **Text message (SMS)** can be chosen only when this site has a working channel for them: an e-mail channel that is switched on, with the **Email Server (SMTP)** page's **Host** and **From Address** filled in; an SMS channel that is switched on, with its Twilio secret stored. Otherwise the box is greyed out and says why, and an administrator sees a link to **Notification Channels**. **In-app** is always available.
 - A subscription that already had **Email** or **Text message (SMS)** when its channel went keeps it: the box shows ticked and greyed out, saving the rest of the form leaves it and its addresses as they were, and it sends again once an administrator sets the channel up. **Details** marks it **not set up on this site**. Giving the subscription to another subscriber drops it, since its addresses were the previous person's.
3. **Save**. A ticked **Email** with no email address chosen, or a ticked **Text message (SMS)** with no number that can receive texts, is refused with a message under the channel and nothing is saved: choose an address, clear the channel, or — if none is listed — add one to the profile first. A ticked

channel the site cannot send by (a page opened before the channel was switched off, say) is refused the same way.

Teams and Slack are not on a subscription. A Teams or Slack webhook posts into a shared channel, not to a person, so an administrator adds the channel to the publication's **Audience** instead (see the next section) and each message is posted there once, whoever subscribes.

A message goes out on a channel only when all of these agree: the subscription enables the channel, the profile's **Channel Permissions** allow it, the subscription has an address for it, and — for text messages — the number has agreed to receive them.

This split — define the event once, subscribe many recipients — is why you can add a new manager to "all hold alerts" without touching the event itself.

Send a publication to a role or a channel (Administrators)

A publication's **Audience** adds recipients beyond the people who subscribe themselves. Administrators see it under the publication's own fields on its **Create** and **Edit** pages (read-only on **Details**), and it is saved with the publication by the page's one **Save**: both are saved, or, if anything is refused, neither is and the page says why.

- **Role groups** — choose one or more **Roles** (for example *Production* and *Recipe*; the list stays open while you tick several), then how they receive it, **Send by In-app, Email** and/or **Text message (SMS)**, and a **Minimum Severity**. The chosen roles share these settings. A group reaches the people IntelliTrack recorded in that role when they last signed in at this site, through their own profile: its primary email address and its primary phone number (once that number has agreed to texts), on the channels its **Channel Permissions** allow. Someone who has not signed in since this release, or has no profile, is not reached. A member who may not view the publication's type is skipped. As on a subscription, **Email** and **Text message (SMS)** can be chosen only when the site has a working channel for them; greyed out, they leave each role's own setting as it was, and a newly chosen role is not sent that way.
- **Teams and Slack** — choose one or more of the site's Teams and Slack **Channels** and a **Minimum Severity** they share. Every message of the publication at or above that severity is posted to each chosen channel once, whoever subscribes to the publication. Only channels that are switched on with a webhook URL stored are offered; a channel already chosen that has since been switched off, or cannot send, stays chosen, marked **switched off** or **not set up on this site**, until you clear it.

To send different publications to different Teams or Slack channels, choose the channels on each publication. To split one kind of event between channels (say, holds on Line 1 to one channel and on Line 2 to another), make one publication per line, each with its own plant scope and channels.

Clear a role or channel to take it off. If the roles (or channels) chosen before had different settings, the page says so, and saving applies the settings shown to all of them. Every change is recorded, role by role and channel by channel, with who made it, and appears in the audit trail.

Check your notifications (Inbox)

Notifications sent to you **In-app** go to your **Inbox**. The **Inbox** button (a bell) under your name, at the right of the menu bar, opens it from any page. A red bubble on its corner shows how many you have not read yet: 1 to 9, and **9+** for more than nine (a screen reader announces the exact number); with nothing unread there is no bubble. The number is taken as each page opens. The Inbox is also the first tile of **Configuration → Notifications**.

The Inbox lists your notifications, newest first: when each was received, its **Severity**, its title, the publication it came from, and its text. The ones you have not read are in bold and marked **New**.

1. **All** lists every notification; **Unread** lists only the ones you have not read, with how many there are.
2. Select a notification's title to open it: its own page, the same one the link in an email or text opens. Opening it marks it read.
3. **Mark as read** marks one read without opening it; **Mark all as read** marks every unread notification read.

The list is paged by your **Items per Page** preference ([Getting Started](#)). Only you see your Inbox: no one, an administrator included, can open another person's. A notification stays as long as its send stays in the **Delivery Log** (a year before the current month, unless your IT administrator changes it).

To receive notifications here, keep **In-app** ticked on your subscriptions (a new subscription starts with it ticked) and **Allow In-App Notifications** on your profile. When your profile turns in-app notifications off, the Inbox says so and links to your profile: what is already there stays, but nothing new is added. When you have no notification profile yet, the Inbox says so and links to **Subscriptions**, which takes you through creating one.

Review what was sent

Configuration → Notifications → Delivery Log (administrators) lists every send of this site — notifications, test messages and text-message codes — over the last day, 7, 30, 90 or 366 days, filtered by channel, status, type and text. For each: when, the publication and message (which opens the notification), the channel, the recipient, the status (**Queued**, **Sending**, **Sent**, **Delivered**, **Not delivered by the carrier**, **Not sent**, **Held back**), the tries, and why it was held back or not sent. Above the list are the totals by status, the sends held back by reason (hourly cap reached, within the subscriber's quiet window, below the channel's severity, number has not agreed to texts, channel missing or switched off), and the repeats counted within a publication's cooldown. A text that went as MMS shows **MMS** as its channel. (Older rows may show **Too long for a text**: summaries were held back from texts before they went as MMS.)

Twilio accepting a text is not the carrier delivering it. For 15 minutes after a text is sent, IntelliTrack asks Twilio what became of it, and **Sent** becomes **Delivered**, or **Not delivered by the carrier** with Twilio's error code and what it means — for example *30034: the sender is not registered for US A2P 10DLC*. A text Twilio has no answer for by then stays **Sent**. A text the carrier did not deliver is not sent again, since it

would be refused again: fix what the reason names. Choose **Not delivered by the carrier** under status to list them.

The log keeps a year before the current month unless your IT administrator changes it.

When someone says "I didn't get the alert", check, in order:

1. The **Delivery Log**: is there a send to them, and what happened to it?
2. The publication **Is Active**, and its settings and plant scope include what happened; its type reaches their roles.
3. The subscription (or role group) is **Enabled**; its **Minimum Severity** is not above the message's **Severity**; its **Suppression Window Minutes** is not holding back a repeat.
4. The channel is ticked on the subscription, allowed under the profile's **Channel Permissions**, and has an address chosen; for text, the number **Receives texts**. An **In-app** send is in their **Inbox**, under **Unread** until they open it or mark it read.
5. The site's channel is switched on, its **Minimum Severity** lets the message through, and its test succeeds. For email, the **Email Server (SMTP)** page also has its **Host** and **From Address** filled in and no warning under **Password** ([System Configuration](#)).

To be told when delivery itself fails, subscribe an administrator to **Notification Delivery Failure Detected** — on a channel other than the one that is failing. It carries the reason the send was given up on; for email that is the mail server's own answer — a refused sign-in, a sender the email service has not verified, or no answer before the **Timeout (seconds)**.

Set up notification channels (Administrators)

Configuration → **Additional Configuration** → **Connections & Notifications** → **Notification Channels**

holds the site's channels. Each site has its own; a server not yet bound to a site has none.

- **Link address** — the address people use to open IntelliTrack at this site, such as `https://intellitrack.example.com`. Emails, texts and Teams and Slack posts link to the notification there. Leave it empty to send them without a link.
- **E-mail** — sent through the site's own mail server, set on **Email Server (SMTP)** ([System Configuration](#)): your organization's server, or an email service's SMTP relay such as Twilio SendGrid or Brevo. The card shows the server, its TLS mode and the from address. Add the e-mail channel to switch it on.
- **SMS** — sent through the site's **own Twilio account**, from a number Twilio has approved for texting US phones. Enter the **Account SID**, how it signs in — an **API key** (recommended: its **API key SID** and secret) or the account's **auth token** — and the **Messaging Service SID** (or a number to send from). A text longer than one SMS goes as one MMS, so the sending number must be able to send MMS. Send a **Long text (MMS)** test to check it. Twilio bills an MMS at its MMS rate, which is higher than one SMS. See [Set up the SMS channel with Twilio](#).

- **Microsoft Teams** and **Slack** — add one channel per Teams or Slack channel, with its **incoming webhook URL**; choose which publications post to it on each publication's Audience. Teams or Slack makes the URL, not IntelliTrack; see [Get a Teams or Slack webhook URL](#).

Each channel has a **Minimum Severity** (messages below it do not go through it; SMS starts at **Alert**) and two **hourly caps** — per recipient and in all — beyond which sends are recorded as held back. A new channel has **no caps**: nothing is held back until you enter one. Set them to stop a flood — a machine alarming every few seconds, say — from filling phones and inboxes; for example 6 texts per person and 200 in all per hour. Leave a cap empty for no cap. Clear **Enabled** to switch a channel off without losing it.

Send a test on a channel's page sends one message — to the email address or number you enter (a number that has agreed to texts), or to the Teams or Slack channel — through the same queue as a notification, even while the channel is switched off. A test goes out ahead of every notification waiting to be sent, as does the code that confirms a phone, so a backlog never keeps you waiting for it. On an SMS channel choose **Short text (SMS)** or **Long text (MMS)**: the long test is longer than one SMS, so it goes as one MMS, as a long notification would, and only to a +1 or +61 number. Its result appears on the page and in the Delivery Log. For a text, the page waits for the carrier's answer: **Delivered**, or **Not delivered by the carrier** with the reason. A text Twilio accepted and the carrier refused is how a 10DLC registration that is not yet approved shows.

Numbers that replied STOP are listed at the top of the page until you **Acknowledge** each one. The number stays stopped until its owner agrees again.

Set up the SMS channel with Twilio

Why this matters. IntelliTrack texts through your company's own Twilio account. Twilio bills you directly, and the texts come from your number. IntelliTrack only sends. It asks Twilio what the carrier did with each text, and it learns of a **STOP** reply when Twilio refuses the next text to that number. So Twilio never needs to reach your server, and you change nothing under the number's **Inbound messaging** in Twilio. US carriers deliver a business's texts only from a number registered to that business, so Twilio must approve the number before any text arrives.

1. **Get a number and have it approved.** In the Twilio Console, buy a number that lists **SMS** and **MMS**, then register it. Twilio reviews the registration, which can take days or weeks:

- **A toll-free number** (800, 833, 844, 855, 866, 877 or 888) needs **Toll-Free Verification**. The number's **Compliance registration** tab shows its status. This is the simpler choice: no Messaging Service is needed.
- **A local number** (10DLC) needs an **A2P 10DLC Brand and Campaign**, with the number in the campaign's **Messaging Service**.

Register under the same business name as **Company Name** on Branding (see [Agree to receive text messages](#)). When the registration asks how people opt in, the answer is: on IntelliTrack's **Phone Numbers** page, by entering a code texted to the phone.

2. **Create an API key.** In the Twilio Console, open **API keys & tokens** and create a **Standard** key named for the site. Copy its **SID** (it begins with `SK`) and its **Secret**: Twilio shows the secret only once. A key is better than the account's auth token because you can revoke it on its own. IntelliTrack then stops sending, and nothing else that uses the account is affected. Changing the auth token breaks everything that uses it.
3. **Add the SMS channel.** On **Notification Channels**, add an SMS channel and enter:
 - **Account SID**: the `AC...` value on the Twilio Console's account details. Do not use the **Test Account SID** that the **API keys & tokens** page lists under test credentials. It is also `AC...` , but nothing sent with it reaches a phone, and your number and API key do not belong to it.
 - **Signs in with: API key**, with the key's SID and secret.
 - **Messaging Service SID** (`MG...`) or **Or send from this number**. For a toll-free number, enter the number itself in international form, such as `+18005550100` .

Save the channel. It starts at **Minimum Severity Alert**. Set its hourly caps as described above.

4. **Agree to texts on your own phone, then test.** A test goes only to a number that has agreed. On **Configuration** → **Notifications** → **User Profile** → **Phone Numbers**, select **Send a code** and enter the code. That code is the first text through the new channel, and it goes only while the channel is **Enabled**. Then **Send a test** on the channel, once as **Short text (SMS)** and once as **Long text (MMS)**. A test to a number that has not agreed is refused before anything is sent, with *That number has not agreed to receive texts*.

If Twilio accepts a text but it is **Not delivered by the carrier**, the reason usually says the number is not approved yet. Twilio error 30032 means a toll-free number is not verified, and 30034 means a local number is not registered for A2P 10DLC.

Get a Teams or Slack webhook URL

Why this matters. IntelliTrack never signs in to Teams or Slack, and it cannot see your teams, workspaces or channels. It posts each message to an **incoming webhook URL**: a private address that Teams or Slack creates for one channel, and that accepts messages from outside. The person who creates the URL chooses the channel, and posts appear there under the name of the workflow or app that owns the URL. So someone with rights in Teams or Slack must create the URL first. You then paste it into the channel's **Incoming webhook URL** field.

The link you copy from a channel is not a webhook URL. A channel's **Copy link** in Slack

(`https://<workspace>.slack.com/archives/...`) or in Teams

(`https://teams.microsoft.com/l/channel/...`) opens the channel in a browser, and it does not accept messages. IntelliTrack accepts any `https` address when you save, so it cannot tell a channel link from a webhook URL until it posts. After saving, **Send a test**. If the test is not sent, or nothing appears in the channel, check the URL first.

Microsoft Teams. A Teams *workflow* (Power Automate) owns the URL:

1. In Teams, point at the channel and select **More options (...)** → **Workflows**.
2. Search for and choose the **Send webhook alerts to a channel** template.
3. Name the workflow, for example *IntelliTrack – Line 1*. Sign in if Teams asks: the workflow posts through this account's connection.
4. Check the **Team** and **Channel**, then add (save) the workflow.
5. Copy the URL Teams shows. It is a Power Automate address, on a `logic.azure.com` or `api.powerplatform.com` host.
6. Paste it into **Incoming webhook URL** on a **Microsoft Teams** channel in IntelliTrack, save, and **Send a test**. The post is a card with the severity and title, the message, and an **Open in IntelliTrack** button when **Link address** is set.

Things to know about a Teams workflow:

- **It belongs to the person who created it, not to the channel.** If that account is disabled, for example when the person leaves, the workflow stops and IntelliTrack's posts to it fail. Create it with an account that will last, or add a co-owner to it in Power Automate.
- Only a member of the team can create the workflow. Guest accounts cannot. The template needs no premium Power Automate license.
- Microsoft retired the old **Incoming Webhook** connector in May 2026. Addresses on `webhook.office.com` no longer accept messages. A channel that still uses one needs a workflow URL made as above.

Slack. A Slack *app* owns the URL:

1. Go to api.slack.com/apps, sign in, and select **Create New App** → **From scratch**. Name the app, for example *IntelliTrack*, and choose your workspace.
2. Under **Features**, open **Incoming Webhooks** and turn **Activate Incoming Webhooks** on.
3. Select **Add New Webhook to Workspace**, choose the channel, and select **Allow**.
4. Copy the **Webhook URL**. It begins with `https://hooks.slack.com/services/`.
5. Paste it into **Incoming webhook URL** on a **Slack** channel in IntelliTrack, save, and **Send a test**. The post is one message: the severity and title in bold, the message, and an **Open in IntelliTrack** link when **Link address** is set.

Things to know about a Slack app:

- **One URL posts to one channel.** To post to another Slack channel, add another webhook to the same app (step 3). Then add another IntelliTrack channel with the new URL.
- If your workspace requires approval for apps, **Allow** reads **Request to Install**, and a Slack workspace administrator must approve the app first.
- Posts appear under the app's name and icon. Set them under **Basic Information** → **Display Information**.

- Removing the webhook or the app in Slack stops the URL at once. Do this if the URL leaks. IntelliTrack's posts to the removed URL are then refused, and the Delivery Log shows the answer, for example *The Slack webhook answered 404 Not Found*.

How the channel secrets are protected

Why this matters. A Twilio API key secret or auth token, or a Teams or Slack webhook URL, is enough on its own to send messages *as your plant*. IntelliTrack stores them in the Reporting database, which reporting users, backups and restored copies can all read, so it stores them **only encrypted with this server's encryption keys**. A secret is **write-only**: once saved it is never shown again, only its last four characters and when it was set. Leave the field empty to keep it; enter a new one to replace it. Every change to a channel is recorded with who made it, and appears in the audit trail.

If a channel stops with "cannot be decrypted on this server". The secret was encrypted with keys this server does not have — typically because the database was restored onto a different server, or the server's key folder was deleted or moved. Its sends are tried again and then recorded as not sent, with that reason. Enter the secret again on the channel's page. This is deliberate — it is what stops a copied database from being used to send messages as your plant.

If saving is refused because the server is not saving its encryption keys to disk, a secret stored then would be lost the next time the application restarts, so IntelliTrack refuses to store it. The message names the folder; your IT administrator must allow the application to write to it and restart the application before you try again.

Note: The older **Notification Configuration** page, which imported Twilio and SendGrid credentials supplied by Factory Data Systems, has been removed. If your site used it, set up the channels here with your own accounts.

Field Reference

User profile / addresses:

Field	Required?	What it does / valid values
Username	Yes	Whose profile it is — your IntelliTrack user name (administrators may choose). Up to 200 characters.
Time Zone	Yes	Recorded with the profile (UTC unless changed); message times are not converted to it in this release.
Allow In-App / Email / SMS Notifications	No	The channels this person accepts. A channel not allowed here is never delivered.
Email Address	—	An address for email; optional Label, Sort Order, Primary Address .

Field	Required?	What it does / valid values
Phone Number	—	With its country code (stored as +15550102030); texted only once it agrees; optional Label , Sort Order , Primary Number .

Publication:

Field	Required?	What it does / valid values
Publication Type Code	Yes	The kind of notification; fixed once saved.
Name / Description	—	How the publication appears in the list.
Is Private	No	Visible to, and subscribable by, only its creator and administrators.
Is Active	No	Whether it is evaluated and sent.
Publication Settings	Per type	Thresholds, filters, Severity , Cooldown Minutes , Title Template / Body Template , and usually a plant scope.

Subscription:

Field	Required?	What it does / valid values
Subscriber	Yes	You; administrators may choose any profile.
Publication	Yes	One the subscriber may view.
Minimum Severity	—	Information, Warning or Alert; messages below it are not sent.
Suppression Window Minutes	—	0 to 525,600; skip repeats from this publication for this long after a delivery.
Enabled	—	Whether the subscription delivers.
In-app / Email / Text message (SMS)	—	The channels to use (each must also be allowed on the profile). In-app puts the notification in the subscriber's Inbox . Email and Text message can be chosen only when the site has a working channel for them.
Delivery Addresses	When Email or Text message is ticked	Which of the profile's email addresses and phone numbers receive it. A ticked Email needs at least one email address, a ticked Text message at least one number that can receive texts; the primary is chosen when a channel is ticked, and a channel that is cleared keeps none.

Notification channel:

Field	Required?	What it does / valid values
Name	Yes	Up to 100 characters, unique on the site.

Field	Required?	What it does / valid values
Enabled	—	Whether notifications go through it.
Minimum Severity	—	Information, Warning or Alert.
Per recipient per hour / In all per hour	Yes	1 to 100,000.
Account SID	SMS	AC and 32 hexadecimal characters.
Signs in with	SMS	An API key (its SID, SK..., and secret) or the account's auth token.
Messaging Service SID / Or send from this number	SMS	MG..., or a number in international form; the Messaging Service is used when both are given.
Secret	SMS, Teams, Slack	The API key secret or auth token, or the incoming webhook URL (https). Write-only.

Related Topics

- [System Configuration](#) — **Email Server (SMTP)**, the mail server notification email goes through; synchronization jobs
- [AI Insights](#) — the AI page's setup checklist includes the notification channels
- [OOE](#) — alarm and downtime publications
- [Quarantine & Disposition](#) — confirmed quarantines raise **Hold Placed**
- [Electronic Signatures & Audit Trail](#) — the signature alerts

Chapter 16: ERP Integration

Overview

IntelliTrack can exchange transactions with an external ERP system. In this release the ERP it can talk to is **Microsoft Dynamics 365 Business Central**: SAP and NetSuite appear in the **ERP System** list, but no adapter exists for them yet, so a connection to either is never used — its connection test says so, and anything a site sends through it is dead-lettered. You configure a **connection** to the ERP, define **field mappings** that translate values between the two systems, choose which **transactions** are enabled per site, and use **diagnostics** to monitor the message flow and re-process failures. A background worker does the actual exchange; these screens configure it and show its activity.

Four transactions exist:

Transaction	Direction	What it carries
WorkOrderImport	ERP → IntelliTrack	The ERP's production orders, created or updated as IntelliTrack work orders (Chapter 7).
ProductionConfirmation	IntelliTrack → ERP	Each unit's good or scrap output at each machine, against the work order it was built under.
MaterialConsumption	IntelliTrack → ERP	Each component consumed into an assembly, against the assembly's work order.
OrderStatus	IntelliTrack → ERP	Each release of a work order (as <i>Started</i>) and each completion (as <i>Complete</i>).

Why this matters: the ERP and the plant floor have to agree without anyone re-keying data — a work order created in the ERP should appear on the line, and completed production should flow back. This integration is what removes that manual hand-off and the errors it causes. The configuration here is the contract that makes the two systems speak the same language; the diagnostics screen is how you prove they actually did.

Before You Begin

- The entire ERP area requires the **Administrator** role.
- Message processing is done by a background worker that runs inside the IntelliTrack web application — one loop sends, one imports, one retries. It does nothing for a site until one of its transactions is enabled *and* the site has an active connection, and nothing at all on an installation without an ERP integration database. The web screens configure it and surface its ledger.
- What IntelliTrack sends is read from the **Reporting** database, so an event reaches the ERP after it has been copied there (see the synchronization jobs in [System Configuration](#)).

- The integration numbers its sites itself: **Site** on every ERP page is a whole number (1 or more) of your choosing, not a plant-model site. Use the same number on a site's connection, its field mappings and its transaction rows. Mappings and transactions are configured **per site and transaction type**, so have your site numbers and the agreed transaction set ready.
- For Business Central you need an app registration in your Entra ID tenant (its client id and secret, for the OAuth client-credentials flow) and, in Business Central, the extension that publishes the production endpoints IntelliTrack posts to — `productionOutputEntries`, `productionConsumptionEntries` and `productionOrderStatusEntries` — alongside `productionOrders`, which the work order import reads. Business Central's standard API has no such endpoints; the connection test checks each one.

How to Get Here

The ERP pages have no menu entry. An administrator opens them by typing IntelliTrack's address followed by `/Erp/Connections` (**ERP Connections**), `/Erp/FieldMappings` (**Field Mapping**), `/Erp/Transactions` (**Transaction Enablement**) or `/Erp/Diagnostics` (**Diagnostics**).

How the integration flows

IntelliTrack ⇌ worker ⇌ ledger ⇌ ERP



Field mappings translate each field/value; transaction enablement decides which transaction types flow, per site. Only an active connection is used.

A failed or dead-lettered ledger entry can be retried once the underlying problem is fixed.

Configure an ERP connection

A connection tells the worker where a site's ERP is and how to sign in to it. Nothing flows for a site until it has one that is **Active**.

1. Open **ERP Connections**. The list shows each connection's **Site**, **ERP System**, **Display Name**, **Base URL** and whether it is **Active**.
2. Select **Add Connection**, fill in the fields below, and **Save**.
3. Select **Test Connection** on its row (see below) before you rely on it.
4. **Edit** or **Delete** it as needed.

Field	What it is
Site	The integration's number for this site. Its field mappings and transaction rows use the same number.
Plant Site (GUID)	The IntelliTrack plant-model site whose production this connection reports — the identifier at the end of that site's Details page address in the equipment model (Chapter 2). Leave it blank to report production from every site in the Reporting database.
ERP System	BusinessCentral — the only one implemented. Sap and NetSuite can be saved but are never used.
Display Name	Your name for the connection.
Active	Only an active connection is used, and a site can have only one — so you can stage a new connection, test it, and switch it on when ready.
Base URL	The root of the Business Central API for your tenant and environment; the worker adds the company and entity path (<code>companies(...)/productionOrders</code> and so on) to it.
Token Endpoint	Your Entra ID tenant's OAuth 2.0 token endpoint.
Tenant Id, Environment	Kept with the connection for reference; Business Central is reached through the Base URL and Token Endpoint , which carry them.
Company Id	The Business Central company the transactions belong to.
Client Id, Client Secret	The app registration's credentials. The secret is stored encrypted and is never shown again; when editing, leave it blank to keep the current one.
OAuth Scope	Leave blank to use Business Central's default scope (<code>https://api.businesscentral.dynamics.com/.default</code>).

A site can hold one connection per ERP system. Changing a connection takes effect on the worker's next pass; there is nothing to restart.

Test a connection

Test Connection checks a saved connection — active or not — by walking the same path a real transaction takes, without writing anything: it signs in and then only reads, so it is safe to run against a live ERP. The result page shows **The connection test passed.** or **The connection test failed.** **Review the steps below.**, then each step with its **Outcome**, **HTTP** status, **Elapsed (ms)**, **Endpoint** and **Detail**:

Step	What it proves
Configuration	The connection has everything a call needs (Base URL , Company Id , Token Endpoint , Client Id and a stored secret); otherwise it names what is missing, and the later steps are skipped.

Step	What it proves
Authentication (OAuth token)	Entra ID issues a fresh token for the client id and secret — a revoked or mistyped secret fails here.
Read probe (work orders)	The production-orders endpoint the work order import reads can be reached and read.
Endpoint discovery (outbound)	Each of the three endpoints IntelliTrack posts to exists. A Warning means the endpoint exists but refuses reading, so posting to it could not be proved without writing to the ERP. A Failed step shows what Business Central answered; a 404 means the extension that publishes that endpoint is missing, and the transaction that posts there will fail.

Warnings do not fail the test. For **Sap** or **NetSuite** the page says that no adapter is registered, so the connection cannot be tested.

Map values between the systems (Field Mapping)

The two systems rarely name things the same way: IntelliTrack's unit *Each* may be *PCS* in the ERP, and a material's name in IntelliTrack may not be the ERP's item number. A **Field Mapping** row translates one IntelliTrack value into the ERP's value, for one site and one transaction — the step that lets the ERP accept what IntelliTrack sends without anyone renaming anything in either system.

1. Open **Field Mapping** and select **Add Mapping**.
2. Choose the **Site** and the **Transaction** the row applies to and its **Kind** — **UnitOfMeasure**, **Item** or **Field** — then enter the **IntelliTrack Value** and the **ERP Value**, and **Save**.
3. **Edit** or **Delete** rows as the integration evolves.

What mappings do:

- **Only differences need a row.** A value with no mapping is sent unchanged. Matching ignores upper and lower case. There can be one row per site, transaction, kind and IntelliTrack value.
- **A row applies to one transaction.** With Business Central, **UnitOfMeasure** rows translate the unit sent with **ProductionConfirmation** and **MaterialConsumption**, and **Item** rows translate the component's item number sent with **MaterialConsumption** — so a unit both transactions send needs a row under each. **Field** rows are not used by the Business Central adapter in this release.
- **Mappings apply only to what IntelliTrack sends.** The work order import does not use them: an ERP order's item number and unit must match IntelliTrack's own names (see *What the work order import does* below).
- **Mappings are read each time a message is sent**, so correcting one also corrects the next **Retry** of a message that failed because of it. If the ERP rejects a message because it does not recognize a unit or an item, add or correct the mapping, then retry the message (see *Monitor and retry* below).

Enable transactions

Transaction Enablement decides which transactions actually flow for each site, and how fast — your master switch.

1. Open **Transaction Enablement** and select **Add Transaction**.
2. Choose the **Site** and the **Transaction**, tick **Enabled** (a row saved without it is switched off), and set:
 - **Cadence (sec)** — how often the worker runs this transaction for this site (120 by default).
 - **Batch Size** — how many IntelliTrack events one pass reads for sending (100 by default; the work order import takes whatever the ERP returns).
 - **Max Attempts** — how many times a message that keeps failing with a temporary error is tried before it is dead-lettered (10 by default).
3. **Save**. Each site can have one row per transaction; a second is refused with *A row already exists for this site and transaction type*.
4. **Edit** a row to change those settings (its site and transaction are fixed), clear **Enabled** to pause the transaction and keep its settings, or **Delete** the row.

The worker processes only rows with **Enabled** ticked, and only for a site with an active connection; while a site has no active connection its enabled transactions simply wait.

Note: The worker remembers how far it has read for each site and transaction (the **Watermarks** on **Diagnostics**). The **first** time a transaction runs for a site there is no mark yet, so it starts from the beginning: an outbound transaction works through every qualifying event the Reporting database still holds, one batch per pass, and **WorkOrderImport** asks the ERP for every order it has. After that only new events are read. Pausing, or deleting and re-adding the row, keeps the mark: when the transaction runs again it resumes where it stopped and sends everything that happened in between.

What the work order import does

When **WorkOrderImport** is enabled for a site, the worker asks the ERP, once per cadence, for the production orders changed since its last import, and creates or updates the matching IntelliTrack work orders ([Work Orders](#)). This is what puts the ERP's plan on the floor without anyone retyping it: the ERP's order number becomes the **Work Order Number**, its item number the **Material**, its quantity the **Planned Quantity**, its unit the **Unit of Measure**, and its due date **Required By**.

- **New orders arrive in Draft**, whatever their status in the ERP, with an **Expiry Date** from parameter 13. Releasing them to the floor is still done in IntelliTrack — the import never changes an order's status.
- **Updates follow the ERP's plan**. An existing order that is not Canceled or Closed takes the ERP's material (only until production has started against it), unit, planned quantity and due date. The import also writes **Description**, **Group** and **Planned Start**, which Business Central does not supply, so edits made to those three in IntelliTrack are cleared the next time the ERP sends that order.

Customer, Planned Resource, Overbuild Tolerance (%) and **Expiry Date** stay as IntelliTrack has them. Canceled and Closed orders are never changed.

- **The material must already exist in IntelliTrack.** The ERP's item number must equal the name of an IntelliTrack material with a revision enabled for production (the newest such revision is used), and its unit must equal a unit's name or abbreviation — field mappings are not applied on the way in. An order whose material or unit is not found is skipped and recorded in **System Errors & Warnings** (*Work order "... " was not imported: material "... " was not found or is not enabled for production.*). The import then re-reads from the same point on every pass, so the order arrives on the first pass after the material exists (or the ERP is corrected); until then the warning is recorded again on each pass.
- An order number that appears twice in one fetch is imported once, and a warning is recorded.
- Every create and update is recorded in the order's **Lifecycle History** as **Create** or **Modify** by **erp:import**. An update that changes nothing is not recorded.
- **The import does not ask for signatures.** It does not consult the signature policies, so a planned-quantity change arriving from the ERP is applied without a signature even where your site signs such a change made on the web (Chapter 7).
- Business Central's dates are read in a fixed format, independent of the server's regional settings, so a due date arrives as the ERP sent it.

What a production confirmation reports

When **ProductionConfirmation** is enabled, each unit's result at each equipment module is sent to the ERP as good or scrap quantity against a work order, with the machine's name as the operation, the unit's quantity and unit of measure, and the time the unit finished there as the posting date. A result that is neither good nor scrap (still in process) is not sent. A connection with a **Plant Site** sends only production at machines in that site.

- **The order is the one the unit was worked under.** IntelliTrack records the work order on each processing event when a machine adds the part or the part is checked out (at a machine or on **Production → Check-Out**), and the confirmation posts against that order. Moving a unit to a different order afterwards does not move production already recorded. Work done under the first order stays with it, and later work goes to the new one.
- **Work done with no work order is not sent.** A unit processed before it belonged to any order has nothing to post against, and assigning it to an order later does not send that earlier production.
- **One confirmation per unit, per equipment, per order.** A unit that runs through the same equipment more than once under one order is reported once, with its last result. A unit reworked at that equipment under a different order is reported to each order separately.
- **Refused check-outs and changes made on the web are not production.** A check-out the gate refused is not reported. Neither are receiving a product, edits on the Products pages, **Set as Reject**, deleting a product, or scrapping a product from a quarantine.

Why this matters: an order's output in the ERP has to be what was actually built against that order, so moving a unit to another order never moves production that has already been recorded. The order's own Started and Completed counts can move with the unit, depending on how it is moved (Chapter 7). If a unit was on the wrong order when it was built and its production has already been sent, correct that in the ERP. IntelliTrack does not re-post it.

What material consumption reports

When **MaterialConsumption** is enabled, each genealogy link that records a component consumed into an assembly (see [Products](#)) is sent to the ERP as a consumption against a work order: the component's material as the item number (translated by **Item** mappings), the quantity consumed and its unit (translated by **UnitOfMeasure** mappings), the component's serial number as the lot number, and the machine where the link was made as the operation.

- **The order is the assembly's, at the moment of consumption** — the work order stamped on the processing event the assembly had open when the component went in. It is the same stamp production confirmations use, so a unit's output and the material that went into it are reported against the same order, and moving the assembly to another order later moves neither.
- **Consumption with no order in force is not sent.** If the assembly was not in a processing cycle at that moment (as with most links made on the web), or that cycle carried no work order, there is nothing to post against.
- The link that records a unit's own creation is not a consumption and is never sent.
- Removing a genealogy link after its consumption was sent does not reverse it in the ERP.
- A connection with a **Plant Site** sends only links made at machines in that site.

What order status reports

When **OrderStatus** is enabled, the ERP is told when an order starts and finishes, so its own planning sees the order's real state without anyone reporting it by hand. Each **Release** of a work order is sent as *Started*, and each completion as *Complete* — whether someone selected **Complete**, **Block** mode completed the order at its plan, or a machine's **Close** completed a running order. Other changes (Schedule, Return to Draft, Start, Stop, Reopen, Close, Cancel) are not sent. That includes the automatic start when the first product is built against a released order: the release already told the ERP the order had started. Releasing or completing the same order again sends it again.

Each message carries:

Item	What is sent
Order	The Work Order Number — the ERP's production order number.
Status	<i>Started</i> or <i>Complete</i> .

Item	What is sent
Operation	The name of the machine that made the change; empty for a change made on the web, through the work-order API, or automatically by IntelliTrack.
Effective date	When the change was recorded in IntelliTrack.
External document number	The same on every send of the same change, so the ERP can recognize a re-send as a duplicate (see <i>Monitor and retry</i> below).

What the ERP does with it is up to the ERP. IntelliTrack reports the change to the Business Central extension's `productionOrderStatusEntries` endpoint; it does not set the order's status in Business Central itself. Whether a *Complete* moves the Business Central production order to *Finished*, or is only recorded for planners, depends on how that extension is built. Confirm this with whoever maintains the extension before you rely on it.

Note: A connection with a **Plant Site** sends only the status changes recorded at a machine in that site. A release or completion made on the web, through the work-order API, or automatically by IntelliTrack has no machine, so it is sent only by a connection whose **Plant Site** is blank.

Tip: An ERP or scheduling system that needs more than *Started* and *Complete* — every status, or live Started, Completed and Rejected quantities — can ask for it through the work-order API instead: `GET api/v1/work-orders/{number}/summary` returns an order's current status and counts, and `GET api/v1/work-orders?status=Completed` lists the orders in a status (by its name as shown in IntelliTrack). This does not require the ERP integration to be set up, only an API key (see *Bulk scheduling and the work-order API* in [Work Orders](#)).

Monitor and retry (Diagnostics)

Diagnostics shows the ledger of everything IntelliTrack has sent or tried to send, so you can prove the two systems agree and pinpoint where they didn't. Each IntelliTrack event gets one ledger entry per ERP system and transaction, and an entry marked **Sent** is never sent again.

Ledger Summary counts the entries in each status:

Status	Meaning
Pending	Recorded and waiting for its first delivery; the worker sends it on its next pass.
Sent	Delivered. Final — it is never sent again.
Failed	A delivery attempt failed with a temporary error (a timeout, or the ERP busy or unavailable). The worker retries it by itself, waiting longer each time — from about 30 seconds, doubling, up to an hour — and when a site's ERP fails three times running it sets that site's remaining retries aside until the next retry pass.

Status	Meaning
DeadLettered	Stopped. Either the failure was permanent — the ERP rejected the message, sign-in failed, the site has no active connection or no adapter for its ERP, or the message lacks something it must carry — or the transaction's Max Attempts ran out. Nothing more happens until someone selects Retry .

Watermarks shows, for each site, transaction and direction (`erp:1:ProductionConfirmation:out` , `erp:1:WorkOrderImport:in`), the timestamp of the last source record examined (**Last Seen**) and when the worker last completed a pass (**Last Run**). A **Last Run** that stops moving means that transaction is not being processed — it is not enabled, its site has no active connection, or its passes are failing. A **Last Seen** that stays put while **Last Run** moves means there is nothing new — or, for the work order import, that an order is being held back (see *What the work order import does*). A **Last Seen** in the year 1 means nothing has been examined yet.

Recent Failures lists the 100 most recently changed entries that are **Failed** or **DeadLettered**, with their **Site**, **Transaction**, **Source Id** (the IntelliTrack record the message came from), **Attempts** and **Last Error**.

To re-send an entry, fix the cause, then select **Retry** on its row. The entry goes back to **Failed** with its wait cleared, and the worker's next retry pass (about once a minute) sends it — *The transaction has been re-queued for delivery*. Its attempt count is not reset, so an entry that had used up its **Max Attempts** gets one more try before it is dead-lettered again.

Warning: Retry re-sends the message exactly as it was first recorded — it does not read IntelliTrack again. It helps when the cause lay outside the message: a mapping, the connection, or the ERP's own setup, all of which are read again when the message is sent. A message whose own content is wrong (a **Last Error** such as *UnitOfMeasure is required.*) fails the same way every time; correct it in the ERP instead. Re-sending does not post twice: every message carries the same external document number each time it is sent, and Business Central treats one it has already accepted as a duplicate, which the ledger records as **Sent**.

Field Reference

Screen	Key fields / actions
ERP Connections	Site , Plant Site (GUID) , ERP System , Display Name , Active , Base URL , Token Endpoint , Tenant Id , Environment , Company Id , Client Id , Client Secret (stored encrypted, never shown; blank on edit keeps it), OAuth Scope . Test Connection per row. One connection per site and ERP system; one active per site.
Field Mapping	Site , Transaction , Kind (UnitOfMeasure, Item, Field), IntelliTrack Value , ERP Value . Unmapped values are sent unchanged.

Screen	Key fields / actions
Transaction Enablement	Site, Transaction (WorkOrderImport, ProductionConfirmation, MaterialConsumption, OrderStatus), Enabled, Cadence (sec), Batch Size, Max Attempts . One row per site and transaction.
Diagnostics	Read-only Ledger Summary, Watermarks and Recent Failures; Retry per failed or dead-lettered entry.

Related Topics

- [Work Orders](#) — what imported orders become, releasing them, and the work-order API (a related integration point)
- [Products](#) — the genealogy links material consumption reports
- [Production Execution](#) — the check-outs whose results production confirmations report
- [System Configuration](#) — the Reporting database synchronization the outbound transactions read, API Clients, System Errors & Warnings

Chapter 17: System Configuration

Overview

This chapter gathers the administrative screens and background jobs that keep IntelliTrack running: connection and email setup, the three **database synchronization jobs**, **system parameters**, the site's **date and time format**, units of measure, inventory locations, approval statuses, **branding**, **API clients**, **licensing**, and the **system logs**. Most require the Administrator role to change, though the logs and several lists are viewable by any signed-in user. Who holds which role is decided on the **Security** pages ([Chapter 19](#)); electronic signatures and the audit trail have a chapter of their own ([Chapter 20](#)).

Why this matters: these are the settings with the **longest reach**. A single system parameter can change how the whole floor behaves; an API key grants a machine-to-machine door into your data; the connection settings underpin every other screen. Small changes here have large, system-wide effects — which is why they live behind the Administrator role, why what they save in IntelliTrack's databases is recorded in the **audit trail** (who changed what, when, from what to what), and why, on a site that runs with GxP Mode on, some of them ask for an **electronic signature**.

Before You Begin

- Unless noted, changing these requires the **Administrator** role. The exception is **inventory locations**: they are part of the plant model, so the **Engineering** role maintains them too ([Chapter 2](#)). Viewing parameters, units of measure, locations, approval statuses and the system logs — and the licensing **overview** — is available to **any signed-in user**, and so is the **Additional Configuration** page itself. **System Connections**, **Email Server (SMTP)**, **Branding**, **API Clients**, and the **Security** and **Compliance** pages are for Administrators only, even to look at.
- **Electronic signatures**. When your site runs with **GxP Mode** on, some changes in this chapter ask for your signature before they are saved: a change to one of the signature settings (parameters 111–116) is signed by default, and editing or deleting any other system parameter, or editing an approval status, is signed wherever your site's signature policy requires it. The form then shows an **Electronic signature** panel above its buttons — see [When you are asked to sign](#). With GxP Mode off, none of these changes is signed (turning GxP Mode itself on or off always is — [Chapter 20](#)).

How to Get Here

- Additional configuration hub: **Configuration** → **Additional Configuration**. It carries the **Items per Page** and **Date and Time Format** cards, then one tile for each group of setup pages — **Materials & Logistics** (**Locations**, **Units of Measure**, carriers, suppliers), **Quality & Change Control** (**Approval Statuses**, dispositions, ECO patterns), **Connections & Notifications** (**System Connections**, **Email Server (SMTP)**, notification channels), **Security & Administration** (**Security**,

System Parameters, Branding) and **Equipment & Processes** (the plant model, Chapter 2) — and the **Compliance** and **Validation Documentation** tiles. A group's tile opens a page with a tile for each of its pages. You see only the tiles for pages your role can open, and a group with none is not shown. On each of those lists, and on the plant model, administrators also have **Show deleted**, which lists deleted items with a **Restore** for each ([Restore a deleted item](#)).

- System parameters: **Configuration** → **Additional Configuration** → **Security & Administration** → **System Parameters**.
- API clients: **Configuration** → **API Clients**.
- Licensing: **Configuration** → **Licensing**.
- System logs: **Configuration** → **System Errors & Warnings**, which opens **Log Search**. The home page's **Diagnostics** tile opens a different, older page, **Error Logs** (see *System logs* below).
- Who holds which role, and the sign-in settings: **Configuration** → **Additional Configuration** → **Security & Administration** → **Security** — **Role mapping** is described in [Chapter 19](#).
- GxP Mode, signature policies and the audit trail: **Configuration** → **Additional Configuration** → **Compliance**, a page with a tile for each ([Chapter 20](#)).
- The validation Document Library, Coverage Register and Package Generator (Engineering and Administrators): **Configuration** → **Additional Configuration** → **Validation Documentation**, a page with a tile for each ([Chapter 21](#)).

System parameters — small switches, system-wide effects

System parameters are numbered settings that tune application behavior. Several are referenced throughout this guide because they change what the floor experiences:

Examples of what a system parameter can govern

90 Work Order Quantity Enforcement	Off / Warn (log & keep running) / Block (auto-complete & refuse over-build) — Chapter 7
91 Work Order Required	refuse production not tied to a work order — Chapter 7
13 Work Order Expiry Time	days a new work order stays valid; expired, empty orders are pruned — Chapter 7
40 No Route Approval	on (default): a process no route leads into skips the routing check — Chapter 8
27 Shipping Verification Delay	seconds a shipping code stays up; 0 skips the final-step ship check — Chapter 8

Any signed-in user can view a parameter; Administrators change it. Change a value only when you understand its effect — it applies to the whole site.

The work-order parameters (90–93) are site defaults: an equipment module can carry its own value (Equipment enforcement, Chapter 7), which wins there.

Each Site database keeps its own set of system parameters — they are **not** kept in the shared Configuration database. A change therefore applies to this site only, and at once: the next check-out, prune run or page that reads the parameter uses the new value, with no synchronization to wait for.

How the parameters are numbered

Parameters are numbered in bands of ten by what they govern, so the number tells you the area before you read the name:

Band	Governs	Parameters
1–9	System and identity	1 Site Id
10–19	Retention and pruning	10 Minimum Product Retention Time, 11 Retention Time, 12 Report History Retention Time, 13 Work Order Expiry Time, 14 Max Recipe Revisions, 15 Prune Row Limit, 16 Reporting Database Name, 17 Alarm Archive Retention Time
20–29	Production and check-out	22 Allow Recipe Change (0 as shipped: only a Production Supervisor or Administrator may change the recipe an equipment is running at check-out; 1: Production may too — see Who may change the recipe), 23 Count Edits as Reruns, 24 Error on Failed Check-Out, 25 Check Consumed Quantity, 26 Skip Units of Measure Verification, 27 Shipping Verification Delay (180 seconds as shipped; 0 also skips the shipping check after a final-step check-in)
30–39	Disposition and status defaults	30 Default Disposition Code for New Products, 31 Default Disposition Code for New Products at Final Step, 32 Completed Product Status, 33 Default Auto-Scrap Status, 34 Genealogy Product Status
40–49	Routing and genealogy	40 No Route Approval (on as shipped: a process that no route leads into is not route-checked), 41 Genealogy Rows, 42 Max Generations
50–59	Shipping and supplier	50 Use Default Supplier Id, 51 Length of Shipping Code, 52 Use Alpha Characters in Shipping Code
60–69	Recipe	60 Max Inactive Or Obsolete Recipe Revisions, 61 Max Active Recipe Revisions — explained, with 14, in Chapter 5
70–79	Audit	70 Default Audit Override Time — the step an audit is postponed by: minutes for a time override, units for a units override
80–89	Equipment and OEE status	80 Downtime Threshold, 81 Communication Outage Threshold
90–99	Work orders	90 Work Order Quantity Enforcement, 91 Work Order Required, 92 Bind Work Order To Equipment, 93 Work Order Recipe Enforcement — site defaults that one machine can override on the Equipment enforcement page (Chapter 7)
100–109	Materials and integration	100 Max Image Size, 101 Event Engine Host (not currently used by any page)
110–119	Electronic signatures and audit trail	110 GxP Mode, 111–116 the signature and audit-trail settings
130–139	Presentation	130 Display Date Format
140–149	Report views and curves	140 Report Panel Limit, 141 Report Layer Limit, 142 Report Decimation Threshold, 143 WebGL Panel Budget, 144 Percent Of Target Minimum, 145 Maximum 2D Curve Samples, 146 Maximum 3D Curve Samples, 147 Maximum Surface Samples

Band	Governs	Parameters
1000 and up	Parameters your site creates	—

A few are worth knowing about here:

- **17 Alarm Archive Retention Time** (days; default **0**). When the purge job removes an alarm record older than the Report History Retention Time (12) from the Reporting database, 0 lets it go, as IntelliTrack always has. Any other value first copies the record to an alarm archive and keeps the copy for that many days from the alarm's start — what a site with a records-retention obligation needs, at the cost of the space the copies take. (Alarms are covered in [Chapter 11](#).) The Product Report lists the machine alarms that were active while each unit was worked, reading the archive too, and keeps the unit itself for the **Minimum Product Retention Time** (10); a site that relies on those alarms should set 17 to at least 10 ([Reports](#)).
- **81 Communication Outage Threshold** (seconds; default **30**, allowed 1 to 86400). When a machine's heartbeats stop for longer than this, the gap is recorded as a communication outage and listed under **Communication Anomalies** on the equipment's **Details** page ([Chapter 2](#)). Set it longer than the machine's normal heartbeat interval, or every ordinary gap is logged as an outage.
- **110–116** switch electronic signatures on and tune them: GxP Mode itself, the failed-signature limits and alert, the inactivity lock, local-account signing, and the audit trail's per-save limit. [Chapter 20](#) lists them with their defaults.
- **130 Display Date Format** holds the site's date and time format, which is set on the **Date and Time Format** card — see *Date and time format* below.
- **140–147 are the report charts' own limits**, and every one has a shipped default that suits an ordinary workstation: how many panels one report draws (**140**, 7) and how many curves one panel overlays (**141**, 20) before it offers them side by side; the point count above which a plot is thinned (**142**, 5000 — the panel says when it has thinned one, and the data table and the export never are); how many live 3D views one page keeps (**143**, 8); the smallest target a percent-of-target view will divide by (**144**); and the most samples one 2D curve, 3D curve or surface submission may carry (**145**, no limit; **146**, 50,000; **147**, 262,144 — a 512 by 512 grid). The three curve caps refuse an oversized submission outright: a curve is never stored in part. See [Reading a chart](#).

View and change a parameter

1. Open **Configuration** → **Additional Configuration** → **Security & Administration** → **System Parameters**. Search by name or description; results are paged. Any signed-in user can open a parameter's **Details**; **Administrators** also see **Edit**, **Create New** and — on parameters your site created — **Delete**.
2. **Edit** a parameter and set its value: **Numeric/Boolean Value** for a number or an on/off switch (1 = on, 0 = off), **String Value** or **Timestamp** for the few that hold text or a date.

- Every numeric parameter carries a **Minimum Value** and **Maximum Value**: the range the processes that read it can actually take, fixed by IntelliTrack. A value outside them is refused with a message naming the limit, and a value that reached the database some other way is read as the nearest limit.
 - A parameter IntelliTrack depends on cannot be left blank — a blank would stop the processes that read it.
 - On a shipped parameter only the **value** is yours. Its name, description, units and limits are restated by IntelliTrack every time the Config → Site job runs (see *Database synchronization jobs* below), so change those only on parameters you created.
3. Three parameters are not edited here. **Edit** on **GxP Mode (110)** opens **Configuration** → **Additional Configuration** → **Compliance** → **GxP Mode**, where switching it either way is always signed ([Chapter 20](#)). **Site Id (1)** is set by marking the site **Is Current Site** in the equipment model ([Set the current site](#)), and **Display Date Format (130)** on the **Date and Time Format** card; neither offers **Edit**.
 4. **Create New** adds a parameter of your own, numbered from 1000. IntelliTrack itself reads only the parameters it ships, so create one only for your own scripts, reports or integrations to read. You choose its **Units** and its limits (they start at the widest possible range; clear both for a text or timestamp parameter), and everything you enter is kept.
 5. **Delete** removes a parameter your site created — the shipped ones are required and cannot be deleted. The confirmation page names the parameter, and exactly that parameter is removed.

Every create, edit and delete is recorded in the audit trail, with the settings before and after. With GxP Mode on, editing one of the signature settings (111–116) is signed by default and asks for a reason — an unsigned change to one of them would be a quiet way round the signatures it governs. Editing or deleting any other parameter is signed if your site's signature policy requires it; the signature covers the parameter's settings before and after, and if the signature is refused, nothing is changed. Creating a parameter is never signed. Once a change has been signed, the parameter's **Details** page shows an **Electronic signatures** section ([Seeing who signed a record](#)).

Warning: Parameters change behavior for everyone at the site. For example, turning **Work Order Required** on will start refusing any production not tied to a work order — across every station that does not carry its own module-level override — the moment you save.

Date and time format — the same dates for everyone

The **Date and Time Format** card is at the top of **Configuration** → **Additional Configuration**, beside **Items per Page**. Anyone can see the format in force, with today's date shown in it; only an **Administrator** can change it, and the card says so: *Changing the date and time format needs an IntelliTrack administrator.*

Why it exists: a plant that works to a documentation-practice standard — GMP good documentation practice, for instance — has to write dates one agreed way on every record, and usually in a form nobody

can misread. 04/03/2026 is 4 March in one country and 3 April in another; 04/Mar/2026 is the same day everywhere. This one setting makes IntelliTrack's screens and reports print dates the same way for every operator, on every computer. It is deliberately **site-wide, not personal**: the **Items per Page** card beside it is your own preference, this one is everyone's.

1. Open **Configuration** → **Additional Configuration**.
2. In the **Date and Time Format** card, choose a format. Each choice in the list shows the current date and time as it would appear.
3. Select **Set**. *The site date and time format has been changed.* confirms it, and every page from then on uses the new format.

Choice	20 September 2026, 3:04:05 PM appears as
Month first (United States) — the default	9/20/2026 3:04:05 PM
ISO 8601	2026-09-20 15:04:05
Day first, 24-hour (GMP)	20/Sep/2026 15:04:05
Day first, 12-hour (GMP)	20/Sep/2026 3:04:05 PM
Day first, numeric	20/09/2026 15:04:05
Day first, dotted	20.09.2026 15:04:05

What it changes: the dates and times on IntelliTrack's screens and reports, and on chart axes. Where a page shows a time without seconds, it takes the same choice and drops the seconds. The default is exactly what IntelliTrack showed before the setting existed, so an upgrade changes nothing until an administrator chooses.

What it does not change:

- **Date and time pickers** — the boxes where you choose a date from a calendar — are drawn by your web browser and keep your computer's own format. What you pick is stored and displayed correctly; only the picker itself looks different.
- **Signature and audit-trail records.** The **Electronic signatures** sections and the audit-trail pages always give a moment in full — year-month-day with its time zone offset, and the same moment in UTC (2026-09-20 15:04:05 -04:00, and 2026-09-20 19:04:05 UTC) — because a signed or audited record must say exactly when, whatever the display choice.
- **A few fixed-format pages.** Some pages still show dates as year-month-day whatever the choice — among them **Licensing**, the **Alarm Journal**, the calculation times on **Work In Process Status**, and the **Quarantine Aging**, **Shift Performance** and **Where-Used / Recall Impact** reports.
- **Notifications.** The emails and text messages IntelliTrack sends write their dates and times in the web server's own regional format (on a United States English server, *9/20/2026 3:04 PM*), whatever the choice here.

- **Language.** Month and day names are always English, whatever language you use IntelliTrack in, so a dd/MMM/yyyy standard reads *Sep* on every screen — never a translated or four-letter abbreviation that would break it.
- **What is stored.** The setting changes how dates are shown, never the data, so it can be changed back at any time. Files meant for other programs — the audit trail export and the CSV exports of the **OEE Event Log**, **Fault Detail Report** and **Reject Detail Report** — keep their dates in ISO 8601 so they read back without ambiguity.

The format is kept in **System Parameter 130 (Display Date Format)**, and each change is recorded in the audit trail. The System Parameters page does not offer it for editing: the card offers only checked formats, because on a validated system a mistyped date pattern is a compliance finding. If your support team has written a pattern that is not one of the choices directly into that parameter, the card lists it as **Custom (set directly in System Parameters)**.

Connection and email setup

Two tiles on **Configuration** → **Additional Configuration** → **Connections & Notifications** hold the connection and mail settings (Administrators only):

- **System Connections** — where IntelliTrack finds its three databases: the **Config Database**, **Site Database** and **Reporting Database** (server, database name, SQL sign-in or Windows authentication, encryption, timeout), with the **Synchronization Settings** for centralized reporting below them. Each database has its own test button (**Test Config Connection**, **Test Site Connection**, **Test Reporting connection**) that tries the connection there and then, without saving anything, and says how long it took. **Save Configuration** stores the settings on the web server (passwords protected for that machine; leave a password blank to keep the saved one), registers the site for centralized reporting (see *How sites register themselves* below), and can run a full synchronization straight afterwards — **Perform full Config sync after save** and **Perform full Reporting sync after save** each ask you to confirm, because on a large site they take a long time. The web application starts using changed connection details only after it is restarted (recycle its application pool), as the confirmation message says.
- **Email Server (SMTP)** — the mail server IntelliTrack sends *all* of its email through: email notifications to subscribers ([Chapter 15](#)) and the license-expiry warnings (see *Licensing*). Text messages do not use it; they go through the site's own Twilio account (the SMS channel, [Chapter 15](#)). It can be your organization's own mail server or the SMTP relay of an email service (see *Sending through an email service* below). Notification email goes through it once an administrator adds the site's e-mail channel on **Notification Channels** ([Chapter 15](#)). Until **Host** and **From Address** are filled in, no email is sent: each email notification fails with *The site has no SMTP server; set one on the Email Server (SMTP) page* (it is tried again, and given up on after its last try), and no license-expiry email goes out.

- **Host, Port, Encryption** — where the server is and how the connection is secured.
Encryption is **STARTTLS** (usually port **587**: the connection opens plain and is then switched to TLS), **TLS from the start** (usually port **465**, "implicit TLS", where TLS starts with the very first byte), or **None** (a relay inside the plant network only). Most servers and email services take port 587 with STARTTLS. Notification email works with any of the three; the license-expiry email cannot use TLS from the start and is not sent with it.
- **Username, Password** — the sign-in. The **Password** is stored encrypted and never shown again — not on this page, not in the audit trail (which records only that it changed, and who changed it), not in the log — and the note beneath the box says whether one is stored. When you change anything else, leave the box empty and the stored password is kept; type a new one to replace it; clear **Username** to remove it, since with no user name the web server signs in to the mail server with its own Windows account and no password is used. If the note says the password **cannot be decrypted on this server** (the database was restored from another server, or the server's encryption key folder was lost), type it again: until you do, no email at all is sent — each email notification fails with that explanation, and the license-expiry email is skipped. If it says the password is **not yet encrypted**, this server cannot save its encryption keys; fix the folder the note names and restart the application, and the password is encrypted automatically.
- **Domain** — only for a Windows mail server (such as Exchange) that signs users in as `DOMAIN\user`. Leave it empty for an email service: whatever is here is put in front of the user name, and the service refuses the sign-in.
- **Timeout (seconds)** — how long one email may take before it counts as failed. A server that accepts the connection and then goes quiet is given up on after this long, so it cannot hold up the notifications queued behind it.
- **From Address** — the sender your recipients see, for example `IntelliTrack <alerts@yourplant.com>`. An email service accepts only a sender address, or a domain, that you have verified with it; any other is refused, and every email fails.
- **License Expiry Recipients (CSV)** — who receives the license-expiry warnings, separated by commas. Email notifications do not use it: each goes to the addresses its subscriber chose ([Chapter 15](#)).

Sending through an email service. Any service with an SMTP relay works; nothing in IntelliTrack is specific to one. Two common ones:

Service	Host	Port	Username	Password
Twilio SendGrid	<code>smtp.sendgrid.net</code>	587	<code>apikey</code> — that word, as it is	An API key with Mail Send permission
Brevo	<code>smtp-relay.brevo.com</code>	587	The SMTP login shown on Brevo's SMTP & API settings page	An SMTP key — not the account password, and not an API key

For either, choose **STARTTLS**, leave **Domain** empty, and verify the **From Address** (or its domain) with the service before you save.

More than one site. Each site has its own Email Server settings and its own e-mail channel. When the sites share one Reporting database (centralized reporting), a notification is sent by the server of the site its publication belongs to, through that site's mail server. A publication saved before this release belongs to no site until it is next edited (or an administrator assigns it: *Deployment Runbook* §1am), and any site's server may send its notifications; until then, give every site the same mail server settings.

Warning: Connection settings affect the whole application. Test each connection before saving — a bad connection takes every screen down.

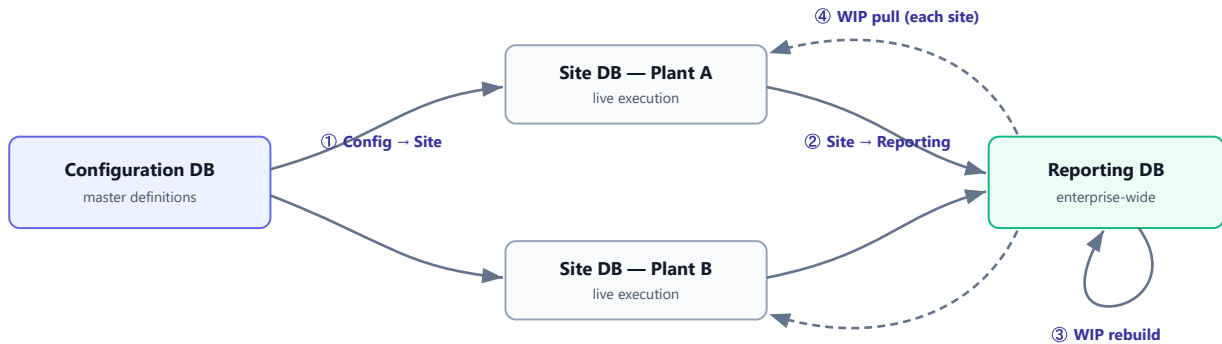
When IntelliTrack cannot reach its databases — on a new installation, or when the Site or Reporting database has moved or is down — every page sends an administrator straight to **System Connections**, and other signed-in users are refused until it is fixed. Because the role mapping that says who is an administrator is itself kept in a database, a member of the web server's own local **Administrators** group is let in to System Connections (and only there) while the databases cannot be reached.

Database synchronization jobs — you must schedule four

IntelliTrack runs on **three databases**, each with a distinct job:

- The **Configuration** database is the single master for everything you *define* — the equipment model, processes, materials, recipes, BOMs, units of measure, dispositions, approval statuses, locations, and so on — together with the electronic signatures and audit-trail entries made against those definitions. (System parameters are the exception: each site keeps its own.)
- Each **Site** database holds one plant's live *execution* data — the check-ins, check-outs, audits, and events that happen on that floor — plus a read-only replica of the configuration it needs to run.
- The centralized **Reporting** database gathers execution data from **every** site so the Reports, Dashboards, OEE, and WIP screens can query the whole enterprise in one place.

Data does not move between these databases on its own. **An administrator must create four scheduled jobs** so that definitions flow out to the floor, execution data flows in for reporting, and the work-in-process counts stay current and reach each site's WIP limit check. Without them, new recipes or materials never reach a site, reports and dashboards silently go stale, and WIP limits are checked against old counts.



1. Configuration → Site (one job per site). Runs **inside each Site database** and executes `sync.ConfigMaster`, which pulls the latest configuration replica from the Configuration database. This is how a new or edited material, recipe, BOM, unit or piece of equipment actually reaches the floor. It is a one-way *pull* — nothing a site does is written back to Configuration — and it is self-throttling, so a run finds almost nothing to do and returns quickly when configuration has not changed. Schedule one of these in **every** site database.

```
EXEC sync.ConfigMaster;
```

Each run of job 1 does two more things a site relies on:

- It copies into the Site database the electronic signatures and audit-trail entries recorded in the Configuration database for this site — a signed recipe approval, for example — so the site, and through job 2 the Reporting database, holds the complete record. Until the Site database has been bound to its site (by marking the site **Is Current Site** in the equipment model — [Set the current site](#)), these records wait; the first run after the binding brings everything since.
- At the end of the run it adds any system parameter a new IntelliTrack version has introduced, and restates the names, descriptions, units and limits of the shipped ones. It never changes a value you have set, except to pull one that lies outside its parameter's limits back to the nearest limit (recorded as updated by SYSTEM).

2. Site → Reporting (one job total). Runs **inside the Reporting database** and executes `sync.SyncMaster`, which pulls execution data — and each site's signature and audit-trail records — from every enabled site into the central Reporting store that feeds Reports, Dashboards, OEE, and Quality. Only one job is needed no matter how many sites you run — it processes each configured site in turn. If this job stops, every report and dashboard keeps showing the last data it received and quietly falls behind.

```
EXEC sync.SyncMaster;
```

3. WIP Synchronization (one job total). Runs **inside the Reporting database** and executes `sync.RebuildReportingAggregates`, which rebuilds the work-in-process and inventory rollups that power

the **WIP** and **Inventory** reports (Chapter 12), and keeps each equipment module's work-in-process counts in the Reporting database for its site to pull (job 4). This is deliberately a **separate** job from job 2: recomputing WIP over millions of records is expensive, so it runs on its own schedule and does not slow down the Site → Reporting flow. Run it as often as your WIP and inventory reports need to be current, and as often as your WIP limits need: the WIP limit check at check-out and receipt ([Production Execution](#)) uses the counts this job computes, once job 4 has brought them to the site.

```
EXEC sync.RebuildReportingAggregates;
```

4. WIP pull (one job per site). Runs **inside each Site database** and executes `wip.PullCounts`, which pulls that site's work-in-process counts from the Reporting database: the counts the WIP limit check uses and the **Work In Process Status** page shows. Like job 1 it is a one-way *pull* — the Reporting database never writes into a Site database. It reads only this site's equipment modules, and only those whose counts changed since its last run, so a run with nothing new is one small read and no writes; it never changes the WIP limits themselves. The Reporting database must be on the **same SQL Server instance** as the Site database, and System Parameter 16 (*Reporting Database Name*) must name it; a Site database on another instance cannot pull its WIP counts. Schedule one of these in **every** site database.

```
EXEC wip.PullCounts;
```

How to schedule them: Create each job in **SQL Server Agent (SSA)** or in the **FDS Event Engine** — they are **not** started from a screen in IntelliTrack. Configure each of the four jobs to run **every 2 minutes**, and set the job/command **timeout to 30 minutes (1,800 seconds)** so a long rebuild or a large first-time sync is never cut off mid-run. Running every 2 minutes is safe even for the heavier jobs: jobs 1 to 3 each take a single-run lock and simply exit if the previous run is still going, and job 4 writes only what changed, so overlapping schedules never pile up. The point of this section is that **all four must exist**. A missing job 1 starves a site of new configuration; a missing job 2 or 3 freezes reporting; a missing job 2, 3 or 4 leaves the WIP limit check working from old counts.

Grant the job's login VIEW SERVER STATE . The single-run lock heals itself: if a run is ever cut off mid-flight (a command timeout, a server restart, a deployment), the next scheduled run checks whether the previous holder is genuinely still executing and, if it is not, takes over immediately. That check reads the server's session list, which requires the server-level **VIEW SERVER STATE** permission. Without it nothing breaks — but after an interruption the takeover waits out a 30-minute safety window, and syncs quietly pause until it passes. Grant it once per SQL Server instance to the login the jobs run under (`sysadmin` accounts, including `sa` , already have it):

```
GRANT VIEW SERVER STATE TO [YourJobLogin];
```

In a multi-site deployment, grant it on **every server involved**: on each Site's server for the login that runs job 1, and on the Reporting server for the logins that run jobs 2 and 3 — including each site's Event Engine login when more than one site triggers the Reporting jobs. Job 4 takes no lock and does not need it.

Tip: If a report or dashboard looks out of date, suspect these jobs first; your scheduler's job history (SQL Server Agent or the Event Engine) shows every run. A failed Config → Site run or WIP pull is logged in the site's own error log, so **Log Search** (*System logs*, below) shows it. Jobs 2 and 3 run in the Reporting database, which keeps an error log of its own, and a failure there raises the **Database Job Failure** notification ([Chapter 15](#)) — make sure someone is subscribed to it.

How sites register themselves for centralized reporting

Job 2 pulls from every site registered in the Reporting database's **sync source registry**. There is nothing to register by hand: whenever an administrator saves **Configuration** → **System Connections**, the site registers itself in the Reporting database it just pointed at — stamped with the site's identity so a central Reporting store can tell plants apart — and, when a Config database is configured, the shared Config source is registered the same way. In a **multi-site** deployment, each plant's own web application does this the moment its connections are saved; re-saving is harmless (the site simply updates its own registration, never another site's).

Two things still live in SQL Server itself:

- **Linked servers** — when a site's database is on a different SQL Server instance than Reporting, the linked server named in the sync settings must already exist **on the Reporting instance** (the save reports a warning if it does not — create it in SQL Server first).
- The **VIEW SERVER STATE** grant above for any new site's job login.

The first Job 2 run after a new site registers pulls **everything** that site has ever recorded, so expect one long first run (it holds the single-run lock; overlapping scheduled runs quietly wait their turn) before the

cycle settles back to fast incremental pulls. Confirm the site appears with fresh timestamps in the sync logs afterwards.

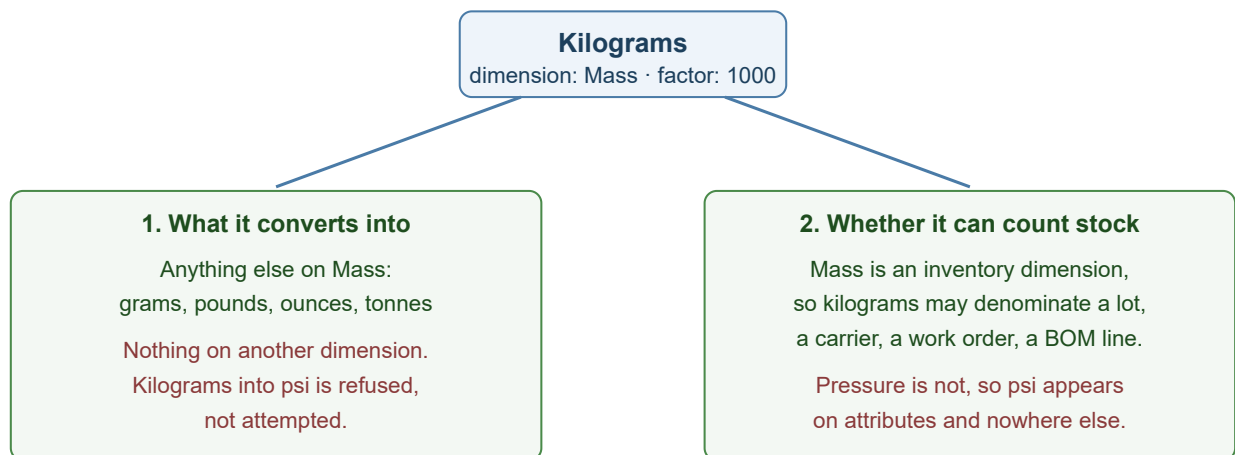
Units of measure

Configuration → **Additional Configuration** → **Materials & Logistics** → **Units of Measure**. View available to any user; **Administrators Create/Edit/Delete** the units used by materials, products, and work orders. Because quantities across the system are expressed in these units, retiring one in use has downstream effects — prefer adding over editing established units.

What a dimension is, and why every unit needs one

A unit's **dimension** is the physical quantity it measures: mass, volume, length, temperature, pressure. Every unit carries one, along with a **conversion factor** saying how big it is compared with the other units on that dimension.

The dimension decides two things, and they are the two things that make units useful rather than decorative:



Inventory quantity or measurement only

Every dimension is one or the other, and the Units of Measure list says which in the **Used for** column.

- **Inventory quantity** — the unit can count material. It is offered wherever a quantity is set: receiving a lot, a carrier's contents, a work order's build quantity, a bill-of-material line. Quantities are added, subtracted and converted as material moves, so the arithmetic has to balance. Mass, Volume, Length and Count ship this way.
- **Measurement only** — the unit records a reading. It is offered on attributes and system parameters, and never as a quantity of stock. Temperature, Pressure, Torque, Speed and the rest ship this way.

The benefit is that a lot can no longer be denominated in something that cannot be counted. Before dimensions existed, "5 Hours" was an accepted lot quantity, and splitting a lot measured in pounds into one measured in kilograms either refused the transaction or — worse — wrote a genealogy record claiming a transfer that the quantities denied.

Note: Attributes deliberately offer **every** unit, including the inventory ones. A reading can be of anything the plant instruments, and a quantity is a perfectly good thing to measure. It is only the quantity fields that are narrowed.

Adding a standard unit

Almost every unit a plant needs is already a standard one, and the system knows all of them — 1,722 units across 135 kinds of measurement, with their exact conversions. You do not have to look anything up.

1. **Configuration** → **Additional Configuration** → **Materials & Logistics** → **Units of Measure** → **Create New**.
2. Under **Find a standard unit**, type what you want — the name or the abbreviation. "kg", "litre", "psi", "degF" all work, and so do the spellings printed on the Units of Measure page.
3. Pick it from the list. Each result tells you what it measures and whether it can count stock, because that is what separates two similar-looking units.

That is the whole job. The name, the abbreviation, the dimension and the exact conversion are all filled in from the standard definition, and the banner confirms what you have chosen. Nothing below needs touching.

Note: The list only offers units the system can place. If you search for something and get nothing, it is not a standard unit — use the section below.

Adding a unit of your own

Units a plant invents are almost always containers: a drum, a tote, a reel, a spool. The system cannot know how much yours holds, so it asks — in the only terms you actually have.

1. Enter the **Name** and **Abbreviation** as usual.
2. Choose the **Dimension** — what the unit measures. The list is grouped under *Inventory quantity* and *Measurement only*, and the banner underneath restates in full what your choice means before you commit to it.
3. Open **My unit is not in the standard list** and answer the question there: *One of my units is* — say, **55 US Gallons**.

The conversion is worked out from that. You are never asked for a conversion factor, and you never need to know that the system keeps volumes in millilitres.

Checking a conversion

Further down the same **My unit is not in the standard list** section you can see and edit the raw values — the **Conversion factor** and **Zero offset** — and, under **Check the factor before you save**, convert against any unit you already recognise: choose it in **Compare against** and select **Check** — "1 Drums = 200 Kilograms (kg)".

You should not normally need this. It is there because a plant that knows better must not be locked out, and because a **wrong** factor is the one failure this screen cannot catch for you: a missing factor is refused and explained, but a wrong one is trusted silently on every transaction afterwards, and a mass factor out by a thousand looks exactly like a correct one until a lot draws down a thousand times too fast. If you do edit these by hand, convert against something familiar first.

Note: The check uses the numbers currently in the fields, not the ones last saved, so it answers for the change you are about to make.

Units that have no dimension

A unit added before dimensions existed — every custom unit at an existing site — starts unclassified, and the list flags it as **no dimension assigned**. Such a unit still works exactly as it did: records already denominated in it keep their quantities, and it stays selectable on any record already using it, so nothing goes read-only. What it cannot do is convert, and it will not be offered on new quantity fields until someone classifies it.

Classify one by opening it, choosing a dimension and entering a factor. A dimension without a factor, a factor without a dimension, or an offset without a factor are all refused — each of those looks classified and converts like nothing.

Temperature and the zero offset

Most units convert by a factor alone. Temperature does not: zero degrees Celsius is not zero degrees Fahrenheit, so the scale needs a shift as well as a stretch. That is what the **Zero offset** field is for, and it is the only place you will normally need it. Leave it empty for everything else.

Adding your own dimension

Rare, and most sites never will. **Configuration** → **Additional Configuration** → **Materials & Logistics** → **Units of Measure** → **Manage Dimensions** → **Create New** (Administrators).

1. **Name** the dimension.
2. In the **Description**, say which unit the factors are measured against — "Factors are relative to litres." Write this before anyone adds a unit to the dimension: there is no separate base-unit field,

so the description is the only record of what a factor of 1 means.

3. Leave **Units on this dimension can count stock** switched **off** unless the dimension really is a way of counting material.

Warning: Switching a dimension to count stock lets every unit on it be chosen as a lot, carrier, work order or BOM quantity, and those quantities are added and subtracted. Turning it on later is easy; turning it off after lots exist in those units is not.

Deleting a dimension does not delete its units — it unclassifies them. The delete page states how many units that affects before you confirm.

Inventory locations

Locations are where inventory sits, referenced by equipment and products. **Configuration → Additional Configuration → Materials & Logistics → Locations:** view available to any user; the **Engineering** role and **Administrators Create** (optionally under a **Parent Location**, and placed in the plant model), **Edit**, and **Delete** them. Locations are kept in the Configuration database and shared by every site. Deleting one also clears it as the default location of any equipment that used it, and detaches this site's products from it, so nothing is left pointing at a location that no longer exists.

Approval statuses

The states of the approval lifecycle that materials, recipes and bills of material move through — Draft, Under Review, Approved, Active (with the system-set *Active - Pending Effective Date*), Inactive, Obsolete and Archived ([Chapter 4](#)). **Configuration → Additional Configuration → Quality & Change Control → Approval Statuses** lists them; any signed-in user can open a status's **Details**.

What an **Administrator** can change is the wording, not the rules. **Edit** offers a status's **Label** and **Description** — its **Name** is fixed, and the **Display Name** (the label and description together, as status lists show it) follows as you type. What each status permits — whether a recipe's values and a bill of material's items can still be changed while it holds that status (as shipped, only in Draft), whether it may be used in production or for evaluation, and whether it is a system-set state that nobody can choose — is set by IntelliTrack and cannot be changed on this page. So relabelling a status to match your quality system's wording changes what people read wherever the status appears, never how the lifecycle behaves. Approval statuses are kept in the Configuration database and shared by every site.

Saving without a change saves nothing. A change is recorded in the audit trail, and it is signed if your site's signature policy requires it ([Chapter 20](#)) — the signature covers the label and description before and after, and what the status permits. Once a change has been signed, the status's **Details** page shows an **Electronic signatures** section.

Branding

Configuration → **Additional Configuration** → **Security & Administration** → **Branding** (Administrators) puts your organization's identity on IntelliTrack: its name, colors, logo, browser-tab icon and contact details. The benefit is that operators and auditors see your plant's system rather than a vendor's — without a custom build, and reversibly.

- **Company Identity** — the **Company Name**, shown in the menu bar and on the sign-in page when there is no logo, and in the page footer.
- **Brand Colors** — the **Primary Color** (header, footer and table headers), the **Secondary Color** (buttons and highlights) and the **Accent Color** (links and hover accents), each typed as #RRGGBB or picked. The text color on primary surfaces is worked out for you so it stays readable, and a warning appears if the accent color would be hard to read on white. The **Live Preview** shows the result before anything is saved.
- **Logo & Favicon** — SVG, ICO, PNG or JPG; a logo up to 512 KB, a favicon up to 64 KB. Leave a field empty to keep the current image. Each file is checked: one whose contents are not the image type it claims, or an SVG containing scripting or external references, is refused.
- **Contact Information** — **Contact Email**, **Support Email**, **Contact Phone**, **Address** and **Company URL**, shown in the footer in place of the default Factory Data Systems contact details.

Select **Save Branding**; the new look applies across the application on the next page load. Anything left empty falls back to the IntelliTrack default. **Clear All Branding** removes every customization in one step, after a confirmation, and returns the installation to stock IntelliTrack branding; the "Powered by IntelliTrack" mark and the IntelliTrack software copyright notice always remain. Branding is kept in the Configuration database, so it applies to every site — another site's web server picks it up the next time it restarts — and every change is recorded in the audit trail.

API clients

API clients hold the **API keys** external systems use to call IntelliTrack's APIs (e.g., the work-order API, Chapter 7). They are kept in the Site database, so a key works only at the site that issued it: an integration that calls two sites needs a client at each.

1. Open **Configuration** → **API Clients** (Administrators).
2. Select **Create New**, give the client a unique **Name** and a **Description**, and select **Create New** again. The **API Key Created** page shows the key **once**: copy it into the external system now. IntelliTrack keeps only a fingerprint of the key and can never show it again; the system sends the key in the `X-Api-Key` header of every request.
3. **Deactivate** stops a client's key working immediately, without deleting anything; **Activate** restores it. The list shows each client's **Key** prefix (to tell keys apart), whether it is **Active**, and when it was **Last Used**.

Everything a client records through the API is attributed to that client: it acts as its own service identity (listed under **Security** → **Users**, [Chapter 19](#)), never as a person, and it can never sign. One client per integration therefore also tells you which system did what.

Warning: An API key is a credential that bypasses the interactive login — treat it like a password. Issue one client per integration so you can revoke one without affecting the others, and deactivate clients you no longer use. A lost key cannot be recovered: create a new client, and deactivate the old one.

Licensing

Configuration → **Licensing**. Any signed-in user can open the page and see where the site stands; only **Administrators** can activate, import or allocate.

- **License Requirements** shows whether this site is within its license. **Equipment Module Capacity** gives the equipment modules in use against the licensed capacity (and, where capacity is reserved site by site, this site's share and the shared pool). **Installed Licenses** lists each license with its type, capacity, start and expiration, whether it is **Valid**, and a **Diagnostic** saying why not. Until a base license is installed IntelliTrack runs on a 60-day trial, and the page shows when the trial ends.
- **License Allocation by Site** (when an equipment-module license is installed) reserves part of the capacity for particular sites, so one plant cannot use up another's equipment modules; capacity you leave unallocated is shared by every site. Change the counts and **Save** — the total cannot exceed what the license grants.
- **License Activation** installs a license: **Activate License Online** with the license **Id**, **Password** and an **Installation Name** (the web server must be able to reach the licensing service), or **Import License** to load a license file you were sent (**Import *_license.xml**). Licenses are stored in the central Configuration database and shared by every site, so one installation covers them all.
- **Notification History** lists the license-expiry emails IntelliTrack has sent — 90, 60 and 30 days before a license expires, and daily over its last 10 days — to the **License Expiry Recipients (CSV)** on the **Email Server (SMTP)** page, with whether each succeeded. The same expiry can also reach subscribers as a notification ([Chapter 15](#)).

Note: A license banner appears across the top of the application when attention is needed — a trial's remaining days, a license expiring within 30 days, the equipment-module limit reached, or a license that needs reactivating. Act on it before it starts gating functionality: with neither a valid base license nor a running trial, every page except Licensing, the System Connections, Email Server and Branding settings, and the error pages sends people to Licensing, and the API refuses calls.

If the server changes. A license activated online can be tied to the server it was activated on. If IntelliTrack later starts on different hardware — a server restored onto a new virtual machine, for example

— the license keeps working for 60 days while the banner and its **Diagnostic** ask for it to be reactivated (**Activate License Online** again); after that it is no longer valid on that machine. A license that was never tied to a server is not affected.

System logs — Log Search

Configuration → **System Errors & Warnings** opens **Log Search**: one page that searches this site's error log and warning log together, newest first. Any signed-in user can use it, and it is the first place to look when diagnosing a problem — a failed check-out from the production gate (Chapter 8), a work-order quantity crossing logged in Warn mode (Chapter 7), a genealogy transaction whose units could not be reconciled, a failed Config → Site synchronization.

- Type one or more words in the search box: **every** word must appear somewhere in an entry's message, procedure, serial number or equipment name. Where the server has SQL Server full-text search installed, a word matches the start of a word in the entry; otherwise it matches any part of the text — so the same search can return slightly different results on two servers.
- Narrow the list by severity (**Errors Only** or **Warnings Only**), procedure, equipment, and **From Date** / **To Date** (whole days in the web server's time zone; the To Date is included). The filters stay in place as you page through the results.
- Each row shows its **Severity** (**Error** or **Warning**), **Id**, **Insert Time**, **Error Message**, **Procedure**, **Serial Number** and **Equipment**; **Details** opens the full entry.

The home page's **Diagnostics** tile opens a different page, **Error Logs**, which links to **Log Search** and to the older, separate **IntelliTrack Errors** and **Warnings** lists. Those search only their own log, for a single phrase (typing an error's number in the IntelliTrack Errors search opens that error directly).

Field Reference

Screen	Key fields / actions
System Connections	Config Database , Site Database and Reporting Database connection details, each with its own Test ... Connection button; Synchronization Settings (linked servers); optional full Config / Reporting sync on Save Configuration .
Email Server (SMTP)	Host , Port , Encryption (STARTTLS, usually 587; TLS from the start, usually 465; None), Username , Password , Domain (empty for an email service), Timeout (seconds) , From Address , License Expiry Recipients (CSV) — all email: notifications (through the site's e-mail channel) and the license-expiry warnings (not with TLS from the start).
Date and Time Format	One of six preset formats for the whole site, with a live example; Set (Administrators). Stored as System Parameter 130.
System Parameter	Numeric/Boolean Value , String Value or Timestamp ; Minimum Value / Maximum Value (fixed on shipped parameters); Units ; Can Be Edited . Create/edit/delete; applies to this site; edit and delete signable, 111–116 signed by default under GxP Mode.

Screen	Key fields / actions
Unit of Measure	Find a standard unit fills in everything from the standard definition; for a unit of your own, name what it measures and how much of something familiar it holds. Conversion numbers and a live check are available but not normally needed. Create/edit/delete.
Unit Dimension	Name, base-unit note in the description , and whether units on it can count stock ; create/edit/delete (Administrators).
Location	Name, description, optional Parent Location , and its place in the plant model; create/edit/delete.
Approval Status	Label and Description (the Display Name follows); what a status permits is fixed. Edit signable.
Branding	Company Name ; Primary , Secondary and Accent Color ; Logo and Favicon ; contact details; Save Branding , Clear All Branding .
API Client	Name , Description ; the API key , shown once; Activate / Deactivate ; Last Used .
Licensing	Installed licenses, equipment-module capacity and site allocations; Activate License Online or Import License ; notification history.
Log Search	Search words; severity, procedure, equipment and date filters; Details for each error or warning.

Related Topics

- [Work Orders](#) — parameters 90–93 and 13, the **Equipment enforcement** page, API clients
- [Production Execution](#) — the routing (40) and shipping-delay (27) parameters, warnings
- [Materials & Suppliers](#) — the approval lifecycle behind approval statuses
- [Plant & Equipment Model](#) — binding a Site database to its site (**Is Current Site**); communication anomalies (parameter 81)
- [OEE](#) — the alarm records parameter 17 archives
- [Notifications](#) — the notification channels (email, SMS, Teams, Slack); sync-job failure alerts; license expiry
- [Reports](#) — WIP and Inventory reports refreshed by the WIP sync job
- [ERP Integration](#)
- [Security and Sign-In](#) — who holds the Administrator role; API clients' service identities
- [Electronic Signatures & Audit Trail](#) — GxP Mode, signature policies, parameters 110–116, the audit trail

Reporting Database — Data Dictionary

This reference describes the **reporting views** in the IntelliTrack Reporting database. It is written for report authors who build their own custom reports (for example in SQL Server Reporting Services, Power BI, Excel, or direct T-SQL).

To keep reporting simple and safe, end users are given access to a **curated set of read-only views** — not the underlying tables. Each view is "friendly": it resolves internal identifiers (GUIDs) to names, decodes status codes into readable text, and joins in the related lookups for you, so you can build a report without knowing the physical schema.

Access model. Your IntelliTrack administrator grants report authors read access to the views below (and the one table-valued function, `inventory.fnCarrierActivity_Friendly`) — together they are the whole reporting surface. There is no built-in report-author role: access is granted at each installation, object by object. The base tables are not part of the surface, and no data can be written through the views. If a query fails with a permission error, you are trying to read a table directly; use the matching view instead. The view definitions are encrypted in the database, so this document — not the database — is where their columns are explained.

How to read this document

- Views are grouped by subject area. Each entry gives the view's **grain** (what one row represents) and its columns as **Column | Type | Description**.
- Types are the effective SQL type of the column. Columns described as *(decoded)* are readable text derived from an internal code; *(computed)* columns are calculated in the view.
- Every view is read-only. `JOIN` views together on the natural keys shown (for example `Equipment`, `SerialNo`, `Material`) to combine subject areas.

Conventions used throughout

Concept	Convention
Names, not GUIDs	Equipment, materials, processes, units, etc. appear as their names . Where a view keeps an identifier (e.g. <code>EquipmentId</code> , <code>SiteId</code>), it is provided so you can join views to one another.
Timestamps	Date/time columns are <code>DATETIMEOFFSET</code> (date + time + UTC offset). <code>DateCreated</code> = when created; <code>LastUpdated</code> = when last changed.
Pass/Fail	Production and measurement results are decoded to text: Good / Reject / In-Process (processing) or Pass / Fail / Pending (audits).

Concept	Convention
Status	Product, work-order, and approval statuses appear as their configured names (e.g. <i>Released</i> , <i>Hold</i> , <i>Completed</i>), not numeric codes.
Quantities vs. measurements	Quantities — Amount , work-order quantities, BOM minimums and maximums — are exact decimal(18,6) values, so they add up without rounding. Measured values and recipe limits (Actual , Target , alarms) are floating-point (real / float).
Serial numbers	SerialNo and Alias hold up to 100 characters here (a site limits them to 82, so every value fits). They are unique within one site but not across sites: in a Reporting database that several sites feed, the same serial or alias can belong to two plants' parts, so add SourceSite to a join on them, or join on ProductId where a view offers it.
SourceSite	The name of the site a row was synchronized from. It is filled only for rows that arrived after the site registered itself with its site identity (saving System Connections — see System Configuration); rows from an installation never registered that way, or from before it was, have it empty . Treat an empty SourceSite as "not attributed", not as "no site": when you filter to one site, include the empty rows too if they may be that site's.
Current records	Definition views (recipes, materials, BOMs) already exclude soft-deleted records — every row is a current record. Operational views (products, production history) keep every record; trace.Genealogy_Friendly lists only links that have not been removed.

View overview

View	Subject	One row =
plant.PlantModel_Friendly	Plant & equipment	one piece of equipment (with its full hierarchy)
recipe.Recipe_Friendly	Recipes	one recipe revision
recipe.Material_Friendly	Materials	one material revision
trace.BillofMaterial_Friendly	Bills of material	one BOM component line
inventory.Product_Friendly	Inventory	one product/unit
track.ProductionHistory_Friendly	Production	one processing (check-out/check-in) event
track.ProductChanges_Friendly	Production	one product field changed by a user on the web
wo.WorkOrder_Friendly	Work orders	one work order
trace.Genealogy_Friendly	Genealogy	one produced↔consumed link
audit.AuditResults_Friendly	Product audits	one QA measurement

View	Subject	One row =
<code>wip.Inventory_Friendly</code>	Work-in-process	one equipment/location/process/material bucket
<code>ppd.AttributeDataView_Friendly</code>	Measurements	one measured attribute value (with recipe limits)
<code>oee.StateHistory_Friendly</code>	OEE	one equipment state/mode interval
<code>oee.FirstTimeThrough_Friendly</code>	OEE / quality	first pass of a unit at a piece of equipment
<code>oee.LastTimeThrough_Friendly</code>	OEE / quality	last pass of a unit at a piece of equipment
<code>track.CarrierHistory_Friendly</code>	Carriers	one carrier event (a whole-tray cycle or a maintenance)
<code>inventory.CarrierCycleLife_Friendly</code>	Carriers	one carrier (cycle counts and last maintenance)
<code>inventory.CarrierYield_Friendly</code>	Carriers	one carrier at one piece of equipment (runs and rejects)
<code>ppd.CarrierAttribute_Friendly</code>	Carriers	one carrier, equipment and attribute (measurement statistics)
<code>inventory.fnCarrierActivity_Friendly(...)</code>	Carriers (function)	one carrier at one piece of equipment in a chosen window (events)

Plant & equipment

`plant.PlantModel_Friendly`

The plant model flattened to one row per piece of equipment, with the full ISA-95 hierarchy and the process performed. The `...Id` columns let you join the operational views (production, OEE, WIP, inventory) to any level of the hierarchy. Active (non-deleted) equipment only.

Column	Type	Description
EquipmentId	uniqueidentifier	Equipment key — join target for other views.
Equipment	varchar	Equipment name.
EquipmentDescription	varchar	Equipment description.
EquipmentCode	varchar	Optional business/external code.
DefaultIdealRate	real	Ideal production rate (for OEE performance).

Column	Type	Description
ProcessId	uniqueidentifier	Process key.
Process	varchar	Process/operation name.
IsFinalProcess	bit	1 = final process in a route.
IsReworkProcess	bit	1 = rework process.
WorkcellId	uniqueidentifier	Workcell key.
Workcell	varchar	Workcell name.
AreaId	uniqueidentifier	Area key.
Area	varchar	Area name.
SiteId	uniqueidentifier	Site key.
Site	varchar	Site name.
EnterpriseId	uniqueidentifier	Enterprise key.
Enterprise	varchar	Enterprise name.

Recipes

`recipe.Recipe_Friendly`

One row per recipe revision, with type, produced material, approval status, and owning site resolved. Current (non-deleted) recipes only.

Column	Type	Description
RecipeId	uniqueidentifier	Recipe key.
Recipe	varchar	Recipe name.
Revision	varchar	Revision label.
Description	varchar	Description.
RecipeType	varchar	Recipe type name.
Material	varchar	Produced material name.
MaterialRevision	varchar	Produced material revision.
ApprovalStatus	varchar	Approval status name.
ApprovedForProduction	bit	1 = this status is usable in production.
RequiredApprovals	tinyint	Approvals required to release.
RequiredOverrideApprovals	tinyint	Approvals required for an override.

Column	Type	Description
Site	varchar	Owning site name.
EffectiveDate	datetimeoffset	When the revision becomes effective.
DateCreated	datetimeoffset	Created timestamp.
LastUpdated	datetimeoffset	Last change timestamp.
UpdatedBy	varchar	User who last changed the recipe.

Materials

`recipe.Material_Friendly`

One row per material (part) revision, with approval status resolved. The binary part image is intentionally omitted. Current (non-deleted) materials only.

Column	Type	Description
MaterialId	uniqueidentifier	Material key.
Material	varchar	Material name.
Revision	varchar	Revision label.
Description	varchar	Description.
Code	varchar	Optional business/external code.
ApprovalStatus	varchar	Approval status name.
ApprovedForProduction	bit	1 = this status is usable in production.
RequiredApprovals	tinyint	Approvals required.
EffectiveDate	datetimeoffset	When the revision becomes effective.
ExpiryTime	int	Shelf-life quantity.
ExpiryUnits	tinyint	Shelf-life units.
SNFormula	varchar	Serial-number formula key.
SyncAcrossSites	bit	The material's Synchronize Across Sites tick-box, as stored. It does not currently decide which sites receive the material: every material reaches every site.
DateCreated	datetimeoffset	Created timestamp.
LastUpdated	datetimeoffset	Last change timestamp.
UpdatedBy	varchar	User who last changed the material.

Bills of material

`trace.BillOfMaterial_Friendly`

A static bill of material flattened to one row per component line: the finished material, the component material, the allowed consumption range, and the optional bound process. Current (non-deleted) BOMs only.

This view becomes available after the first data synchronization has run (BOM definitions are authored centrally and read from there), and is refreshed by every synchronization after that — so an administrator can grant access to it only once that first synchronization has created it. To compare a BOM against what was actually consumed, join `FinishedMaterial` / `ComponentMaterial` to `trace.Genealogy_Friendly`, or use the **BoM Consumption / Variance** report ([Reports](#)).

Column	Type	Description
BomId	uniqueidentifier	BOM key.
Bom	varchar	BOM name.
BomRevision	varchar	BOM revision.
BomDescription	varchar	BOM description.
FinishedMaterial	varchar	Parent (finished) material name.
FinishedMaterialRevision	varchar	Finished material revision.
ComponentMaterial	varchar	Component material name.
ComponentMaterialRevision	varchar	Component material revision.
MinAmount	decimal(18,6)	Minimum required quantity of the component (NULL = no minimum).
MaxAmount	decimal(18,6)	Maximum allowed quantity of the component (NULL = no upper limit).
Units	varchar	Unit of measure name.
UnitsAbbreviation	varchar	Unit abbreviation.
Process	varchar	Process the component is bound to (if any).
ApprovalStatus	varchar	BOM approval status name.
ApprovedForProduction	bit	1 = this status is usable in production.
EffectiveDate	datetimeoffset	When the BOM becomes effective.
LastUpdated	datetimeoffset	Last change timestamp.

Inventory

inventory.Product_Friendly

One row per product (serialized unit or lot), with material, supplier, status, and its current equipment/location/carrier resolved. Every product is included, whatever its status — filter on `Status` (or `OKToWork` / `OKToShip`) for live stock.

Column	Type	Description
ProductId	uniqueidentifier	Product key — join target for other views.
SerialNo	varchar	Serial number / lot identifier.
Alias	varchar	Alternate identifier.
Description	varchar	Description.
Material	varchar	Material (part) name.
MaterialRevision	varchar	Material revision.
Supplier	varchar	Supplier name (for received material).
Status	nvarchar	Product disposition/status name (e.g. <i>Released, Hold, Shipped</i>).
OKToWork	bit	1 = the unit's status lets it be worked.
OKToShip	bit	1 = the unit's status lets it be shipped.
PassFailStatus	varchar (decoded)	Decoded from the product's pass/fail code: 1 → <i>Pass</i> , 0 → <i>Fail</i> , any other code → <i>Not Evaluated</i> . Read it with care: IntelliTrack records 0 while a product is in process (when it is created and at every check-out) and –1 when a check-in rejects it, so here <i>Fail</i> means in-process and <i>Not Evaluated</i> includes rejected products. For good / in-process / reject judgements use <code>track.ProductionHistory_Friendly.Result</code> .
IsRework	bit	1 = the unit is in rework.
Equipment	varchar	Current/last equipment.
Location	varchar	Current location.
Carrier	varchar	Carrier holding the unit.
Position	smallint	Position within the carrier.
Amount	decimal(18,6)	Quantity (for lots).
Units	varchar	Unit of measure name.
UnitsAbbreviation	varchar	Unit abbreviation.
ExpiryDate	datetimeoffset	Expiration date/time.
ShippingCode	nvarchar	Shipping/label code.

Column	Type	Description
DateCreated	datetimeoffset	Created timestamp.
LastUpdated	datetimeoffset	Last change timestamp.
SourceSite	varchar	Originating site (see <i>Conventions</i>).

Production

`track.ProductionHistory_Friendly`

The core throughput fact: one row each time a product is processed at a piece of equipment. Use it for yield, cycle-time, and throughput reporting — and, through the work-order columns, for yield and scrap by order, and for schedule adherence (compare an order's earliest `StartTime` and latest `EndTime` with its `PlannedStartTime` and `RequiredTime`). On an installation that does not use work orders those columns are simply empty.

Column	Type	Description
HistoryId	uniqueidentifier	Processing-event key.
SerialNo	varchar	Serial number the product was worked under. On an event that renamed the part it reads <i>old-serial</i> --> <i>new-serial</i> , so <code>WHERE SerialNo = '...'</code> does not match that row — use <code>LIKE</code> .
Material	varchar	The product's current material name (not necessarily the material at this event).
MaterialRevision	varchar	Material revision.
Equipment	varchar	Equipment used.
Process	varchar	Process performed.
StartOperator	varchar	Who started the event (checked the product out).
EndOperator	varchar	Who closed the event (checked it back in). Events recorded before the two were kept apart carry the closing operator in both columns.
StartTime	datetimeoffset	Process start.
EndTime	datetimeoffset	Process end (empty while in process).
CycleTimeSeconds	real (<i>computed</i>)	Duration in seconds; 0 while the event is still open and for spans of 20 days or more.
Result	varchar (<i>decoded</i>)	<i>Good / Reject / In-Process</i> .

Column	Type	Description
FailedCheckout	bit	1 = the equipment presented the product but the check-out was refused.
Amount	decimal(18,6)	Quantity processed.
Units	varchar	Unit abbreviation.
WorkOrder	varchar	Work order the event was stamped with when it happened; empty for production without an order, or once that order has been purged.
WorkOrderId	uniqueidentifier	Key of the stamped work order — kept after the order is purged, so its events still group together.
WorkOrderGroup	varchar	The work order's grouping label.
WorkOrderMaterial	varchar	Material the work order produces.
Customer	varchar	The work order's customer.
PlannedQuantity	decimal(18,6)	The work order's planned quantity.
PlannedStartTime	datetimeoffset	The work order's planned start.
RequiredTime	datetimeoffset	The work order's due/required time.
SourceSite	varchar	Originating site (see <i>Conventions</i>).

track.ProductChanges_Friendly

One row per **field changed** when a user changes a product on the web — editing it, deleting it, or moving it between carriers — so "who changed this expiry date, and from what to what?" is a plain **WHERE** clause. The old and new values are stored as the text they showed at the time (a status, location or work-order name, for example), so renaming something later does not rewrite the record. Processing by equipment makes no rows here.

Column	Type	Description
HistoryId	uniqueidentifier	Key of the history event that recorded the edit (joins to <code>track.ProductionHistory_Friendly.HistoryId</code>).
SerialNo	varchar	The product's current serial number.
Material	varchar	The product's current material name.
Equipment	varchar	Equipment recorded with the change, if any (an ordinary edit records none; a deletion records where the product was).
ChangedBy	varchar	Who made the change.
ChangedAt	datetimeoffset	When.

Column	Type	Description
Field	nvarchar	Which field changed: <i>SerialNo, Alias, Description, Material, Disposition, Amount, Units, PFStatus, ExpiryDate, Location, Carrier, Position, Supplier</i> or <i>WorkOrder</i> .
OldValue	nvarchar	The value before the change.
NewValue	nvarchar	The value after it.
SourceSite	varchar	Originating site (see <i>Conventions</i>).

Work orders

wo.WorkOrder_Friendly

One row per work order, with customer, material, unit, and status resolved.

Column	Type	Description
WorkOrderId	uniqueidentifier	Work-order key.
WorkOrder	varchar	Work-order number/name.
Group	varchar	Grouping label.
Description	varchar	Description.
Customer	varchar	Customer name.
Material	varchar	Produced material name.
MaterialRevision	varchar	Material revision.
Status	nvarchar	Work-order status name (e.g. <i>Released, Running, Completed</i>).
PlannedQuantity	decimal(18,6)	Planned quantity.
StartedQuantity	decimal(18,6)	Quantity started.
CompletedQuantity	decimal(18,6)	Quantity completed.
RejectedQuantity	decimal(18,6)	Quantity rejected.
YieldPercent	decimal(9,2) (computed)	Completed ÷ (completed + rejected) × 100; empty until the order has produced anything. Answers scrap rate without joining back to the products.
OverbuildTolerance	real	Percentage production may run past the planned quantity (0 = none).
Units	varchar	Unit abbreviation.
PlannedResource	varchar	Planned resource.
PlannedStartTime	datetimeoffset	Planned start.

Column	Type	Description
RequiredTime	datetimeoffset	Due/required time.
ExpiryDate	datetimeoffset	When the order <i>may</i> be cleaned up — expiry does not stop the order; pruning also needs it to hold no product and not be dispatchable (Work Orders).
DateCreated	datetimeoffset	Created timestamp.
LastUpdated	datetimeoffset	Last change timestamp.
UpdatedBy	varchar	User who last changed the work order.

Genealogy

`trace.Genealogy_Friendly`

One row per produced↔consumed relationship. Each row ties a **produced** (assembly/receiver) unit to a **consumed** (component) unit. Active (non-deleted) links only.

Column	Type	Description
GenealogyId	uniqueidentifier	Genealogy-record key.
TransactionType	varchar	Transaction type name.
ProducedSerialNo	varchar	Assembly / receiving unit serial.
ProducedMaterial	varchar	Produced material name.
ProducedAmount	decimal(18,6)	Amount produced.
ProducedUnits	varchar	Produced unit abbreviation.
ConsumedSerialNo	varchar	Component unit serial.
ConsumedMaterial	varchar	Consumed material name.
ConsumedAmount	decimal(18,6)	Amount consumed.
ConsumedUnits	varchar	Consumed unit abbreviation.
Equipment	varchar	Where the transaction occurred.
UserName	varchar	Who created the link (the operator).
DeletedByUserName	varchar	Who removed the link. Because this view lists only links that have not been removed, it is empty on every row.
DateCreated	datetimeoffset	When it occurred.
SourceSite	varchar	Originating site (see <i>Conventions</i>).

Product audits

`audit.AuditResults_Friendly`

One row per QA-entered measurement, with the parent audit record's overall result, the audited equipment/product, and the attribute resolved.

Column	Type	Description
RecordId	uniqueidentifier	Audit-record key (groups the measurements of one audit).
Equipment	varchar	Equipment audited.
SerialNo	varchar	Product serial number.
AuditOpened	datetimeoffset	When the audit record opened.
AuditCompleted	datetimeoffset	When the audit was completed.
AuditResult	varchar (decoded)	Overall record result: <i>Pass / Fail / Pending</i> .
OpenedBy	varchar	Who opened the record.
CompletedBy	varchar	QA tech who closed the record.
Attribute	varchar	Attribute measured.
Actual	float	Numeric measurement value.
StringActual	varchar	String/categorical measurement value.
LowLimit	float	Low limit the measurement was judged against — the recipe's low alarm for the revision the product was made under (empty when the recipe sets none). QA cannot change it.
HighLimit	float	High limit the measurement was judged against — the recipe's high alarm, as above.
MeasurementResult	varchar (decoded)	Per-measurement result: <i>Pass / Fail / Not Judged</i> .
Skipped	bit	1 = QA chose not to measure this attribute.
Units	varchar	Unit abbreviation.
Notes	varchar	QA free-text notes for the record.

Work-in-process

wip.Inventory_Friendly

Pre-aggregated occupancy counts for fast WIP / inventory dashboards. One row per (equipment, location, process, material, unit) combination. Counts are occupancy — a unit is counted while it has an open processing event (it has been checked out at a piece of equipment and not yet checked back in) — bucketed by the unit's pass/fail status into Good / In-Process / Reject. These are the same totals the Inventory Levels and Work In Process Status reports show, rebuilt by the WIP synchronization job ([System Configuration](#)).

Column	Type	Description
Aggregateld	uniqueidentifier	Row key.
Equipment	varchar	Equipment name.
Process	varchar	Process name.
Location	varchar	Location name.
Material	varchar	Material name.
Units	varchar	Unit of measure name.
UnitsAbbreviation	varchar	Unit abbreviation.
GoodCount	int	Count of good units.
InProcessCount	int	Count of in-process units.
RejectCount	int	Count of reject units.
TotalCount	int (<i>computed</i>)	Good + In-Process + Reject.
GoodAmount	decimal(18,6)	Summed good amount.
InProcessAmount	decimal(18,6)	Summed in-process amount.
RejectAmount	decimal(18,6)	Summed reject amount.
LastCalculated	datetimeoffset	When the row was last rebuilt.

Measurements

ppd.AttributeDataView_Friendly

One row per measured attribute value, with the equipment, product, and unit resolved **and the recipe limits that were in force joined in** — so you can chart a reading against its target and alarm/warning bands without any extra joins. The limits come from the recipe revision the value was recorded under; a

value recorded under no recipe has them empty. Only values recorded at a piece of equipment appear — values loaded without one are not in this view.

Column	Type	Description
PPDId	uniqueidentifier	Collection-event key (groups values recorded together).
Equipment	varchar	Equipment used.
SerialNo	varchar	Product serial number.
DateCreated	datetimeoffset	When the value was recorded.
Operator	varchar	Who submitted the measurement set this value belongs to (which can differ from whoever closed the processing event).
Audit	bit	1 = recorded during a QA audit.
Attribute	varchar	Attribute name.
BitTarget	bit	Target for boolean attributes.
BitActual	bit	Actual value for boolean attributes.
LowAlarm	real	Low alarm limit (from the recipe).
LowWarning	real	Low warning limit (from the recipe).
Target	real	Target value (from the recipe).
Actual	real	Actual numeric reading.
HighWarning	real	High warning limit (from the recipe).
HighAlarm	real	High alarm limit (from the recipe).
StringTarget	varchar	Target for string attributes.
AllowedValues	varchar	For a multi-value attribute, the accepted values in one text, separated by vertical bars — the preset the value was actually judged against (<code>StringTarget</code> is usually blank then).
MultiValue	bit	1 = a multi-value attribute (judge it against <code>AllowedValues</code>).
StringActual	varchar	Actual value for string attributes.
PFResult	int	Per-value limit conformance: 1 = within limits, 0 = neutral (no limits to compare), -1 = evaluated failure (rejects the product), -2 = out of limits on a data-collection (not-evaluated) attribute — informational only, never rejects the product.
Units	varchar	Unit of measure name.
UnitsAbbreviation	varchar	Unit abbreviation.

OEE

`oee.StateHistory_Friendly`

One row per time-in-state interval on a piece of equipment, with mode and state resolved and their OEE classification flags surfaced. The current (open) interval has an empty `EndTime`.

Column	Type	Description
TrackingId	uniqueidentifier	Interval key.
Equipment	varchar	Equipment name.
Mode	varchar	Operating mode name (e.g. <i>Production</i> , <i>Maintenance</i>).
State	varchar	Equipment state name.
IsAvailable	bit	1 = the state counts as available.
IsRunning	bit	1 = running.
IsFaulted	bit	1 = faulted.
StartTime	datetimeoffset	Interval start.
EndTime	datetimeoffset	Interval end (empty if still open).
DurationSeconds	int (<i>computed</i>)	Duration in seconds; 0 while the interval is still open and for spans of 20 days or more.
UDF00 ... UDF04	varchar	User-defined fields (e.g. Shift, Product) — labels are configured per site.

`oee.FirstTimeThrough_Friendly` and `oee.LastTimeThrough_Friendly`

The first (respectively last) pass of a unit at a piece of equipment — the basis for first-time / last-time-through quality metrics. Identical shape.

Column	Type	Description
Equipment	varchar	Equipment name.
SerialNo	varchar	Product serial number.
StartTime	datetimeoffset	Pass start.
EndTime	datetimeoffset	Pass end.
Status	varchar (<i>decoded</i>)	<i>Good / Reject / In-Process</i> .

Carriers

Reusable carriers — trays, pallets, fixtures — and what they do to the parts they carry. Two sources feed these objects, and they answer different questions:

- **Whole-tray cycles** (`track.CarrierHistory_Friendly` , `inventory.CarrierCycleLife_Friendly`) exist only where whole trays are checked out by the carrier's name; on a plant that checks parts out one at a time they stay empty.
- **The carrier a part was on when it was processed** is stamped on every processing event, whole-tray or single-part, so `inventory.CarrierYield_Friendly` , `ppd.CarrierAttribute_Friendly` and `inventory.fnCarrierActivity_Friendly` report on a plant that checks parts out one at a time too. (The Carrier Reports pages are built on these objects — [Reports](#).)

`track.CarrierHistory_Friendly`

One row per carrier-level event: a whole-tray cycle (from group check-out to group check-in) or a maintenance event.

Column	Type	Description
CarrierHistoryId	uniqueidentifier	Event key.
Carrier	varchar	Carrier name.
EventType	varchar	<i>GroupCycle</i> or <i>Maintenance</i> .
Equipment	varchar	Where the cycle ran.
StartTime	datetimeoffset	Cycle start, or when maintenance was recorded.
EndTime	datetimeoffset	Cycle end (empty while the cycle is in process).
CycleStatus	varchar (<i>decoded</i>)	<i>Complete</i> / <i>In Process</i> / <i>Maintenance</i> .
MemberCount	int	Parts on the tray for the cycle.
Result	varchar (<i>decoded</i>)	<i>Good</i> / <i>Reject</i> ; empty for an open cycle or a maintenance event.
Operator	varchar	Who ran or recorded the event.
SourceSite	varchar	Originating site (see <i>Conventions</i>).

`inventory.CarrierCycleLife_Friendly`

One row per carrier that has not been deleted, with its whole-tray cycle counts.

Column	Type	Description
CarrierId	uniqueidentifier	Carrier key.
Carrier	varchar	Carrier name.

Column	Type	Description
PositionCount	smallint	Declared number of positions (empty if none declared).
LastMaintenance	datetimeoffset	Most recent maintenance event (empty if never serviced).
CyclesSinceMaintenance	int	Completed whole-tray cycles since <code>LastMaintenance</code> (all of them if never serviced).
LifetimeCycles	int	All completed whole-tray cycles.
HasOpenCycle	bit	1 = a cycle is in process now.

inventory.CarrierYield_Friendly

One row per carrier and piece of equipment: how often completed processing events for parts on that carrier ended in a reject. Judge `RejectRate` against `Runs` (a funnel plot), not on its own, so a carrier with a handful of runs is not flagged on noise.

Column	Type	Description
CarrierId	uniqueidentifier	Carrier key.
Carrier	varchar	Carrier name.
Equipment	varchar	Equipment the parts were processed at.
Runs	int	Completed processing events.
Rejects	int	Of those, how many rejected.
RejectRate	real	$\text{Rejects} \div \text{Runs}$, as a fraction from 0 to 1.

ppd.CarrierAttribute_Friendly

One row per carrier, piece of equipment and attribute: statistics of the numeric measurements recorded on processing events for parts on that carrier. A defective carrier shows up as a shifted mean or a wider spread than its peers on the measurement it distorts; compare carriers with an analysis of variance. Only measurements tied to an individual part's processing event count — those taken at a whole-tray group check-in are tied to the tray cycle instead, and do not appear here.

Column	Type	Description
CarrierId	uniqueidentifier	Carrier key.
Carrier	varchar	Carrier name.
Equipment	varchar	Equipment the measurements were taken at.
Attribute	varchar	Attribute measured.
N	int	Number of values.

Column	Type	Description
Mean	float	Average value.
StdDev	float	Sample standard deviation (empty with fewer than two values).
MinValue	float	Smallest value.
MaxValue	float	Largest value.

inventory.fnCarrierActivity_Friendly

A table-valued function rather than a view, because it takes a scope and a window:

```
SELECT * FROM inventory.fnCarrierActivity_Friendly(
    @enterpriseId, @siteId, @areaId, @workcellId, @equipmentId,
    @carrierId, @startTime, @endTime, @sinceMaintenance)
```

Each scope argument, `@carrierId`, `@startTime` and `@endTime` may be `NULL`, meaning "any"; always pass `@sinceMaintenance` as 0 or 1 (with `NULL` the function returns nothing). The scope arguments select carriers by their plant-model scope — a carrier scoped to the whole enterprise matches any site, area or equipment you name — and `@carrierId` picks one carrier. `@startTime` / `@endTime` bound the window; with `@sinceMaintenance = 1` they are ignored and each carrier is counted from **its own** last maintenance event (all its history if it has none). Only carriers that have not been deleted are included. One row per carrier and piece of equipment:

Column	Type	Description
CarrierId	uniqueidentifier	Carrier key.
Carrier	varchar	Carrier name.
EquipmentId	uniqueidentifier	Equipment key (empty for an event with no equipment).
Equipment	varchar	Equipment name.
PositionCount	smallint	Declared number of positions.
Events	int	Parts handled — not trips: a full 8-position carrier contributes 8 events per pass. Divide by <code>PositionCount</code> for a rough pass count.
FirstSeen	datetimeoffset	First event in the window.
LastSeen	datetimeoffset	Last event in the window.

This document describes the read-only reporting views exposed to report authors. If you need data that is not covered by a view here, contact your IntelliTrack administrator rather than querying the base tables — the views are the supported reporting interface, and direct table access is intentionally restricted.

Chapter 19: Security and Sign-In

Overview

This chapter covers how you sign in and out of IntelliTrack, what decides what you may do, and how to see your own access. The second half is for administrators: the **Security** pages that map Windows groups and sign-in-service roles to IntelliTrack roles, set how long sessions last, review refused sign-ins, keep one record per person across the ways they sign in, and record every security change. Electronic signatures — which also depend on how you signed in — are covered in [Electronic Signatures & Audit Trail](#).

Why it matters: every record of something done in IntelliTrack — a product's history, a genealogy link, a quarantine, an approval, a signature — names the person who did it, by their name and account name *as they were at that moment*, so a later name change or a reused account name never changes whom the record shows. (Records a machine writes on its own carry the operator name the machine reports.) Signing in is how IntelliTrack knows who that is, and the roles decide what they may do. Neither ever depends on a password IntelliTrack stores: it has none.

Before You Begin

- **Your plant chooses how people sign in.** Most sites sign people in automatically with their Windows sign-in. Wherever Windows sign-in is on, you can instead type your Windows user name and password (for computers that are not on the company network). A site may also offer the organization's sign-in service (OpenID Connect — for example Microsoft Entra ID or AD FS).
- **Roles.** IntelliTrack has eight roles of its own: *General* (the pages everyone sees, and marking a product as a reject), *Production* (line operators), *Production Supervisor*, *Recipe*, *Engineering* (the plant and equipment model), *Quality* (closing process audit records), *Audit* and *Administrators*. An administrator decides which directory groups or sign-in-service roles give each one ([Role mapping](#)). You may hold several. Production Supervisor includes everything Production may do, and Administrators pass every Production, Production Supervisor, Engineering and Quality check, except that with GxP Mode on a release decision must be signed by a Production Supervisor ([Getting Started](#)). That inclusion is built into IntelliTrack, not made by mapping. The other roles give only their own pages — a Recipe, Engineering or Quality user also needs General to see the pages everyone sees, and neither Engineering nor Quality includes a production role.
- **Sessions end on their own.** After a period without activity (15 minutes unless your administrator set another) and after a maximum length (12 hours unless set otherwise), you are asked to sign in again. On a company computer this happens without you noticing.
- **How you sign in decides whether you can sign electronically.** A Windows or user-name-and-password sign-in can give electronic signatures; a sign-in through the organization's sign-in service cannot, in this release ([Sign electronically](#)).

How to Get Here

- **Sign in:** open IntelliTrack. If you are not signed in automatically, the **Sign in to IntelliTrack** page appears and shows the ways your site offers.
- **My access:** select your name (*Current User: ...*) at the right of the menu bar.
- **Sign out:** under your name, beside **Inbox**.
- **Security pages** (Administrators role): **Configuration** → **Additional Configuration** → **Security & Administration** → **Security**, a page of tiles — **Role mapping**, **Security settings**, **Sign-in review**, **Users**, **Provider links**, **Connection test** and **Security audit trail**. (The page also shows an **Electronic signatures** tile, which has no page behind it in this release: signatures are read on the record that was signed and on the **Configuration** → **Additional Configuration** → **Compliance** pages — see [Electronic Signatures & Audit Trail](#).)

Sign in

1. Open IntelliTrack. On a company computer you are usually signed in straight away.
2. Otherwise, on **Sign in to IntelliTrack** (*Choose how to sign in.*), choose:
 - **Sign in with Windows** — uses the Windows account you are signed in to on this computer.
 - **Sign in with a user name and password** — type your Windows user name (for example `jsmith`, or `CONTOSO\jsmith`; a name without a domain is looked up in the domain named under the box) and password, and select **Sign in**. The password is checked by your company's directory over a secure connection and is never stored. This is the same account as **Sign in with Windows** — the same IntelliTrack user record, with the same roles. The form only works over a secure (`https://`) address; if the page says a secure connection is required, ask for IntelliTrack's `https://` address.
 - **Sign in with your organization's sign-in service** (the button carries its name, for example Microsoft Entra ID) — takes you to your organization's sign-in page, including any extra verification it asks for (IntelliTrack cannot style that page), then back.

If your site offers only one way in, you go straight to it: a Windows-only site signs you in with no page at all. If your browser cannot sign you in with Windows (for example on a computer outside the company network), cancel its prompt; the **You are not signed in** page that follows has a **Sign in** button that lists the other ways. Once you are in, your name appears at the right of the menu bar as your directory or sign-in service gives it.

If the sign-in is refused, the page says why — each reason has its own message, and says who can fix it. When the refusal needs an administrator, the page also shows a *Reference for your administrator*: the entry they will find on **Sign-in review**.

Message	What it means	What to do
You do not have access to IntelliTrack	You signed in, but none of your groups or roles is mapped to an IntelliTrack role.	Ask your IntelliTrack administrator; your sign-in is waiting on <i>Sign-in review</i> with the groups it presented.
Your account needs to be confirmed	You match an existing user who signs in another way; to keep your history in one place an administrator confirms it is you.	Ask your administrator to check <i>Sign-in review</i> .
The directory could not be reached	Your name could not be read from the directory, and IntelliTrack never records anyone under a guessed name.	Try again in a few minutes; tell IT support if it keeps happening.
The sign-in service could not be reached	Your password was not the problem: IntelliTrack could not reach the sign-in service. People already signed in are not affected.	Try again shortly; tell IT support if it persists.
IntelliTrack is temporarily unable to sign you in	IntelliTrack's configuration database could not be reached. People already signed in are not affected.	Try again shortly, or tell your administrator.
Sign-in failed	The user name or password was not accepted (or the account is locked or disabled).	Check them; repeated failures may lock your account.
A secure connection is required	The page is not using <code>https://</code> , which this way of signing in needs.	Open IntelliTrack at its <code>https://</code> address.
Too many groups to check	The sign-in service sent a note saying you belong to too many groups to list, instead of the list.	Your administrator can fix this, by granting access through application roles or limiting the groups sent.
Your sign-in is missing information	The sign-in service did not send your name or account name, which IntelliTrack records with everything you do.	Your administrator adjusts IntelliTrack's registration with the service.
You were not signed in	The sign-in service did not sign you in — you may have cancelled, or you are not assigned to IntelliTrack there.	Try again; if it keeps happening, ask whether you are assigned to IntelliTrack at the sign-in service.
The sign-in was interrupted	The answer from the sign-in service could not be matched to a sign-in started in this browser.	Start again, at IntelliTrack's <code>https://</code> address.
Service accounts cannot sign in	The account belongs to an automated service.	Sign in with your own account.
Sign-in method not available	That way of signing in is not turned on for this installation.	Use one the sign-in page offers.

Sign out — and shared computers

Select **Sign out**. You are shown **Signed out** (*Close the browser if you are on a shared computer*). After you sign out, IntelliTrack does not sign anyone back in automatically on that browser — not even with the Windows account the computer is signed in to — until someone selects **Sign in again** and chooses a way in. That is what lets a shared station change hands.

If you signed in through your organization's sign-in service, IntelliTrack signs you out there too, when the service supports it, so the next person at this computer has to sign in as themselves; and the next sign-in through the service from that browser asks for a password again even if the service's own session is still open.

A session also ends on its own after the idle timeout or the maximum session length (see [Security settings](#)); when the sign-in page appears, it says which. If the browser starts presenting a different Windows account (someone else signed in to the computer), the previous IntelliTrack session is ended rather than used. The inactivity lock is different: it does not sign you out, it locks the browser until you give your password (see [The inactivity lock](#)).

My access

My access answers "why can't I open this page?" without an administrator having to look. It shows what IntelliTrack knows about you right now, read from your own session and the current role mapping. The **Your sign-in** card shows **Signed in as** (your name), your **User name, Signed in with** (*Windows sign-on, User name and password (Active Directory), or Your organization's sign-in service (OpenID Connect)*) and **Checked** (when you signed in). The **Access by area** card below it has one row per IntelliTrack role: **Access** — **Allowed** or **Not allowed** — and **Granted by**, the mapped groups or sign-in-service roles that gave it to you (or *None of your groups or roles is mapped to this role*). The rows show what the mapping gives you. The inclusion of Production in Production Supervisor, and of both, Engineering and Quality in Administrators, is not a mapping, so a supervisor whose group is mapped to Production Supervisor alone can see **Not allowed** beside Production and still do everything Production may — and an administrator can see **Not allowed** beside Engineering and still do every Engineering task.

Open it by selecting your name in the menu bar. It is also linked from every page that refuses you, and it works even when your sign-in itself was refused — it then shows the refusal and a **Sign in** button — because the person who most needs it is the one who has just been refused everywhere.

Changes to your directory groups, or to your roles at the sign-in service, reach IntelliTrack only when you sign in again. **Sign in again** does that now: with Windows sign-in it happens silently; otherwise the sign-in page opens. You come back to **My access**, which then says your groups and roles were read afresh.

When your permissions change while you work

If an administrator changes the **role mapping**, the change reaches every signed-in person at their next action — no restart, no need to sign out. A role you gained works at once. A page that needs a role you

lost refuses you with **Unauthorized** and names the roles it needs (*This page needs one of these IntelliTrack roles: ...*). If the change took away every role you had, you see **Your permissions have changed** instead — not "you are not signed in". Nothing you already saved is affected.

Changes to **your own groups** — someone adds you to a group in the directory, or changes your roles at the sign-in service — take effect the **next time you sign in** (or when you select **Sign in again** on [My access](#)), not during the session you already have. Electronic signatures follow the same rule: the roles a signature checks are the ones your session holds. What a signature does check afresh is your password, so a domain account that has been disabled or locked cannot sign, even while its session lasts.

Sign electronically

When your site has switched on **GxP Mode**, many acts — approving a recipe, dispositioning a product, confirming a quarantine and more — can ask for your electronic signature: the page shows an **Electronic signature** panel with your user ID and asks for your password. What you sign, what each panel shows and what happens if a signature is refused are in [When you are asked to sign](#).

Some acts also need a particular role to sign. As shipped, the seven **release decisions** — releasing product from Hold, changing a pass/fail result other than to reject, dispositioning or rolling back quarantine, and skipping or postponing a due audit — must be signed by someone who holds **Production Supervisor**, even when an administrator can open the page (see [Who may sign: roles](#)). A site may likewise name **Engineering** as the role that must sign an engineering change, such as a change to an attribute or to a machine's enforcement; as shipped, no act requires it. Closing a process audit record is signed, as shipped, by someone who holds **Quality** or **Administrators** ([Chapter 9](#)).

How you signed in decides whether you can sign:

- **Sign in with Windows**, or **Sign in with a user name and password**, with a company account: you sign with your Windows password, which your company's domain checks at that moment — being signed in is not enough (a session is not a signature).
- **An account of the IntelliTrack server itself** (such as an [emergency account](#)): only while System Parameter 115 allows local-account signatures.
- **Sign in with your organization's sign-in service**: signing is not available in this release. Every signature — and the inactivity lock's unlock — is refused with *Your account cannot give electronic signatures on this server*. If your site also offers Windows or a user name and password, sign in that way to sign; otherwise ask your administrator.

See [Which password you sign with](#).

Administration: Security pages

These pages need the **Administrators** role. Changes to the role mapping, the security settings, sign-in review decisions and user records ask for a **Reason for the change**, and every change made here is kept, with who made it and when, in the [Security audit trail](#).

Role mapping

Security → **Role mapping** decides which Windows groups and sign-in-service roles give which IntelliTrack role. It is the only thing that does: nobody receives a role that is not mapped here, and a person matching nothing has no access at all — there is no default role and no read-only fallback. Keeping access in IntelliTrack's own roles, rather than in directory group names written into each page, means one page answers "who may do what", the same roles work whichever way a person signs in, and a change needs no restart.

The page shows a card for each role — **Administrators, Recipe, Engineering, Production, Production Supervisor, Quality, Audit, General** — with what the role is for and what grants it (**Granted by, Type, Value, and Added:** when and by whom), or *Nothing is mapped to this role, so nobody holds it*. **Can sign approvals** marks Administrators, Recipe, Engineering, Production Supervisor and Quality: the roles that approve recipes, materials and bills of materials, the role a site may name as the signer of engineering changes, the role that signs release decisions, and the role that signs a closed audit. An automated service identity can never be given any of them.

Production and **Production Supervisor** divide the floor's authority. Production is for line operators: they run the process as it was defined and record what happened — check-out and check-in, shipping verification, receiving, downtime and alarm response, a product's location or carrier, placing product on Hold, containing suspect product. Production Supervisor is for shift leads and supervisors: it includes everything Production may do, and adds the acts that release, correct, override or authorize — releasing product from Hold, changing a pass/fail result, renaming or correcting a product, scrapping, splitting and merging, work-order administration, skipping or postponing audits, dispositioning and rolling back quarantine. Map each group to the one role that describes its work: a supervisor's group needs **Production Supervisor** only, because the inclusion is built into IntelliTrack. Its shipped group is *IntelliTrack Production Supervisor*.

A plant that does not separate operators from supervisors maps the same directory group to both **Production** and **Production Supervisor** here. There is no setting for it: it is an ordinary mapping change — signed where the *Security role mapping* policy asks, and kept in the audit trail — so the decision to give everyone on the line supervisor authority is recorded, with who made it and why.

Engineering is for the controls, process and manufacturing engineers who build and maintain the plant and equipment model: enterprises, sites, areas, work cells, equipment and locations; processes and routes; attributes and attribute maps; OEE modes, states and user-defined fields; equipment enforcement; and each machine's alarm definitions. Without it, the engineer commissioning a machine would have to be made an administrator — which would also hand them security and role mapping, system parameters, licensing, branding, ERP and API connections, AI and notification setup, and signature policy. Engineering gives them the engineering work without control of the system. Administrators pass every Engineering check, so nothing is taken from them. Engineering includes no production role, and no production role includes it: an engineer who also runs the line is mapped to both. Master and reference data — units of measure, suppliers, carriers, GTINs, ECO keys and dispositions — stays with Administrators, and so does

binding the installation to its site: an engineer edits sites but cannot tick **Is Current Site** ([Set the current site](#)). Its shipped group is *IntelliTrack Engineering*.

Quality is for the quality technicians who verify production: they close a process audit record with their calibrated measurements and the verdict ([Chapter 9](#)). Anyone signed in can open the audit station, the completion form and the audit results; only Quality and Administrators can submit the form. It is separate from the production roles on purpose: the audit checks the line, so neither Production nor Production Supervisor passes it, and Quality includes no production role — a technician who also runs the line is mapped to both. Skipping or postponing a due audit stays a Production Supervisor's release decision. Its shipped group is *IntelliTrack Quality*.

Audit is the one role that changes nothing. It opens the audit trail — the workspace, the findings, the reviews and the tamper-evidence verification — and nothing else: no configuration, no production, no recipes, no system parameters, not even the audit trail's own settings. Map your quality function to it rather than making them administrators. A review means little if the only people who can read the trail are also the people most able to change a record outside the application; separating the two is what makes the independence rules elsewhere in the product mean something. Somebody who genuinely needs both is given both.

- **Add a mapping:** choose the **IntelliTrack role** and the **Type** — **Windows group**, or, when an identity provider is configured, **Identity provider application role** or **Identity provider group** — then enter the **Group, role or value**, an optional **Display name** and the **Reason for the change**, and select **Add mapping**.
 - A Windows group is typed as `CORP\IntelliTrack Admins` (suggestions appear as you type, including the built-in *Authenticated Users*, *Everyone*, *Users* and *Administrators*) or pasted as a SID. IntelliTrack looks it up and stores it by its security identifier, so renaming the group in the directory can neither break nor re-point the mapping. A name the server cannot find is refused.
 - For an identity provider, enter the exact value it sends; capitals matter. The page says which claims the values are read from. Prefer application roles: they belong to IntelliTrack's registration and are never cut short, while a long group list can be (*Too many groups to check*).
- **Remove** (with a reason) takes a mapping away.
- **Signed changes:** where the signature policy for *Security role mapping* — the act *Change which groups confer a security role*, not signed as shipped — asks for a signature, **Add mapping** carries the signature panel under the form, and **Remove** opens a confirmation page that asks for the signature before anything is removed. Signed or not, every change goes into the audit trail with who made it and why ([Electronic Signatures & Audit Trail](#)).
- A change takes effect at each signed-in person's next action, with no restart and no need to sign out. Group membership itself is read when a person signs in, so adding someone to a group takes effect at their next sign-in (they can use **Sign in again** on [My access](#)).

- Removing a mapping that would take the Administrators role away from **you** is refused (*That change would remove your own administrator access...*): you would lose the very page that undoes it. Add another mapping that grants you Administrators first, or ask another administrator.
- General is normally mapped to *Authenticated Users*, which every Windows sign-in carries. People who sign in through an identity provider do not carry it: they receive General only from a provider role or group mapped to it.
- The first time IntelliTrack started, it derived this page once from the site's `Security:Roles` settings (as shipped: *IntelliTrack Admins*, *IntelliTrack Recipe*, *IntelliTrack Engineering*, *IntelliTrack Production*, *IntelliTrack Production Supervisor*, *IntelliTrack Quality*, *IntelliTrack Audit*, and *Authenticated Users* for General). The page keeps that derivation's report (*Initial mapping created from the Security:Roles settings on ...*), including any configured group that could not be found — map those by hand. If none could be found at first start (the directory unreachable, say), a warning lists them, every sign-in is refused while nothing is mapped, and the derivation is tried again at each sign-in. After the derivation the settings are never read again: this page is the authority.
- The **Mapping revision** at the foot of the page goes up with every change. A bookmark to the old **Configuration** → **Security roles** page opens this page.

Emergency access when the domain is down

People already signed in carry on while the domain controllers cannot be reached, but signing in afresh with a company account depends on the domain — and IntelliTrack never records a new person under a guessed name. A site that must stay operable through a domain outage prepares an emergency account **before** it is needed: a local account on the IntelliTrack web server, in a local group of that server (for example `SERVER\IntelliTrack Emergency`) that an administrator maps, as a Windows group typed `SERVER\Group`, to **Administrators** and whatever other roles it needs. Windows on the server checks that account's password itself, so the account works while the domain is down, two ways: **Sign in with Windows**, which the browser answers with the credentials it holds; or **Sign in with a user name and password**, typing the account as `SERVER\name` — the server's own name in front of it, not the domain's. Either way everything done with it is recorded under its own name.

The `SERVER\` in front is what tells IntelliTrack to ask the server instead of the domain, so type it even though the password prompt normally takes a bare name. A name with a domain in front of it, or a bare name on a server that is joined to a domain, still goes to the domain controllers and is refused while they cannot be reached — an emergency account is a *different* account, and typing it as one is what keeps a domain outage from being mistaken for a remembered password. On a server with no domain at all, a bare name is the server's own account and needs no prefix.

Prepare it, and **try it, before it is needed**: sign in with it once with the domain up (it works then too), so the day it is needed nobody discovers that the account was never in the mapped group, or that a Windows policy withholds its right to sign in — if it does, the prompt says the sign-in service refused it rather than blaming the password. **Security** → **Connection test** names the account form to type on this

server. It can give electronic signatures only while System Parameter 115 (**Allow Local Account Signatures**) is 1. Keep its password sealed.

Security settings

Security → **Security settings** has one box, **Sessions**, saved with a **Reason for the change**. The foot of the page says when the settings last changed and who changed them.

- **Sessions** — **Idle timeout (minutes)**: a session with no activity for this long ends (1 to 1440; default 15). **Maximum session length (minutes)**: the longest a session can last, however active (15 to 10080, and not shorter than the idle timeout; default 720, so a session outlasts a shift). These are IntelliTrack's own settings and apply to every open session from its next request; nothing on the server may end a session sooner. They are separate from the inactivity lock (System Parameter 114, see [The inactivity lock](#)).

Whether an act needs an electronic signature is not set on this page: **GxP Mode** and the **signature policies** on **Configuration** → **Additional Configuration** → **Compliance** decide it (see [Electronic Signatures & Audit Trail](#)).

If the page opens with a red warning that the encryption keys protecting sign-in sessions are held in memory, everyone is signed out each time the application pool recycles, until the folder the warning names is made writable for the application (see the deployment runbook).

Sign-in review

Security → **Sign-in review** lists the refused sign-ins that need an administrator. Each identity appears once: **Last attempt** (and how many attempts since when), **Who** (name, account name, provider and way of signing in), **Why it was refused** — with **Details** showing what the sign-in presented, such as its groups or roles — and a **Decision**. A decision needs a reason and settles every refused attempt of that identity:

- *None of their groups or roles is mapped* — add the mapping on [Role mapping](#) (the details show exactly which groups or roles to map), then **Dismiss**; or just **Dismiss**.
- *Matches an existing user who signs in another way* — **Same person: link the accounts** (the accounts are linked when this person next signs in, and their history stays in one record), **Different person** (they get a record of their own at their next sign-in), or **Dismiss**.
- *Too many groups to list (group overage), The sign-in did not include their name or account name, The directory could not be reached for their name, A service account was used to sign in* — the cause is outside IntelliTrack; fix it there, then **Dismiss**.

Users and provider links

Security → **Users** lists every IntelliTrack user record — **Name**, **Account name** (with any other account name, such as an e-mail-style one, beneath), **Signs in with**, **Last sign-in** and **Created** — and finds them by **Name or account**. A record is keyed on its own **IntelliTrack identifier**, never on an account name, so

a rename or a change of sign-in service keeps one person's history together, and a record is never re-pointed to someone else. Open a record to see its **Linked sign-in accounts**: each with its provider, account name, identifier at the provider, when, by whom and why it was linked, and whether it is still **active**. **Unlink** (with a reason) stops that account signing in as this person; it changes no record of anything they did.

The list also shows the **Service identity** records that act without a person: IntelliTrack's own background work, and one for each API client (created the first time the client calls), under which everything that client records is kept. A service identity can never sign in interactively, never be given the Administrators, Recipe, Engineering, Production Supervisor or Quality role, and never give a signature: a machine or an API client never maintains the plant model or closes an audit.

Security → **Provider links** moves people to a new sign-in service — from Windows to Microsoft Entra ID, for example — without splitting anyone's history. Choose the **New sign-in provider** and, optionally, an **Account name suffix** (leave it empty to propose each person's user principal name or e-mail as last read from the directory, or give one such as `@example.com` to propose their account name with it), and select **Propose links**. IntelliTrack proposes which existing user each new account belongs to; nothing is linked until you approve it. Tick proposals and select **Approve selected** or **Reject selected** — each is decided on its own, so one wrong match never holds up the rest. An approved link is made when that person first signs in through the new provider (until then it reads *Approved: links at the next sign-in*). Both providers keep working throughout.

Security audit trail

Security → **Security audit trail** records every change made on these pages — role mappings, security settings, user records and their sign-in links, sign-in review decisions, provider-link proposals and decisions, and the initial role mapping — with **When, Who, What** (the kind of change and what it was made to), **Before, After** and the **Reason for the change**. **Show** narrows it to one kind of change; it pages 50 entries at a time. Entries can never be changed or removed. Changes to recipes, signature policies and every other record are in the separate audit trail on **Configuration** → **Additional Configuration** → **Compliance** (see [Electronic Signatures & Audit Trail](#)).

Connection test

Security → **Connection test** → **Run the test** checks each configured way of signing in before anyone tries: the directory's secure connection (LDAPS) and that passwords are accepted only over HTTPS; and, for an identity provider, that this site uses HTTPS, the provider's discovery document and signing keys, the sign-in flow, how IntelliTrack authenticates to the provider and whether it accepts IntelliTrack's credential, whether the two clocks agree, the claims IntelliTrack reads, the addresses to register with the provider, and signing out there. It uses nobody's password and signs nobody in. Each check reads **Passed, Warning, Failed, Information** or **Not configured**, with the detail and what to do. Fix every **Failed** before people sign in.

Field Reference

Field	Where	Meaning
Signed in with	My access	How you signed in: <i>Windows sign-on, User name and password (Active Directory)</i> or <i>Your organization's sign-in service (OpenID Connect)</i> . It decides whether you can sign electronically.
Granted by	My access, Role mapping	The mapped group or sign-in-service role that gives the role.
Idle timeout (minutes)	Security settings	Minutes without activity after which a session ends: 1 to 1440, default 15.
Maximum session length (minutes)	Security settings	The longest a session lasts, however active: 15 to 10080 and not shorter than the idle timeout; default 720.
Reason for the change	Role mapping, Security settings, Sign-in review, user records	Required; kept in the Security audit trail with the change.
IntelliTrack role	Role mapping	Administrators, Recipe, Engineering, Production, Production Supervisor, Quality, Audit or General.
Engineering	Role mapping	Controls, process and manufacturing engineers: maintain the plant and equipment model — the plant hierarchy and locations, processes and routes, attributes and attribute maps, OEE configuration, equipment enforcement and alarm definitions. Includes no production role; Administrators pass every Engineering check. Can be named as a signature policy's signer; never held by a service identity. Shipped group <i>IntelliTrack Engineering</i> .
Production Supervisor	Role mapping	Shift leads and supervisors: everything Production may do, plus releasing, correcting, overriding and authorizing. Includes Production without a second mapping. Can sign release decisions, which with GxP Mode on need a signer who holds it; never held by a service identity. Shipped group <i>IntelliTrack Production Supervisor</i> .
Quality	Role mapping	Quality technicians: close a process audit record with its measurements and verdict. Includes no production role and no production role includes it; Administrators pass every Quality check. Signs a closed audit as shipped (with Administrators); never held by a service identity. Shipped group <i>IntelliTrack Quality</i> .

Field	Where	Meaning
Audit	Role mapping	Reads the audit trail and decides about its findings; changes nothing, including the audit trail's own settings. Held separately from Administrators on purpose — see Chapter 20 .
Type	Role mapping	Windows group; Identity provider application role; Identity provider group (the last two only when an identity provider is configured).
Group, role or value	Role mapping	A Windows group (CORP\Group , or a SID; stored by SID), or the exact value the identity provider sends (capitals matter).
Display name	Role mapping	Optional: how the value reads on the page, for example the group's name when the value is an object id.
Account name suffix	Provider links	Optional: proposes each person's account name with this suffix, instead of their directory user principal name or e-mail.

Related Topics

- [Chapter 1: Getting Started](#)
- [Electronic Signatures & Audit Trail](#) — signing, GxP Mode, the inactivity lock and the audit trail
- [Chapter 5: Recipes](#) — approvals and their signatures
- [Chapter 17: System Configuration](#) — system parameters

Electronic Signatures & Audit Trail

IntelliTrack keeps an **audit trail** of every change to configuration and quality records — who changed what, when, from what to what, and why — and, where your site's quality system asks for it, requires an **electronic signature** before a change takes effect. Together they are how IntelliTrack meets 21 CFR Part 11 and EU GMP Annex 11.

- The **audit trail is always on**. Nothing turns it off, and nobody can edit or delete an entry.
- **Electronic signatures are switched on by GxP Mode** and then required wherever your site's **signature policies** say. With GxP Mode off, no change asks for a signature — except the two acts that are signatures in their own right: signing an audit trail review, and turning GxP Mode on.
- A signature is always given by the person signed in, with a password IntelliTrack checks at that moment. How you sign in (Windows, a user name and password, or your organization's sign-in service — see [Security and Sign-In](#)) decides which password that is, and whether you can sign at all: see [Which password you sign with](#).

Administrators find the pages this chapter describes — the audit trail, its reviews, tamper-evidence verification and the signature settings — on **Configuration** → **Additional Configuration** → **Compliance**, a page with one tile for each.

When you are asked to sign

A form that needs your signature shows an **Electronic signature** panel above its buttons. It appears only when a signature is required for that change, right now — the same page may not ask you next time if your administrator changes the policy.

The panel shows:

Item	What it is
You are about to sign	A plain statement of the change.
Meaning of this signature	What your signature means — for example <i>Approved</i> , <i>Reviewed</i> , <i>Disposition Decision</i> . It is recorded exactly as shown.
Show exactly what is being signed	Opens the full record you are signing: the change and the record's values. Anything not listed there is not part of your signature.
User ID	The account you are signed in to IntelliTrack with (for example <code>CONTOSO\jsmith</code>). You cannot change it: a signature can only be given by the person signed in.
Password	Your Windows password, typed each time — see Which password you sign with .
Reason	Shown when the policy asks for one. Some reasons also need a Comment .

Item	What it is
Comment	Free text kept with the signature. Marked (<i>required</i>) when the meaning demands one (for example <i>Rejected</i> or <i>Deviation Accepted</i>).

To sign

1. Make your change on the form as usual.
2. Read what you are signing (**Show exactly what is being signed**).
3. Choose a **Reason** if one is asked for, and add a **Comment** if it is required.
4. Type your **Password**.
5. Select the form's usual button (**Save, Approve, ...**).

The change and your signature are saved together: if either cannot be saved, neither is.

Your password is used once, to check that you are you. It is never stored, never shown again, and never written to a log. By entering it you sign the record electronically; your signature is the legally binding equivalent of your handwritten signature.

Which password you sign with

Being signed in is not enough to sign: IntelliTrack checks your password again, with Windows, at the moment you sign. Which password that is — and whether you can sign at all — depends on how you signed in. **My access** (select your name in the menu bar) shows it as **Signed in with**; see [Security and Sign-In](#).

Signed in with	User ID in the panel	What happens when you sign
Windows sign-on , or User name and password (Active Directory) , with a company (domain) account	DOMAIN\name	Type your Windows password. Your company's domain checks it, so domain signatures work only while this server's Signature readiness checks pass; when they do not, the panel refuses with the reason.
Windows sign-on with an account of the IntelliTrack server itself (SERVER\name , for example a site's emergency account)	SERVER\name	Type that account's password; the server checks it. Allowed only while System Parameter 115 (Allow Local Account Signatures) is 1 — otherwise the signature is refused.

Signed in with	User ID in the panel	What happens when you sign
Your organization's sign-in service (OpenID Connect)	The account name the service sent — usually an e-mail-style name	You cannot sign in this release. Confirming a signature through the sign-in service is not built yet, and there is no Windows password IntelliTrack can check for this account, so the attempt is refused with <i>Your account cannot give electronic signatures on this server. Contact your administrator.</i> The same applies to everything else that asks for your password: turning GxP Mode on or off, signing an audit trail review, and unlocking the inactivity lock . If your site also offers Sign in with Windows or Sign in with a user name and password , sign out and sign in that way to sign; otherwise ask your administrator. (Only if your site has set the sign-in service to send a Windows-style DOMAIN\name does the first row apply instead.)

Machines never sign. An API client (an ERP, for example) acts under its own service identity, which cannot give a signature: a work-order release, close or quantity change sent through the API that the site's policy says must be signed is refused, with a message to make the change in IntelliTrack, where a person can sign it.

If the signature is refused

Message	What to do
<i>The signature could not be verified. Check your password and try again.</i>	Retype your password. Every attempt is recorded.
<i>Signing is blocked for your account after repeated failed attempts...</i>	Wait the minutes shown, or contact your administrator. IntelliTrack stops checking your password before your domain account would lock, so your Windows sign-in is not affected.
<i>The record changed after this page was displayed...</i>	Someone changed the record while you were reading it. Reload the page, review the record again, and sign.
<i>Select a reason for this change.</i>	Choose a reason in the panel.
<i>A comment is required for this signature.</i>	The meaning or the reason you chose needs a Comment ; add one.
<i>This signature must be given by a user in the ... role.</i>	The policy asks for a signer in a role you do not hold. Ask someone who holds it. On a release decision this is Production Supervisor, and it applies to administrators too; where your site has named Engineering on an engineering change, the same holds for Engineering (see Who may sign: roles).

Message	What to do
<i>You have already signed or approved in this cycle; another person must give the next signature. — or — The person who requested this change cannot approve it; another person must sign.</i>	The change needs a signature from a different person.
<i>The meaning of this signature changed after the page was displayed...</i>	The policy changed while you were reading. Review the change and sign again.
<i>Your account cannot give electronic signatures on this server. Contact your administrator.</i>	The way you signed in cannot sign here — see Which password you sign with .
<i>Signatures cannot be verified on this server right now: ...</i>	The server cannot reach the domain controller the way signatures require. Tell your administrator; the message names the problem.
<i>The directory could not be reached to verify your signature...</i>	Try again shortly; tell your administrator if it persists.
<i>This installation is not bound to a site, so signatures cannot be recorded...</i>	An administrator must set System Parameter 1 (Site Id , the site this installation serves).
<i>This action requires an electronic signature, but the signature policy cannot be read...</i>	Tell your administrator: nothing will be signed until the policy can be read.
<i>Nothing was saved or signed: ...</i>	The database refused the change. Nothing was kept; try again or tell your administrator.

Changes that need more than one person

Some approvals need two or more signatures, sometimes from **different people** and in particular roles. The panel says how many are needed in all (*This change needs 2 signature(s) in all before it takes effect*). The change takes effect when the last one is given. If distinct signers are required, the person who requested the change cannot also approve it. Only approval acts can need more than one signature: a recipe's, material's or BOM's move to Under Review, Approved or Active, and the approval of a control-instance override.

The number of signatures and the distinct-signer rule are fixed when the change is requested: a policy change made while a request waits for approval applies to the next request, not to that one.

Which acts can be signed

Every act below has a signature policy (see [Signature policies](#)). The column **Signed by default** is how IntelliTrack ships:

- **Yes** — the act's policy ships *Required*: as soon as GxP Mode is on, the act is signed (these are the *Tier 1* acts).
- **No** — the act's policy ships *Not required*: it is signed only if your administrator turns its policy on. The **Signature policies** page marks these acts **Tier 2**.
- **Always** — the act is a signature in its own right and is signed even with GxP Mode off.

The meaning, the reason setting and the signer's role are the shipped defaults too. **Signer's role** lists the roles any one of which the signer must hold while GxP Mode is on: the roles of the page that performs the act, so as shipped nobody who may perform an act is refused for their role — except at a release decision (below); — means any role. Your administrator can change any of it — for the whole enterprise, or for one site, area, workcell, process, material group, recipe type or single record — so the panel on your screen is the authority. The chapter in the last column says what each act does.

Record type	Act (as named on Signature policies)	Signed by default	Meaning	Reason	Signer's role	Chapter
Recipe, Material, Bill of materials	Submit for review (Draft to Under Review)	Yes	Reviewed	Optional	Recipe, Administrators	5 , 4 , 6
Recipe, Material, Bill of materials	Approve (Under Review to Approved); Activate (Approved to Active); Reactivate a retired revision, material or bill of materials; Retire (to Inactive, Obsolete or Archived)	Yes	Approved	Optional	Recipe, Administrators	5 , 4 , 6
Recipe, Material, Bill of materials	Return to Draft; Reject a pending status change; Withdraw a pending status change	Yes	Rejected	Optional	Recipe, Administrators	5 , 4 , 6
Attribute	Create, Edit or Delete an attribute; Supersede an attribute; Restore a deleted attribute	Yes	Approved	Optional	Engineering, Administrators	2
Control-instance override	Request an override	Yes	Authored	Required	Production Supervisor, Administrators	5
Control-instance override	Approve an override	Yes	Approved	Optional	Recipe, Administrators	5

Record type	Act (as named on Signature policies)	Signed by default	Meaning	Reason	Signer's role	Chapter
Control-instance override	Reject a pending override	Yes	Rejected	Optional	Recipe, Administrators	5
Control-instance override	Withdraw a pending override	Yes	Rejected	Optional	Production Supervisor, Administrators	5
Control-instance override	Revert an applied override	Yes	Approved	Required	Production, Production Supervisor, Administrators	5
Recipe deviation	Accept a detected deviation	Yes	Deviation Accepted	Optional	Recipe, Production Supervisor, Administrators	5
Recipe deviation	Resolve a deviation	Yes	Verified	Optional	Recipe, Administrators	5
Quarantine incident	Raise a quarantine incident (confirm its staged products into quarantine)	Yes	Disposition Decision	Optional	Production Supervisor, Administrators	10
Quarantine incident	Disposition quarantined products (every acknowledgement type except <i>Contained</i>) — release decision	Yes	Disposition Decision	Optional	Production Supervisor	10
Quarantine incident	Contain quarantined products (acknowledge them as <i>Contained</i>) — protective	No	Disposition Decision	Optional	—	10
Quarantine incident	Roll back a quarantine incident — release decision	Yes	Disposition Decision	Required	Production Supervisor	10
Quarantine incident	Discard a staged quarantine incident	No	Rejected	Optional	Production Supervisor, Administrators	10
Quarantine incident	Edit a quarantine incident	No	Approved	Optional	Administrators	10

Record type	Act (as named on Signature policies)	Signed by default	Meaning	Reason	Signer's role	Chapter
Quarantine incident	Delete a quarantine incident	No	Approved	Required	Administrators	10
Product / lot	Change a product's disposition (a permissive or lateral change, such as releasing it from Hold) — release decision	Yes	Disposition Decision	Optional	Production Supervisor	3
Product / lot	Scrap (delete) a product	Yes	Disposition Decision	Optional	Production Supervisor, Administrators	3
Product / lot	Restrict a product's disposition (such as placing it on Hold) — protective ; Mark a product as rejected — protective	No	Disposition Decision	Optional	—	3
Product / lot	Change a product's pass/fail result (any change other than to reject) — release decision	Yes	Disposition Decision	Required	Production Supervisor	3
Product / lot	Rename a lot or serial number	Yes	Authored	Optional	Production Supervisor, Administrators	3
Product / lot	Correct a committed product record (material, amount, units, expiry, supplier)	Yes	Authored	Required	Production Supervisor, Administrators	3
Product / lot	Move a product to another work order	No	Authored	Optional	Production Supervisor, Administrators	7
Process audit record	Close a process audit record	Yes	Performed	Not asked	Quality, Administrators	9

Record type	Act (as named on Signature policies)	Signed by default	Meaning	Reason	Signer's role	Chapter
Process audit record	Skip a scheduled process audit; Override an audit's time interval; Override an audit's unit interval — each a release decision	Yes	Performed	Required	Production Supervisor	9
Equipment enforcement	Change a machine's enforcement	Yes	Approved	Required	Engineering, Administrators	7
Work order	Release a work order; Close a work order; Change a work order's quantity (a released order, beyond its tolerance)	No	Approved	Optional	Production Supervisor, Administrators	7
Disposition	Create, Edit or Delete a disposition; Restore a deleted disposition	No	Approved	Optional	Administrators	10
Approval status	Edit an approval status	No	Approved	Optional	Administrators	17
System parameter	Change a signature or audit-trail control setting (parameters 111–119)	Yes	Approved	Required	Administrators	below
System parameter	Edit or Delete any other system parameter	No	Approved	Optional	Administrators	17
Signature policy	Create, Edit or Delete a signature policy; Restore a retired signature policy	Yes	Approved	Required	Administrators	below
GxP Mode	Turn GxP Mode on or off	Always	Approved	Required	Administrators	below
Audit trail review	Sign off an audit trail review	Always	Verified	Optional	Audit, Administrators	below
Reason code	Create or Edit a reason code; Retire a reason code; Restore a retired reason code	No	Approved	Optional	Administrators	below
Security role mapping	Change which groups confer a security role	No	Approved	Required	Administrators	19

Release decision marks the seven acts that let stopped or suspect product continue, or waive a quality check. Their Enterprise policies ship with **Production Supervisor** alone as the role the signer must hold, so with GxP Mode on only someone who holds it can sign them — an administrator who does not also hold it cannot (see [Who may sign: roles](#)). Every other act's list includes **Administrators**.

Protective marks the three acts that only stop or flag product: marking it rejected, restricting its disposition, and containing quarantined product. They ship unsigned, so nobody needs a password to pull a bad unit out; a policy may turn a single signature on for them, but can never require a role, a second signer's role, or more than one signature (see [Signature policies](#)).

Always holds as long as the act's policy stays *Required*, as shipped. Unlocking the [inactivity lock](#) also takes your password, but it is not a signature and has no policy: each attempt is kept with the signature attempts.

Reactivate is listed on **Signature policies**, but no screen performs it in this release: a retired recipe revision, material or bill of materials cannot move straight back to Active. It comes back through Draft or Under Review, and those moves are signed as *Return to Draft* and *Submit for review*.

Seeing who signed a record

Every page that shows a signed record has an **Electronic signatures** section listing each signature:

Column	Shows
Signed by	The signer's printed name, and their user ID beneath it.
Date and time	When it was signed, in the time zone it was signed in, and in UTC beneath it.
Meaning	What the signature means.
Action, Reason, Comment	What was done, and why.
Verification	Verified — the signature is intact. The record has changed since it was signed — the record has been edited since; the signature still stands for the version signed. Does not verify — the signature record may have been altered; tell your quality lead. How the password was checked is shown beneath (by the domain, or by the server's local account).

Times here, and on the audit trail and Compliance pages, are always written year-month-day with the time zone offset they were recorded in, and again in UTC — for example `2026-09-18 14:05:00 -04:00` and `2026-09-18 18:05:00 UTC` — whatever date format your site has chosen for other pages, so a record reads the same to every reviewer.

The inactivity lock

If your administrator has turned it on (System Parameter 114, **Application Inactivity Lock**), a browser left alone for the set number of minutes is **locked**: the page is replaced by **This browser is locked**, showing your user ID. Anything the page was doing in the background stops. The lock is not your session ending (see [Security and Sign-In](#)): it does not sign you out, but nobody can use the browser until your password is given.

1. Type your **Password**.
2. Select **Unlock**.

You return to the page you were on. A form you had not submitted is not re-sent; enter it again. The unlock checks your password exactly as a signature does (failures are counted, can block you for a while, and are reported). If you are not the person named, close the browser.

If you signed in through your organization's sign-in service, IntelliTrack cannot check your password here (see [Which password you sign with](#)), so the unlock is refused: close the browser, open IntelliTrack again and sign in.

Reading or typing on a page, and opening another page, count as activity; a page left open that only refreshes its data in the background does not, so it still locks. A dashboard in playback is different: it opens each item as a new page, so it does not lock while it keeps moving — only if one item stays on screen longer than the lock time (see [Dashboards](#)).

Who may read the audit trail: the Audit role

Reading the audit trail needs either the **Administrator** role or the **Audit** role. The Audit role exists so that the two can be held by different people, and it is worth understanding why before deciding who gets it.

A review is only meaningful if the reviewer was not involved in what they are reviewing. IntelliTrack already enforces that between individuals: you cannot sign off a review of a period in which you changed any record in scope, and you cannot decide about a finding that concerns a change you made. What it could not enforce, until this release, was the same thing between *groups of people*. The administrators are the people most able to change a record outside the application — and if they were also the only people who could look for evidence that someone had, the check would be marking its own homework.

The Audit role can: open every audit trail page, run a check for changes made outside IntelliTrack, run the tamper-evidence verification, work and sign off a review, and decide about a finding. It can export any of it.

The Audit role cannot: reach configuration, production, recipes or security; change any system parameter, including the audit trail's own settings. Those settings are shown to a reviewer, read-only, on the workspace, because you need to know how the check is tuned in order to read a finding correctly — and you must not be able to re-tune the check you are reviewing.

Grant it the way every role is granted: **Security** → **Role mapping**, by adding the directory group your quality function belongs to. A person who needs both roles is given both; nothing is taken away from Administrator.

The audit trail workspace (Administrator or Audit role)

Configuration → Additional Configuration → Compliance → Audit trail workspace

Where to start. The page answers four questions before you scroll:

- **Findings needing attention.** How many changes the checks have found that nobody has explained yet, or that somebody escalated. Zero is written as a sentence, not left blank, so that "we looked and everything agreed" can never be mistaken for "nobody has looked".
- **Last check.** When the check for changes made outside IntelliTrack last ran, what it examined and what it found. If none has ever run, the page says exactly that — an empty list of findings is not evidence that there is nothing to find. **Check now** runs one immediately; it runs on its own schedule as well, every System Parameter 124 hours.
- **Review due.** How long since the trail was last reviewed, against the review period (System Parameter 123). It turns amber past the period and red at twice it.
- **Tamper-evidence verification.** The last run and how many problems it found, with **Verify now**.

Below them: the last thirty days of changes and findings as a chart — how you notice that last Tuesday was unusual — the most recent findings, any reviews in progress with how far they have got, and the audit trail's own settings.

When part of the check was not running, a banner says so and says which part, in plain words. A check that could not read the SQL Server audit still checked configuration changes, and the banner tells you which half you are looking at. A banner also appears when the scheduled check has not run for more than twice its interval. Neither can be turned off; both can be dismissed for the session.

Changes made outside IntelliTrack

Everything IntelliTrack changes, it records. The check exists for everything else: an edit made straight in the database with a query tool, by a script, or by another system.

It works by comparing two records of the same change. IntelliTrack writes an audit trail entry; SQL Server itself keeps its own history of the configuration tables, underneath the application, where the application cannot suppress it. A change that has both is a change made through IntelliTrack. A change with only the database's half **did not come through IntelliTrack**, and becomes a finding.

What the check covers. Everything IntelliTrack keeps configuration for, wherever that configuration is written:

Where it is written	What is checked there
The central configuration database	Recipes, materials, bills of material, the plant model (enterprises, sites, areas, work cells, processes, equipment, locations), attributes, routes, approval statuses, dispositions, GTINs, work order event types
This site's database	System parameters, licensing settings, equipment enforcement
The reporting database	OEE fault codes, modes, states, shift times and user-defined fields; dashboards; notification publications, subscriptions and subscriber profiles; the AI provider settings

You do not have to know which is which. The check runs in all of them and the workspace shows the result as one list.

Two things are deliberately left out, and both are visible on the workspace's settings panel rather than hidden: **quarantine criteria**, whose edits IntelliTrack records against their incident rather than against the criterion, so the check cannot pair them up; and **API clients**, which record the time of every API call and would therefore make every API call a finding.

A finding is not an accusation. Most have an innocent explanation — a commissioning data load, a supplier's migration script, an approved correction. What the record needs is for somebody independent to look at each one and write down which it was.

What the check can and cannot see. A direct edit made within a few seconds of a legitimate change to the same record is explained by that change and is not raised; the exact window is System Parameter 120, shown on the workspace. Changes to the production tables the machines write are never versioned — versioning them would slow the line — so they are covered by SQL Server Audit instead, where it is installed. The workspace tells you when it is not.

Working the findings queue (Administrator or Audit role)

Audit trail workspace → Findings

The queue opens on everything still needing attention, oldest first. Each row says when it was detected, what kind of finding it is, which record, which database account made the change, and where it stands. Filter by any of those; the filters travel in the address, so you can send a colleague exactly what you are looking at.

Period narrows the queue to findings detected in the last 7, 30 or 90 days. It is counted back from today rather than being two dates, which matters when you keep the filter: see below.

Saved views. Most people work this queue the same way every week, and a filter you have to rebuild by hand each time is one you stop using — you read whatever the page opens on instead. Set the filters you want, type a name beside **Keep**, and select it: the combination appears as a button above the queue and puts the filter back in one click. Select the small **x** on a view to remove it.

A saved view keeps the period **as you chose it**. "Escalated, last 90 days" saved today still means the last ninety days next month; it does not quietly become a fixed range ending on the afternoon you saved it.

You may keep up to twenty views, and the twenty-first is refused with a message rather than dropped silently. Your views are yours: they are stored against your account, nobody else sees them, and they are not part of the audit trail — they are a convenience, not a record, and nothing you decide rests on one.

Open a finding to see the evidence behind it — the version's period, the entry it was compared with, the statement the database recorded, the account that ran it — and to record your decision:

- **Explained** — this change is accounted for. It needs a reason **and** a reference to an approved change control or work order. "It was fine" is not a decision; "it was change control CC-2026-0412" is, and that is what the record keeps.
- **Escalated** — this needs someone else. It needs a reason.

You cannot decide about a finding that concerns a change you made yourself.

A decision is never replaced. Changing your mind adds a second decision and keeps the first, because "explained, then escalated" is a materially different story from "escalated", and the record should tell the longer one. Every decision ever recorded about a finding is listed on its page with who made it and when.

Searching the audit trail (Administrator or Audit role)

Configuration → Additional Configuration → Compliance → Audit trail

Choose the **Database** first: **This site** holds the changes to this site's own records (products, quarantine incidents, overrides and deviations, process audit records, equipment enforcement, system parameters and the like); **Central configuration** holds the changes to what every site shares (recipes, materials, bills of materials, attributes, the plant model, signature policies, reason codes and the like); **Reporting database** holds the changes to OEE reference data, dashboards, notifications and the AI settings.

Sites is next, and the line above the filters says which you are reading. It opens on **This site**, which is what you almost always want and what makes these counts agree with a review's. **All sites** widens it — useful when you are looking at something written before this installation was bound to a site, or under a previous binding.

Then narrow the search by **Record type**, **Record key**, **Changed by** (the account name), **Operation** (**Created**, **Changed** or **Deleted**), **From** and **To**, or tick **Signed changes only**, and select **Search**.

Each row shows when the change was made, the record, the operation, who made it and in which roles, the reason, how many fields changed, and **Signed** when a signature authorized it. Select the time to open the **Audit trail entry**: every field **Before** and **After**, the comment, the request and the client address, and the signature itself. A date and time reads the way the entry's own time does — 2027-09-10 00:18:08 -04:00, to the second and with the offset it was recorded with — so two values can be compared at a glance. Hold the pointer over it to see it exactly as stored, to a ten-millionth of a second; that is also what **Export (CSV)** writes. The fraction of a second is shown on the page only when the two values are less than a second apart, so a field that changed never reads the same on both sides. A secret — a password,

key or token, the SMTP password among them — shows as *****redacted***** on both sides: the entry says that it changed, and who changed it, but never what it was.

Changed outside IntelliTrack. A record can carry this badge beside its name. It does not mean *this* change was irregular — the entry you are looking at is a perfectly ordinary one, which is exactly the problem: nothing on its row could otherwise tell you that the **same record** also has a change IntelliTrack never recorded. Select the badge to open the finding. A finding somebody has already **explained** against a change control does not badge, because a badge you learn to ignore is worse than no badge. Select the record, or **Whole history** on an entry, to see every change ever made to that record, oldest first, with every signature given on it.

The count above the results is exact on the first page. Page on, and it reads **About *n* entries** — the figure the first page counted, carried forward. Counting a large trail exactly on every page is the slowest thing on this screen and nobody reads the number to that precision, so the page says "about" rather than quoting a figure nothing re-counted. The **Next** and **Previous** buttons do not count pages either: each one starts from the exact entry the page you are on ends at. That is what stops a change written while you are reading from pushing an entry past you unseen, which is the one thing this screen must never do.

The Change History link on a record

You do not have to come to this page to read a record's history. Recipes, materials, bills of materials, the plant model, approval statuses, dispositions and attributes each carry a **Change History** link on their own details page, which opens that record's whole history directly. Sites whose procedures mandate other wording can rename the link with System Parameter 122. A record IntelliTrack does not audit shows no link at all, rather than one leading to an empty page.

Beside the link you may see **changed 3 times since you last looked**. It counts the entries written since the last time **you** opened that record's history, and it is yours alone — the same record is new to one reviewer and familiar to another. Until you have opened the history once there is nothing to count from, so nothing is shown: an indicator reading "0" would tell you that you are up to date with a record you have never read.

Export (CSV) downloads the search — up to 50,000 entries, one line per changed field — for your quality records (cells that a spreadsheet could read as a formula are written as text).

The trail is never recorded in part: a single save that would change more audited records than System Parameter 116 (**Audit Trail Maximum Records Per Save**, 500 unless changed) allows is refused, and nothing is saved. Split the change, or have an administrator raise the limit.

Reviewing the audit trail (Administrator or Audit role)

Configuration → Additional Configuration → Compliance → Audit trail reviews → New review

A periodic review, by someone who was not involved in the changes, is part of GMP (and asked for before batch certification in the EU).

1. Set **From** and **To** (the last seven days unless you change them) and, optionally, one **Record type**, and select **Show the scope**. The page counts the **Entries in scope** — in this site's records and in the central configuration — and lists the most recent of them.
2. Read the entries in scope.
3. Write your **Findings** — what you checked and what you found, or *No findings*. The findings cannot be left empty.
4. Enter your password (and a reason, if asked; the shipped ones are *Periodic review*, *Batch certification* and *Investigation*) and select **Sign the review** (meaning *Verified*). The review is **always** signed, even with GxP Mode off.

You cannot review a period in which you changed any of the records in scope, in either database. **Audit trail reviews** lists every review with its reviewer and roles, scope, period, number of entries and findings; **Signed** opens the review's signature.

Reviewing entry by entry (Administrator or Audit role)

Configuration → Additional Configuration → Compliance → Audit trail reviews → New review → Review entry by entry

The review above is a single attestation over a whole period: you read the entries, write what you found, and sign once. Some sites' procedures ask for more than that — a decision recorded against **each** entry. That is what this is, and a site chooses whichever its procedure calls for. Both end in the same signed review.

Set the period and record type as before, then select **Review entry by entry**. IntelliTrack opens a review and takes you to its worklist.

1. Each entry has two actions. **Reviewed** means you have read it and it is accounted for. **Flag** means it needs following up, and it asks you what is wrong with it — a flag with no reason is not something anybody can act on, so it is refused.
2. The top of the page says how far you have got: "*142 of 310 dispositioned*", and how many you have flagged.
3. Entries **you** changed are marked as your own and have no buttons. A review is by someone not involved in what is being reviewed, so you cannot decide about your own changes; if a period is full of them, it is somebody else's review.

You can stop at any point and come back. The review stays open until it is signed, and the landing page lists it under **Open reviews** with its progress. Nothing is lost, and nothing is signed until you sign it.

Sign off is available only when the review is complete:

- every entry in the period has been dispositioned, and
- no finding detected in that period is still open.

When it is not available, the page says which of the two is in the way and by how much. Close the outstanding findings, or disposition the remaining entries, and it becomes available.

Signing off shows you exactly what you are signing before the password box — the scope, the period, how many entries were in it, how many you dispositioned, how many you flagged, and how many findings were open — and those are the same numbers that go into the signature. Write your findings, enter your password and select **Sign off**; the review joins the list. If the review was signed off in the meantime (from another tab, say), the page says so and signs nothing.

Tamper-evidence verification (Administrator or Audit role)

Configuration → **Additional Configuration** → **Compliance** → **Tamper-evidence verification** → **Verify now**

Checks every signature and every audit-trail entry in both databases against its fingerprint and its place in the chain, and checks that the protections that stop anyone changing or deleting them are still in place. The result, with each problem found, is kept: the page lists the recent runs of **This site's database** and of the **Central configuration database** — when, by whom, how many chains and rows were checked, and **No problems** or the number of problems with their **Details**. Run it after every upgrade, and keep a run with no problems for your validation records.

Administering signatures (Administrator role)

All under **Configuration** → **Additional Configuration** → **Compliance**, one tile per page.

Signature readiness

Shows whether this server can verify signatures made with domain accounts. Five checks — the domain controller's name, DNS records, network ports, clock difference, and a test Kerberos ticket — must all pass; each failing check says how to fix it. **Run the preflight now** repeats them (they also run every 15 minutes). While any fails, domain signatures are refused with the reason shown, and the moment the checks start failing an **Electronic Signature Preflight Failed** notification is raised (see [Notifications](#)). With no domain in use only local-account signatures are possible.

The page also holds the rest of the evidence an installation qualification asks for: the last signature in this site's database and in the central configuration, with how its password was checked (*Kerberos* for a domain signature), the last tamper-evidence run of each, and the settings that govern signing — GxP Mode, local-account signatures, the failure throttle and alert threshold, the inactivity lock, and the site this installation is bound to (a warning appears if it is bound to none: signatures cannot be written until it is).

GxP Mode

The master switch for electronic signatures (System Parameter 110). It belongs to this site — each site's database has its own — while the signature policies are shared by every site. The page says **GxP Mode is**

ON or **GxP Mode is OFF**, lists the signatures given on the switch, and offers **Turn GxP Mode on** or **Turn GxP Mode off** with the signature panel. Turning it **on** or **off** is itself signed, whatever its current state: switching signatures off is signed while they are still on, and switching them on proves this server can take a signature — so GxP Mode can be turned on only by an administrator who can sign (see [Which password you sign with](#)). With it off, policies stay as configured and no signature is asked for anywhere else; turning it back on restores exactly the same behaviour. The audit trail is unaffected either way.

The page warns when GxP Mode is on but the inactivity lock (System Parameter 114) is off, and when domain signatures cannot be verified because the readiness checks are failing. System Parameter 110 cannot be changed on the System Parameters page: opening it there brings you here.

Signature policies

One row per signable act and scope (for example *Recipe — Approve (Under Review to Approved)*, Enterprise), saying whether a **Signature** is required, its **Meaning**, how many **Signers** are needed and whether they must be **Distinct signers**, the roles a signer must hold — **Role**, any one of them will do; *Any role* when none is named (and a **Second signer's role** list for the second and later signatures) — and whether a **Reason** is *Not asked*, *Optional* or *Required*. An exception can also be limited to a period (**Effective from**, **Effective to**). Acts shipped *Not required* carry a **Tier 2** badge; the list of acts and their shipped settings is under [Which acts can be signed](#). A banner says whether GxP Mode is on: while it is off, no row asks for anything.

The rows are listed in the **Policies** card a page at a time (your **Items Per Page** setting), with **Previous**, **Next** and the page numbers beneath. Above them, choose a record type in the list beside the search box (**All record types** by default) to see only its acts, and type in the search box and press Enter to find rows by their record type, act, scope (including an exception's key), meaning, role or **Tier 2** — the same search as every list page, so ***** and **?** work as wildcards. Changing the record type, the search or **Show deleted** goes back to the first page and keeps the others; the page's address holds all of them, so it can be bookmarked or shared.

Every act has an **Enterprise** row, which applies unless a more specific row does; it can be edited but never retired. **Add an exception** creates a row for one **Site**, **Area**, **Workcell**, **Process**, **Material group**, **Recipe type**, or one named object (**Single object** scope, which needs a **Justification**: an exception for a single record is what central policy exists to avoid). The most specific row that applies decides. **Retire** removes an exception; tick **Show deleted** to list retired exceptions, marked **Deleted**, each with **Details** (its settings, read-only) and **Restore**, which brings back that exception with the settings it was retired with. Restore is refused while another row already covers the same act at the same scope.

Edit (or **Add an exception**) opens the row's page, named by its record type and act, in three cards. **Act and scope** shows the record type, the act and the **Scope** — for a new exception, a list of the scopes by name (Site, Area, Workcell, Process, Material group, Recipe type, Single object) — and **Applies to**: the Enterprise row applies *everywhere*, the fallback whenever no exception applies; an exception names its site, area, workcell, process, material group or recipe type, or, for a single object, its key as the audit trail shows it. **Signature and signers** holds **Signature**, **Meaning**, **Reason**, **Signers** with **Distinct signers**, and

the **Role** and **Second signer's role** lists, where several roles can be picked. The page offers only what the act allows: an act one person performs keeps **Signers** at 1; the second signer's roles can be chosen only while two or more signatures are required; and a protective act (below) takes no role. **Exception details**, on an exception only, holds **Effective from**, **Effective to** and **Justification**. The Enterprise row has no effective period — it always applies, so an act is never left without a policy — and saving one with a period is refused.

The three **protective** acts — *Mark a product as rejected*, *Restrict a product's disposition* and *Contain quarantined products* — take one more rule: saving a row for one of them is refused if it names a **Role**, a **Second signer's role**, or more than one signer. They only stop or flag product, so a policy may ask for a signature on them, but configuration can never quietly take them away from the people on the floor.

Which policy applies (a link on the page) answers "does this act need a signature here, and why?": choose a **Record type** and an **Act**, and a place — **Equipment** (which places the act by site, area, workcell and process), a **Material id** (its material group), a **Recipe type id**, or a record's key as the audit trail shows it — and select **Resolve**. It lists every row that could apply, deciding row first, and says whether a signature is required now (GxP Mode included).

Creating, changing, retiring or restoring a policy is itself a signed act while GxP Mode is on, and the row's page shows the signatures given on it. There is no separate policy history to keep: every change to a row is in the audit trail (**Central configuration**, record type **Signature policy — Whole history** shows a row's settings over time), and every signature and every approval request keeps a copy of the policy it was given under, so the rule in force at any past signing can always be shown.

Rows whose **Justification** begins *Migrated exception* were created by the upgrade that introduced central policy, to keep a record whose own approval count differed from its scope's; review them and retire the ones no longer wanted.

Reason codes

The reasons offered in the signature panel, by **Category**. Each act draws its reasons from one category — approvals, overrides, deviations, dispositions, quarantine, change control, audit trail review, or general — so the panel only ever offers reasons that fit what is being signed. **New reason code** adds one: **Category**, **Code** (up to 50 characters, unique within its category), **Name** (what the signer sees), **Description**, **Order**, and **Requires a comment** (the signer must then explain in the **Comment**; the shipped *Other (explain in the comment)* reasons work this way). A reason code is never deleted, because signatures and audit trail entries cite it: **Retire this reason code** takes it off the panel, and signatures that used it keep it. Retired codes are listed only while **Show deleted** is ticked, marked **Deleted**, with **Details** (read-only) and **Restore**, which puts the code back on the panel; a retired code cannot be edited. Codes marked **Shipped** come with IntelliTrack and cannot be retired. Saving or restoring a reason code is signed if your site's policy for reason codes requires it.

Signature attempts

Every attempt to sign or unlock, successful or not, kept centrally for every site: when, the **Signed-in user**, the **User id entered**, what it was **For**, the **Result (Verified or Refused)**, the **Failure**, the **Detail** and the **Client** address. Type part of a user's account name in **Changed by**, or tick **Failures only**, and select **Search**. The **Failure** column names the reason in IntelliTrack's own terms — for example *InvalidCredentials* (a wrong password), *Throttled* (blocked after repeated failures), *PreconditionFailed* (the readiness checks were failing), *MethodNotPermitted* (a server account while local-account signatures are off) or *UnsupportedIdentity* (an account IntelliTrack cannot check a password for, such as one signed in through the organization's sign-in service).

Failures also raise an **Electronic Signature Security Alert** notification — when one person is blocked after repeated failures (System Parameters 111 and 112), when failures across all users reach the alert threshold (113), when the user id submitted is not the person signed in, and when the domain's check of a password did not use Kerberos or proved a different account (see [Notifications](#)). Subscribe your security contact to it, and to **Electronic Signature Preflight Failed**.

Who may sign: roles

Two things decide whether a person can sign an act: the page's own role (to reach a recipe approval at all, for example, a person needs the Recipe or Administrators role), and the roles the act's policy names as its **Role** — any one of them will do. Both are checked against the IntelliTrack roles the person's session holds — the ones the role mapping gives them, whichever way they signed in — at the moment they sign, and the signature records them.

The Administrators exception: release decisions. Administrators pass every production page, and with GxP Mode off they make every production and supervisor change, release decisions included, without a signature. With GxP Mode on, seven acts are different. These are the **release decisions** — the acts that let stopped or suspect product continue, or waive a quality check — and their shipped Enterprise policies name **Production Supervisor** as the signer's **Role**:

Record type	Act	What it releases or waives
Product / lot	Change a product's pass/fail result	A reject changed to a pass, or any other result change except to reject.
Product / lot	Change a product's disposition	Product released from Hold or another restrictive status.
Quarantine incident	Disposition quarantined products	Suspect product given any acknowledgement except <i>Contained</i> .
Quarantine incident	Roll back a quarantine incident	Every part the incident still holds, released at once.
Process audit record	Skip a scheduled process audit	A due audit waived.

Record type	Act	What it releases or waives
Process audit record	Override an audit's time interval	A due audit postponed by time.
Process audit record	Override an audit's unit interval	A due audit postponed by units.

So each must be signed by someone who holds Production Supervisor. An administrator who is not also mapped to that role reaches the page but cannot sign the act: the panel refuses with *This signature must be given by a user in the Production Supervisor role*. The reason is separation of duties: the decision to let product go should be made by the people accountable for the floor, not by whoever administers the system. An administrator who is also a supervisor is simply mapped to both. A site that wants administrators to sign these too can add Administrators to the **Role** of those policies on **Signature policies** (a change that is itself signed). Every other act ships with the roles of the page that performs it, Administrators among them — *Recipe, Administrators* on a recipe approval, for example — so the policy names who signs without refusing anyone the page admits; the column **Signer's role** under [Which acts can be signed](#) lists them all. A signer who holds none of an act's roles is refused with *This signature must be given by a user in one of these roles: ..., naming them*.

Engineering as the signer. The **Engineering** role — held by the engineers who maintain the plant and equipment model — ships on the policies of the engineering acts that are signed: the **Attribute** acts (such as *Create an attribute, Edit an attribute* and *Delete an attribute*) and *Change a machine's enforcement*, as *Engineering, Administrators*, so an engineer or an administrator signs them alike. A regulated site that wants engineering changes signed by an engineer only removes Administrators from those policies: with GxP Mode on, an administrator who does not also hold Engineering then reaches the page but cannot sign, just as with a release decision. The other plant-model pages and a machine's alarm definitions are not signed acts in this release; a change to an alarm's product impact asks for a recorded reason instead (see [Manage alarm definitions](#)).

The signer's role is checked at the moment of signing, in the same step that saves the change and the signature. With GxP Mode off there is no signature, and the page's own role check is what decides.

The role mapping is kept on **Configuration** → **Additional Configuration** → **Security & Administration** → **Security** → **Role mapping** (see [Security and Sign-In](#)); a bookmark to the old **Security roles** page opens it. The mapping applies to every site that shares this installation's configuration, a change reaches signed-in people at their next action, and every change is recorded, with its reason, in the **Security audit trail**. The page will not let you remove the mapping that gives you your own Administrators role. Its changes are signed wherever the *Security role mapping* policy (the act *Change which groups confer a security role*) asks for a signature, which it does not as shipped: **Add mapping** then carries the signature panel, and **Remove** opens a confirmation page that asks for the signature before anything is removed.

SQL Server Audit: the other half of the check (Administrator role)

The comparison above covers everything IntelliTrack keeps history for. Two things are outside it, and SQL Server Audit is what covers both:

- the tables production writes on **every part**, which are never given history because that would slow the line;
- a direct edit made within a few seconds of a legitimate change to the same record, which the comparison explains away (System Parameter 120).

A new installation is asked about it. Setup has a **Change Detection** page offering to configure SQL Server Audit, and it says on the screen what declining costs. Whichever you choose is recorded, so a check months later can tell you *"SQL Server Audit was declined at installation on 22 September 2026"* rather than the unhelpful *"is not running"*.

An existing installation has to be told to turn it on, because that page only appears on a new install. Your IT administrator runs two things on the database server:

1. `Installer\Set-AuditVolumeAcl.ps1 -Path <folder>` — sets the folder's permissions so SQL Server can create audit files and add to them, and **cannot delete them**. Run it again with `-Verify` at any time to check.
2. `Scripts\ConfigureSqlServerAudit.sql` — sets the audit up.

The script refuses to install where the web application and the machine gateway connect as the **same** database login, or where either is a server administrator. Those refusals are the reason it is safe to offer at all: with one shared login the audit cannot tell a machine's writes from anybody else's.

Until it is on, every check reports a degraded result and says why on the workspace. That is deliberate: a check that is not running must never look like a clean one.

Watch the free space. Once the audit is on, SQL Server refuses any write it cannot record. That is safe — the writes it records are, by design, only ones that should not be happening — but the volume still has to have room. The workspace shows the folder and how much is free, and a check reports a degraded result below 1 GB.

How long the audit trail is kept (Administrator role)

The audit trail is never overwritten and never edited. It can be **archived** once it is older than the retention period — **System Parameter 126**, *Audit Trail Retention Years*, seven years unless your site changes it.

Archiving writes a dated copy, counts it, checksums it, and records the count and the checksum before anything else happens, so there is evidence of what was archived and that it was verified. That record is itself permanent and is never pruned.

Two things are worth knowing:

- **A record's history is never partly archived.** A record that is still being changed keeps its whole history, however old the earliest part of it is. Only records untouched for the whole retention period are eligible.
- **Archiving does not delete anything yet.** The copy is made and verified; removing the original rows needs a change to how the tables are stored that has not been made. Your administrator can see what *would* be eligible, and how fast the trail is growing, without changing anything.

System parameters for the audit trail

Shown read-only on the audit trail workspace, and changed on **Configuration → System parameters** by an administrator. None of them turns the audit trail off; there is deliberately no such setting.

#	Name	What it does
120	Audit Trail Match Tolerance	How many seconds apart the database's own record of a change and IntelliTrack's entry for it may be and still count as the same change. Raising it widens the window in which a direct edit is explained by a legitimate one; lowering it too far reports one change as two halves. Five seconds unless changed
121	Audit Trail Baseline	The moment the check starts from. Stamped once, when the release is first installed: changes before it pre-date the audit trail and are never reported
122	Change History Label	What the link to a record's own history is called on detail screens, for sites whose procedures mandate other wording
123	Audit Trail Review Period	How many days a periodic review covers by default, and when the workspace calls a review overdue. Thirty days unless changed
124	Audit Trail Check Interval	How many hours apart the scheduled check runs. Twenty-four unless changed. The workspace warns if no check has run for more than twice this
125	Audit Trail Export Limit	The most entries one export carries. An export of more is refused, naming the count, rather than stopping short without saying so
126	Audit Trail Retention Years	How many years of audit trail are kept before the oldest may be archived

System parameters for signatures

Configuration → Additional Configuration → Security & Administration → System Parameters, parameters 110–116 (the 110–119 band is reserved for signatures and the audit trail); like GxP Mode, they are this site's own settings. Changing 111–116 is signed while GxP Mode is on, with a reason, because an unsigned change to one of them would be a quiet way round every signature it governs; 110 (GxP Mode) is changed only on its own page. Values outside the allowed range are refused.

#	Parameter	Default	Allowed	What it does
110	GxP Mode	0 (off)	0–1	The master switch; see GxP Mode .
111	Signature Failures Before Block	3	1–10	Failed attempts by one person, within the window of 112, after which IntelliTrack stops checking that person's password for a while. Keep it below your domain's lockout threshold, so IntelliTrack stops before Windows locks the account.
112	Signature Failure Window	15 minutes	1–1440	The window over which failures are counted for 111 and 113. A successful signature starts the person's count again.
113	Signature Failure Alert Threshold	10	2–1000	Failures across all users, within the window, that raise a security alert. It alerts only; it never blocks everyone.
114	Application Inactivity Lock	0 (off)	0–240 minutes	Minutes without activity before a browser is locked; see The inactivity lock . 15 or less is recommended wherever GxP Mode is on.
115	Allow Local Account Signatures	0 (no)	0–1	Whether an account of the IntelliTrack server itself (SERVER\name) may sign. Needed on a server that is not joined to a domain, or to sign with an emergency account while the domain is down.
116	Audit Trail Maximum Records Per Save	500	25–10000	The most audited records one save may change; a larger save is refused rather than recorded in part.

Chapter 21: Validation Documentation

Overview

Validation Documentation helps you produce the documents a regulated customer needs to validate an IntelliTrack installation, without writing them from scratch.

- The *validation library* holds FDS's standard, reusable validation tests: one or more for every IntelliTrack function, each with its steps, its expected results and bracketed parameters (equipment names, serial numbers, recipe names) that you fill in for a project. The library also holds a *standard function description* for every function: the approved "shall" statements an FRS is built from.
- The **Document Library** lets you find, preview and download those tests as editable Word documents. No project or URS is needed.
- The **Package Generator** takes a customer's *User Requirements Specification (URS)* and, in one pass, generates a validation package: the *IQ, OQ and PQ* protocols, the *traceability matrix* and, if you ask for it, a *Functional Requirements Specification (FRS)*, in one ZIP.

The generated documents are AI-selected drafts. The AI reads the URS and selects the tests; IntelliTrack assembles them from the library. Your validation team reviews, completes and approves the documents in Word, under your own document control and quality system. Nothing is signed, executed or marked passed, and nothing is saved in IntelliTrack: you download the package and work on it outside the application.

The library is included in IntelliTrack: each release carries the library built for it, so there is nothing to install and it works the same on every installation. It does not need the IntelliTrack source code. Between releases, FDS issues *library updates*: signed *library baselines* that take the included library's place (see [Library updates](#)).

Before You Begin

- **Who can use it.** The **Engineering** and **Administrator** roles use the Document Library, the Coverage Register and the Package Generator. No separate license or feature entitlement is needed.
- **The library is included in IntelliTrack.** If a library baseline installed on the server is not signed by FDS, built for another IntelliTrack version, or altered, the pages say **Validation Documentation is unavailable on this server**. Administrators see the reason on the same page. The rest of IntelliTrack is unaffected.
- **AI is needed for the Package Generator only.** It uses the AI connection your administrator assigns to the **Validation Package Generator** workload under **Configuration → AI Settings → Routing** ([Chapter 14](#)). Without one, the Package Generator page says why and links to the AI settings. The Document Library works without AI.

- **Every test has a revision, and the latest is current.** The library lists each test at its latest revision; its earlier revisions stay available from the test's page.

How to Get Here

Open **Configuration** → **Additional Configuration** → **Validation Documentation** (the tile is shown to Engineering and Administrator users). The page has a tile for each:

- **Document Library:** search, preview and download tests.
- **Coverage Register:** each function and the tests and steps that demonstrate it.
- **Package Generator:** generate a validation package from a URS.

Every Validation Documentation page shows the library baseline in use: its identifier, release date and IntelliTrack version, with a link to its **Change Notice**.

Find a test

1. Open **Configuration** → **Additional Configuration** → **Validation Documentation** → **Document Library**.
2. Type in **Search the library** and select **Search**. You can type:
 - words: every word must match, and a word matches from its beginning (*quarant* finds *Quarantine*); put a phrase in quotation marks;
 - a test ID (*ITV-RCP-010*), an area (*ITV-RCP*), a function ID (*FN-TRK-116*) or a step (*ITV-RCP-010/S10*), which opens the test at that step.
3. Narrow the list with **Filters: Area, Test type, Interfaces exercised, Parameter groups, Roles exercised, Signed acts, Supported IntelliTrack version** and **Changed in this baseline**. Values within a filter are combined with OR; different filters with AND. Each value shows how many results it would give.
4. Choose **Sort by** and **Group by area** if you like. With a search, results are in relevance order; each result shows where it matched (**Matched in:**), with links to the steps.

Every criterion is in the page address: bookmark or share it to come back to the same list. When nothing matches, the page lists the active criteria with **Remove this criterion** beside each; the search is never widened silently. A test that declares no supported IntelliTrack version is listed under **Compatibility not stated** when you filter by version.

Preview and download a test

Select a test's ID or title to open its read-only preview: the same content as the Word document, rendered from the same definition. The preview shows the test's official marking: its revision, revision date, change class and the library baseline. In the parameters table, **Show in steps** highlights every step where that

parameter is entered. **Revisions of this test** lists its revisions, newest first; an earlier revision says at the top that the library holds a later one.

Select **Download Word document** for the master document. It is an ordinary Word document: no editing restrictions, named styles, and every parameter and execution field blank. From the library landing page, finding and downloading a test takes three actions: search, open, download. To use your own template for the download, see [Document templates](#).

Download several tests

1. In the result list, tick **Select** for each test (or **Add to batch download** on a test's page) and select **Download selected**.
2. The **Batch Download** page lists the selected tests. If a selected test needs another test first, **Prerequisite tests not in your selection** lists it with **Required by** and **Reason**. Prerequisites are never added for you: choose **Add to selection** for one, or **Add all prerequisites**.
3. Select **Download ZIP** to download one Word document per test.

Download with values

If you already know some of a test's parameter values (user IDs, server names, recipe or material names), you can enter them before you download, and the documents arrive with those values in place. The master download is unchanged.

1. On a test's page, or on the **Batch Download** page, select **Download with values**. The page lists the selected tests and a field for each parameter that is entered before execution:
 - **Site values** come first. A value that belongs to the site, such as a user ID, a server or a signature reason, is one field however many of the selected tests use it; the field names the tests and steps that use it.
 - Then each test has its own section with its own fields.
 - The parameters listed under **Recorded during execution** have no field: the tester records them while executing the test.

Each field shows its label, its key, its format or allowed values, where the value usually comes from, and the steps where it is entered. A field with allowed values is a list; **(blank)** leaves it empty.

2. Fill in the values you know. Every field is optional: a field left blank stays blank in the document, to be completed in Word. Spaces before and after a value are removed; everything else (capitals, leading zeros, spaces inside) is kept as you typed it. Enter numbers and dates the way you normally write them in your language (in German, `1,5` and `05.10.2026`); a date can also be written `2026-10-05`, or with a time `2026-10-05 14:30`.
3. Under **Documents**, optionally choose a **Word template (optional)** (see [Document templates](#)), then select **Download documents**. One test downloads as a Word document, several as a ZIP.

Every value is checked before anything is downloaded. If one is not accepted, the page says **The values were not accepted, and nothing was downloaded. Correct the fields listed below.**, lists each problem with its test, key and rule, and shows each one beside its field. Everything you entered stays in the form. A value is refused when it contains a tab or a line break, is longer than 1,024 characters, is not one of the allowed values, is not a number (or a whole number) where one is asked for, is not a date where one is asked for, or is not a full `http` or `https` address where an address is asked for. Some values must also differ from others in the same test: a second approver, for example, must not be the first approver or the requester. User IDs are compared ignoring capitals. The test's parameters table says which values must differ (*Differs from:*).

Nothing you enter is saved in IntelliTrack: the values exist only while the page sends them and the documents come back, and the browser is told to keep no copy. A template file is not kept between submissions either: choose it again before you download the documents. To use the values again, download them as a parameter set (see [Parameter sets](#)).

Site, Test and Instance values

Every parameter entered before execution has a scope, which says how widely one value applies:

- **Site.** One value for the site, the same in every test that uses it: user IDs, servers, signature reasons. It is entered once, under **Site values**.
- **Test.** One value for a test, the same in every instance of it in a package: a recipe, a material, a product.
- **Instance.** One value for one instance of a test: a serial number that can be used only once, for example. In a download with values the document is one instance, so you enter it with the test's own values; its field says "One value per test instance." The Package Generator leaves Instance fields blank (see [Values from a parameter set](#)).

Parameter sets

A *parameter set* is a small file (JSON) holding the values of the form. Keep one to use the same values again: with the next download, or with a Package Generator run. IntelliTrack never stores one.

- **Download parameter set** saves the values in the form as they are now, leaving out the blank ones, in a file named `Parameter Set <yyyy-MM-dd>.json`. The values are sorted by test ID and key, so two sets compare cleanly line by line. Numbers and dates are stored in one form whatever your language: `1.5`, `2026-10-05`.
- To use a set, choose it under **Parameter set file (.json)** and select **Upload parameter set**. The fields are filled in from it: check them, then download.
- Entries of the set that do not apply to the tests on the page are listed under **Not applied**, each with the reason, and skipped: a key no selected test uses, a test that is not selected, a value under the wrong scope, or a parameter that is recorded during execution. The rest of the set is applied; this is not an error.

- A set made with another library baseline is accepted: its values are checked against the baseline installed now, and the page names both.
- A file that is not a parameter set, is damaged, is larger than 1 MB, holds more than 10,000 values or a value longer than 1,024 characters is refused with the reason, and the form keeps what you had entered.

A parameter set holds your values in plain text. Keep it as you keep your other validation records.

The parameterized document

A document downloaded with at least one value is a *parameterized instance* of the test, not the master:

- its title page says "Parameterized instance I01. Contains user-supplied parameter values. Review before approval.";
- its header shows the instance ID after the revision, for example `ITV-MAT-030 Rev A / I01` ;
- its parameters table has a *Value* column with each value supplied;
- its document properties record where the values came from (`Entered` , or `Uploaded:` and the parameter set's file name) and how many were applied.

The values are ordinary text: you can change any of them in Word. A download in which every field was left blank is the master document, exactly as **Download Word document** gives it.

See what the library covers

Configuration → **Additional Configuration** → **Validation Documentation** → **Coverage Register** (the **Function-to-Test Coverage Register**) lists every function of the library baseline with the tests and steps that demonstrate it. A function with no test is listed as a **Gap**; one that is deliberately not tested is **Excluded**, with its reason; one the product does not implement is **Not implemented**. Filter by **Area** and **Status**.

Generate a validation package

Everything happens on one page, **Configuration** → **Additional Configuration** → **Validation Documentation** → **Package Generator**:

1. Under **Generate a validation package**, choose the **User requirements specification (URS)**: a PDF or Word (`.docx`) file. Optionally choose a **Word template (optional)** (see [Document templates](#)) and a **Parameter set (optional)** (see [Values from a parameter set](#)), and tick **Include FRS** if you want an FRS in the package. Select **Generate package**. The files are checked first: a password-protected, damaged or unsupported file, or a URS with no text to read, is refused with the reason, and nothing runs.
2. While the run is in progress the page shows where it is: "Assessing n of N requirements.", then "Generating documents." It updates by itself, and you may leave it and come back. **Cancel** stops

the run; nothing of it is kept. When the server is already running other packages, the page shows your place in the queue.

3. When the package is ready, select **Download** to get the ZIP. The page summarizes it: tests per protocol, gaps, requirements not assessed, unreadable pages, open items, and the template applied.

The ZIP can be downloaded again for 30 minutes after it is generated. Then it is discarded, together with the URS and the template: nothing is saved in IntelliTrack. If the application restarts during a run, the run is gone and the page says so: upload the URS again.

A long URS takes a while: the requirements go to the AI a few at a time, and an answer the AI could not finish within its length limit is asked for again in smaller parts, so no requirement is dropped because an answer was cut short.

If the AI provider did not answer for some requirements, or refused the request (for example because the provider account has run out of credit), the package is still generated: those requirements are marked **Not assessed** in the matrix, and the page states how many there are and why, in the provider's own words. Requirements not assessed for the same reason are one open item that names them all, for example `R209-R560`. A refusal needs fixing first, at the provider or under **Configuration** → **AI Settings**. **Run again** starts a new run with the same URS and template.

Values from a parameter set

To have the package's documents carry values you already know, choose a parameter set (see [Parameter sets](#)) under **Parameter set (optional)** when you submit the URS. Every value in it is checked with the URS and the template, before anything is sent to the AI. If a value is refused, nothing runs: the page lists each refused value with its test, key and rule, for you to correct in the file (or in **Download with values**) and submit again.

After the AI has chosen the tests, the values are applied:

- each Site value goes to every selected test that uses its key;
- each Test value goes to every instance of its test in the package;
- Instance values are not applied: instances exist only once the tests are chosen, so those fields stay blank for you to complete in Word.

The values are never sent to the AI provider, and they have no effect on which tests are chosen. When the package is ready, the page shows under **Parameter set** the number of values applied, counted per test instance, and lists under **Not applied** every entry that was not, grouped by reason: **Unknown key**, **Test not in the package**, **Wrong scope**, **Recorded during execution**, or **Instance value, left blank by the generator**. In the package, every document's title page names the set file (**Parameter Set**), and the traceability matrix's **Values Supplied** column shows, for each test instance, how many of its fields were filled, for example `3 of 11`. The set is discarded with the run, like the URS.

How the tests are chosen

- **IQ.** The standard installation tests are always included. Tests for optional components (the Event Engine, the ERP connector, each notification channel) are included when the URS names the component, and when the URS does not make it clear.
- **OQ.** The AI splits each URS requirement into obligations and selects the operational and integration tests that cover them. Every test it names is checked against the library; anything that does not check out is rejected and listed in the open items.
- **PQ.** Scenarios for several lots produced end to end, and for production counts and reports reconciled against physical counts, are always included. Scenarios for a sustained production rate, recovery from an interruption, and data-integrity spot checks are included when the URS states such a requirement, and when it is unclear.
- **Prerequisites.** Every prerequisite of a selected test is included, in its own protocol. A test whose prerequisite is in another protocol refers to it by test ID, protocol and section.
- **Open questions.** A question the URS does not settle keeps the test in the package and is listed in the open items, for you to decide.
- **No test twice in a protocol.** Each test appears once in its protocol. A test of the equipment interface is executed at every equipment module the URS names: its Test Procedure lists the modules, with a row to record each module's execution (the first on the protocol's own pages, each further module on an attached copy). A module the URS names two ways, for example by its name and by its abbreviation, is listed once.

Every step of a test is one action with one result, so a failed step says exactly what failed. Expected results state what the system does, not the wording of its pages: a message is described by its content or a short phrase from it, so a change of wording does not fail a test.

What the package contains

Document	What it holds
IQ, OQ and PQ protocols	One combined protocol per phase: approval and revision history, table of contents, introduction (purpose, scope, the URS's own system description, references, definitions), execution requirements, signature log, the tests in order with installation and prerequisites first, and the attachments log, deviation log and test index as appendices.
FAT and SAT protocols	The Factory Acceptance Test (every OQ test, run at FDS before the system ships) and the Site Acceptance Test (the IQ tests, then the integration tests, run on site against your systems), with the same structure as the protocols above. Every package has both.
Traceability matrix	Each URS requirement and obligation with its responsibility, the tests and steps that cover it, its coverage, and the outcomes that do not apply with their reason; the coverage summary, the test instances, and the open items.

Document	What it holds
FRS (with Include FRS)	The functional requirements by area, each tracing to the URS and to the standard function descriptions it comes from; requirements outside IntelliTrack's scope; open items. It states intended function only.

The IQ, OQ and PQ protocols

- IQ verifies that IntelliTrack is installed as specified. Product facts — versions, file hashes, database packages and minimum prerequisites — are printed as expected results from the release; site facts (server, instance and group names) are parameters for you to fill in.
- OQ verifies that the functions the URS needs work as specified.
- PQ demonstrates the system in routine production. Your organization owns PQ: you set its acceptance criteria, rates and periods — every one is a blank parameter — and you execute it in production. FDS does not set PQ acceptance criteria.

Every Pass/Fail checkbox is unchecked and every execution field blank. Each document's revision is "Draft", and its title page and header say "Draft — review before approval": your document control assigns the revision when you approve it. The title page also identifies what produced the package: the source URS, the IntelliTrack version, the library baseline, when it was generated, and, under **Test Selection Proposed With**, the AI provider and model and what the model proposed. The test steps and expected results are the library's own, unchanged. The generator and the model are also recorded in each document's properties (`IntelliTrack.GeneratedWith`), so a document stays marked as produced with AI wherever it is filed. Two runs of the same URS may select somewhat differently.

The header of every page starts with your company's logo when an Administrator has uploaded one under **Configuration** → **Branding**, and with the Factory Data Systems logo otherwise. The same applies to tests downloaded from the Document Library. A Word template you upload keeps its own header.

The footer of every page starts with your company name, as set under **Configuration** → **Branding**, or "Factory Data Systems, LLC" when none is set, followed by the project and the library baseline. The project is the system your URS names on its title page, such as "Fill Line 3"; a title that names only the kind of document, such as "User Requirements Specification", is passed over, and when the URS names no system its document ID stands in. The footer carries the copyright notice under that line: Factory Data Systems' notice on the IntelliTrack validation test content always, and, when a company name is set, your company's notice on the project content in front of it. The first page carries the same notice, so a document in a template with its own footer has it too.

The FAT and SAT protocols

- The FAT holds every test of the OQ protocol and is executed at FDS before the system ships. Its introduction has fields for the FAT location and for the simulator or test system used for each site

system that is not available at FDS. A test whose prerequisite is an IQ installation test needs that condition met on the FAT system before the FAT starts.

- The SAT holds the IQ tests, then the integration tests of the OQ protocol, and is executed on site, after the FAT, against your real systems. Its introduction has a field for the system under test: the system being qualified, before it is released for production use.
- A test keeps its instance ID in every protocol that holds it, so a result recorded in the FAT or SAT can be found from the IQ or OQ. Each protocol numbers its own sections and attachments, and a test's references to its own sections ("Execute the steps in Section 4.10") use the protocol's numbers. Where your quality system allows, the IQ and OQ may refer to FAT and SAT results instead of repeating a test; the protocol approvers decide.

Each protocol's table of contents lists its sections, each a link to the section. Word does not ask to update anything when the document opens; after you have edited the protocol, right-click the table and select **Update Field** to add the page numbers.

Identifiers

Every document of the package explains, near its start, the identifiers it uses and the area codes inside them:

Identifier	Refers to
R005 , R005.2	A requirement of your URS, numbered in the order the URS states them (its own ID is shown beside it), and one obligation of it.
FR-AUD-130	A requirement of the FRS.
FN-AUD-010 Rev A	A standard IntelliTrack function description, at that revision. The FRS names the function wherever it cites one and lists every one it cites in its Appendix B; each test lists the functions it demonstrates under Functions Covered .
ITV-AUD-010 Rev A	A standard test of the library; the protocol's test index gives its section.
I20 , S05 , E3	A test instance (one execution of a test, listed in the matrix), a step, and an item of evidence.

The letters in the middle are the catalog area, for example **AUD** for audit trail and records or **INS** for installation and configuration.

The traceability matrix

The matrix shows, for each URS requirement, how it is met and where it is tested (every the coverage: complete, partial, a gap, external evidence, or not assessed. With the FRS it also shows the FRS IDs and an FRS-to-test view. Its open items list everything that needs a decision or evidence from someone, with the proposed responsible party.

The FRS

Ticking **Include FRS** adds the FRS. Its statements come only from the library's function descriptions. Values the URS does not give are visible placeholders for you to complete, and requirements IntelliTrack does not meet are listed as outside its scope. Appendix B lists every function description the FRS cites, with its name, its criticality, the FRS requirements taken from it and the test steps of the package that demonstrate it.

Review and complete the package in Word

The package is yours to review and change in Word before approval:

- **Complete the parameter fields.** Every bracketed field — equipment, serial numbers, recipes, users, and in PQ every acceptance criterion — is blank, unless a parameter set supplied its value. Fill each one in where it appears in the steps; each test's parameters table lists them.
- **Delete a test.** To remove a test the package does not need, delete its section from the protocol, its row in the protocol's test index (Appendix C) and its entries in the traceability matrix, and record the reason in your review.
- **Add a test from the Document Library.** Download it with **Download Word document**, copy its content into the protocol as a new section, add a row to the test index, and add it to the traceability matrix for the requirements it covers.
- **Resolve the open items.** Each names the requirement, the reason and the proposed party.

Document templates

By default every Word document uses the built-in FDS template. To give the documents your organization's look, choose a **Word template (optional)** with a download from the Document Library or with a Package Generator run. The template is used for that download or run only; it is never stored.

Download the built-in FDS template gives you a `.dotx` to start your own from.

What a template can change

A template can supply paragraph, character and table styles; fonts and colors; page size and margins; headers and footers, including logos; and a cover page and an approval page. In its headers, footers, cover page and approval page it can place document fields as Word content controls tagged with the field name: `DocumentTitle`, `DocumentId`, `Revision`, `DocumentType`, `ProjectName` (the project, as in the footer) and `GeneratedDate`. A template defines portrait pages; the landscape pages (the test procedures, the matrix) are derived from its page size and margins.

What a template cannot change

A template changes appearance only, never the content. The text, steps, parameters, checkboxes, tables, section order and required sections are the same whichever template you use. Any other content in the

template's body is ignored.

When a template is refused

The template is checked as soon as you choose it. A template with macros, password protection, ActiveX controls, embedded objects or external links is refused with the reason, and so is one whose landscape page is too narrow for the procedure table (the message states the minimum). An accepted template's report lists the styles it does not restyle: those keep the built-in definition. The same check runs again when you download or generate.

Library updates

Between IntelliTrack releases, FDS may issue a library update for the current release and the previous one. An update is a complete, signed baseline for exactly one IntelliTrack version, delivered as a small program with its Change Notice.

Install a library update

An administrator runs the update program (`IntelliTrack.Validation.Update.exe`) as an administrator on every IntelliTrack web server. It checks that the baseline is signed by FDS, built for the installed IntelliTrack version and newer than the installed one, installs it, and keeps the baseline it replaces. The Document Library shows the new baseline within a minute, without a restart. Running the program with `-rollback` restores the kept baseline. An update that is unsigned, altered, for another version or older than the installed baseline is refused with the reason, and the installed baseline is unchanged.

Read the Change Notice

Select **Change Notice** beside the baseline at the top of any Validation Documentation page. It lists every test the update adds, revises or removes, with its old and new revision, its change class, the reason and the affected functions; unchanged tests are not listed. The change classes are:

- **Editorial**: wording or presentation only; no effect on execution or results.
- **Correction**: steps, expected results or acceptance criteria corrected; protocols executed with the earlier revision should be reviewed.
- **Product**: follows a change in product behavior; applies only to the versions stated.

In the Document Library, the **Changed in this baseline** filter shows the changed tests by change class.

Compare it with your protocols

IntelliTrack stores no packages, so it cannot tell you which of your protocols an update affects. Each protocol's test index (Appendix C) records the revision of every test it used: compare it with the Change Notice, and review any protocol that used a revision the notice lists as a **Correction**.

Basis for the document content

This section lists what the form and content of the generated documents rest on: for each feature, the regulation, guidance or standard it follows, or, where there is none, that it is a Factory Data Systems design decision. It lets a reviewer or an inspector trace a feature of the documents to its source. It is not legal advice: your quality unit decides which of these references apply to your products and markets. Guidance documents are not binding, and two of the references below were drafts when this section was checked (October 2026): EU GMP Annex 22 and FDA's guidance on AI for regulatory decision-making. Check their current status.

The documents script every selected test in full. FDA's Computer Software Assurance guidance (Section V.A(4)) and GAMP 5 scale the formality of testing to risk: when your validation team reviews the package, it decides where less formal testing is enough.

What each feature rests on

Feature of the documents	Basis
Every document is a draft: its revision is "Draft", its header and title page say "Draft — review before approval", and it is approved and given its revision under your own document control.	EU GMP Annex 15 §2.2 (qualification and validation documents are approved and authorised by appropriate personnel, as the quality system defines) and §2.6 (protocols supplied by a third party providing validation services are confirmed suitable by the manufacturer's personnel before approval, and may be supplemented). EU GMP Chapter 4 §4.3 (documents are approved, signed and dated, uniquely identifiable, with a defined effective date). FDA CSA guidance Section V.A(6) (review and approval, with a signature and date where needed). IntelliTrack EULA §8 (the licensee validates the software's outputs).
The title page identifies what produced the document: the source URS, the IntelliTrack version, the library baseline, the generation date and time, and the AI model with what it proposed.	EU GMP Chapter 4 §4.3 (uniquely identifiable). FDA Data Integrity Q&A, question 1.a (data are attributable and accurate: ALCOA). Draft EU GMP Annex 22 §1 (a model with a probabilistic output may not give identical outputs for identical inputs), which is why two runs of the same URS can select differently and why the page records what produced each package.
The AI's part is stated once, on the title page, and what the model proposed is labelled as proposed (the matrix's Proposed Rationale / Notes and Proposed Responsibility columns).	Draft EU GMP Annex 22 §1 (generative AI and large language models are not used in critical GMP applications; in non-critical applications, qualified personnel are responsible for the suitability of the outputs: a human in the loop). FDA's draft AI guidance, Section II (it does not address AI used for operational efficiencies, such as drafting, that do not affect patient safety, drug quality or the reliability of study results). GAMP 5 Second Edition, Appendix D11 (artificial intelligence and machine learning).

Feature of the documents	Basis
Each package document records the generator, its version and the AI model in its document properties, under IntelliTrack.GeneratedWith.	EU AI Act Article 50(2) (text generated by an AI system is marked in a machine-readable format and detectable as artificially generated), applying from 2 August 2026 (Article 113).
IQ, OQ and PQ protocols, each holding its phase's tests; OQ tests include limits, invalid entries and error handling where a function has them.	EU GMP Annex 15 §3.8 to §3.14 (what IQ, OQ and PQ include; OQ tests upper and lower operating limits and worst-case conditions, §3.11) and §2.5 (qualification documents may be combined). EU GMP Annex 11 §4.7 (test scenarios consider parameter limits, data limits and error handling).
The PQ's acceptance criteria, rates and periods are blank parameters that you set.	EU GMP Annex 15 §2.4 (protocols define the critical parameters and their acceptance criteria) and §3.14 (PQ uses production materials under normal operating conditions, which only the manufacturer knows). Leaving them blank rather than proposing values is an FDS design decision.
The FAT protocol holds the OQ tests executed at FDS before the system ships; the SAT protocol holds the IQ and integration tests executed on site.	EU GMP Annex 15 §3.4 to §3.7 (evaluation at the vendor before delivery; a FAT supplemented by a SAT on receipt; a test performed at the FAT need not be repeated at IQ or OQ where transport and installation are shown not to affect it). GAMP 5 Second Edition (leveraging the supplier's knowledge, experience and documentation). FDA CSA guidance Section V.A(5) (leveraging vendor evaluation and validation records). ASTM E2500-20 (use of supplier documentation in verification).
A test appears once in each protocol; the FAT and SAT repeat a phase's tests.	FDS rule: no regulation requires it; it removes duplicate executions that add no assurance. Whether a test passed at the FAT is executed again on site is your decision, which EU GMP Annex 15 §3.6 allows you to justify.
Each step has a Step ID, Instruction, Expected Result, Actual Result / Recorded Data, Status, Initials and Date; expected results are written before execution; Pass and Fail are checkboxes; each protocol has a deviation log.	EU GMP Annex 15 §2.4 (pre-defined acceptance criteria), §2.7 and §2.8 (changes and results that fail the criteria are recorded as deviations and investigated). FDA CSA guidance Section V.A(6) (the record describes the testing, the issues found, and who tested and when). EU GMP Chapter 4 §4.6 (space for entries), §4.7 (entries clear, legible and indelible) and §4.8 (records made at the time each action is taken).
One action and one result per step; an expected result checks behavior, not page wording.	FDS test-authoring rule. It applies EU GMP Chapter 4 §4.3 (unambiguous contents) and §4.4 (easy to check): a step with one action has one result to record against it.
Every execution field, checkbox and signature is blank; the generator never records a result.	EU GMP Chapter 4 §4.8 (records are made or completed at the time each action is taken). FDA Data Integrity Q&A, question 1.a (contemporaneously recorded, original).

Feature of the documents	Basis
The traceability matrix traces each URS requirement and obligation to the test, instance and steps that demonstrate it.	EU GMP Annex 11 §4.4 (user requirements are traceable throughout the life cycle). EU GMP Annex 15 §3.2 (the URS is a point of reference throughout the validation life cycle). FDA CSA guidance Section V.A(4) (traceability as part of robust scripted testing).
The FRS.	EU GMP Annex 15 §3.2 (the specification is defined in a URS and/or a functional specification).
Each document explains its identifiers and area codes; the protocols and the FRS also list the other documents of the package.	EU GMP Annex 15 §2.3 (the inter-relationship between documents in a validation project is clearly defined). EU GMP Chapter 4 §4.3 (unambiguous contents).
The copyright notice: Factory Data Systems' on the validation test content, always, and your company's on the project content when a company name is set.	17 U.S.C. §401(b) (the form of a notice: ©, the year, the owner) and §401(d) (with a notice, an infringer cannot claim innocent infringement to reduce damages). IntelliTrack EULA §1(A) (the software includes its templates and documentation), §4 (proprietary notices may not be removed), §5 (Factory Data Systems owns the software) and §6 (you own your data).
IntelliTrack™ carries ™, not ®.	IntelliTrack EULA §5 (IntelliTrack™ and the Factory Data Systems logo are trademarks of Factory Data Systems, LLC). 15 U.S.C. §1111 (® is the notice of a registered mark).
A Word template changes appearance only; the steps and expected results are the library's, unchanged.	EU GMP Chapter 4 §4.2 (reproducing working documents from master documents does not introduce errors).
Your company's logo and name in the header and footer, the table of contents written in place, and the page layout.	FDS design decisions; no external requirement.

References

- European Commission, EudraLex Volume 4, EU Guidelines for Good Manufacturing Practice, Chapter 4: Documentation. In operation from 30 June 2011.
https://health.ec.europa.eu/system/files/2016-11/chapter4_01-2011_en_0.pdf
- European Commission, EudraLex Volume 4, Annex 11: Computerised Systems, revision 1. In operation from 30 June 2011. A revised draft was published for consultation on 7 July 2025.
https://health.ec.europa.eu/system/files/2016-11/annex11_01-2011_en_0.pdf
- European Commission, EudraLex Volume 4, Annex 15: Qualification and Validation, 30 March 2015. In operation from 1 October 2015. https://health.ec.europa.eu/document/download/7c6c5b3c-4902-46ea-b7ab-7608682fb68d_en
- European Commission, EudraLex Volume 4, Annex 22: Artificial Intelligence. Draft for public consultation, 7 July to 7 October 2025. <https://health.ec.europa.eu/consultations/stakeholders->

[consultation-eudralex-volume-4-good-manufacturing-practice-guidelines-chapter-4-annex_en](#)

- Regulation (EU) 2024/1689 of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), OJ L, 12 July 2024: Article 50 and Article 113. <https://eur-lex.europa.eu/eli/reg/2024/1689/oj>
- U.S. FDA, Computer Software Assurance for Production and Quality Management System Software, guidance (CDRH, CBER), 3 February 2026. It supersedes the version of 24 September 2025, and Section 6 of FDA's General Principles of Software Validation. <https://www.fda.gov/media/188844/download>
- U.S. FDA, Considerations for the Use of Artificial Intelligence to Support Regulatory Decision-Making for Drug and Biological Products, draft guidance, January 2025. <https://www.fda.gov/media/184830/download>
- U.S. FDA, Data Integrity and Compliance With Drug CGMP: Questions and Answers, guidance, December 2018. <https://www.fda.gov/media/119267/download>
- ISPE, GAMP 5: A Risk-Based Approach to Compliant GxP Computerized Systems, Second Edition, July 2022. Available from ISPE.
- ASTM E2500-20, Standard Guide for Specification, Design, and Verification of Pharmaceutical and Biopharmaceutical Manufacturing Systems and Equipment, ASTM International. Available from ASTM.
- 17 U.S.C. §401, Notice of copyright: visually perceptible copies. <https://www.law.cornell.edu/uscode/text/17/401>
- 15 U.S.C. §1111, Notice of registration; display with mark. <https://www.law.cornell.edu/uscode/text/15/1111>
- Factory Data Systems, LLC, IntelliTrack End User License Agreement, Rev. 1, effective 20 August 2026.

Field Reference

Package Generator:

Field	Required?	What it does / valid values
User requirements specification (URS)	Yes	PDF or Word (.docx), up to 50 MB. Not password-protected; it must contain text (scanned pages are not read). Held in memory for the run only.
Word template (optional)	No	A .dotx or .docx file that changes the documents' appearance. Without one, the built-in FDS template is used.
Include FRS	No	Adds the FRS to the package, and FRS IDs to the matrix. Unticked by default.
Parameter set (optional)	No	A parameter set (.json , up to 1 MB and 10,000 values) saved with Download parameter set . Its Site and Test values are applied to the selected tests; Instance values are left blank. Held in memory for the run only, and never sent to the AI.

Download with values:

Field	Required?	What it does / valid values
Each parameter field	No	The value printed in the documents, in the steps where the parameter is entered. Up to 1,024 characters, on one line; a number, date or address where the field asks for one; one of the allowed values for a list. Left blank, the field stays blank in the document.
Parameter set file (.json)	No	A parameter set to fill the fields from, with Upload parameter set .
Word template (optional)	No	As for a download without values: a <code>.dotx</code> or <code>.docx</code> file that changes the documents' appearance.

Related Topics

- [AI Insights](#): connections and the routing of the Validation Package Generator workload
- [Security and Sign-In](#): roles
- [Electronic Signatures and Audit Trail](#)

Glossary

Terms as they are used in the IntelliTrack application. See the linked chapters for full detail.

Term	Meaning
Enterprise / Site / Area / Workcell / Equipment	The five levels of the ISA-95 equipment model, from the top of the organization down to a single machine. See Chapter 2 .
Show deleted	The administrators' box on the plant model and on the list of every other deletable configuration item. Ticked, deleted items are listed too, greyed, struck through and marked Deleted , with Details and Restore in place of Edit and Delete . A restore brings back that one item under its name from before the delete; what was deleted with it, and what it belongs to, is restored item by item. See Restore a deleted item .
Current site	The site this installation's database is bound to (System Parameter 1, Site Id), set with Is Current Site on the site's form by an administrator (Engineering users see it read-only). Production, electronic signatures and quarantine searches all need it. See Set the current site .
Process	A named manufacturing step that equipment performs; attributes and recipes are defined against processes.
Process Attribute	A data point captured at a process (numeric, boolean, string, or curve). An attribute marked an audit point is verified during product audits. A global attribute belongs to no process and can be used by every process; a curve metric attribute is a number worked out from a curve when the curve is logged, and graded like any other reading.
Communication anomaly	A gap in a machine's heartbeat longer than System Parameter 81 (30 seconds as shipped), listed on the equipment's Details page.
Fault code	A machine's numbered alarm or fault, defined per piece of equipment on its Alarm Definitions pages (Engineering or Administrator role). A retired fault code is no longer in the machine's list but is kept so earlier alarms keep their description. Its product impact (0, 1 or 2) is recorded on each alarm and can filter reports and notifications; it does not yet act on product by itself.
Material	The definition of an item made or consumed; products are instances of a material. Identified by Name + Revision .
Revision	A version of a material, recipe, or BOM.
Supplier	A source of incoming material; its status limits the status material received from it can start in. Its Supplier Id is the vendor code printed on its labels, which receiving matches scanned labels against.
House supplier	The shipped supplier (<i>Plant</i> until renamed) that owns products created on New Product or received without a supplier.

Term	Meaning
Product	A tracked item — a serialized unit, lot, batch, or quantity of material.
Serial Number / Alias	A product's primary identifier (3 to 82 characters, never reused) and an alternate identifier (also unique); both are searchable. Look-ups also find a product by its current shipping code, or by a serial number it used to carry before it was renamed (a former serial).
Disposition	A status code for a product (or a supplier, work order, or customer), with rules such as OK to work and OK to ship . For example, Hold stops a product being shipped but not being worked.
Restrictive disposition	A change of a product's status that only narrows what it may do: it turns OK to work or OK to ship off, turns neither on, and leaves Include in count as it was — <i>Released to Hold</i> , for example. Decided from the two statuses' own flags. The Production role may make it; any other status change (a release from Hold, a move to <i>Scrapped</i>) is permissive or lateral and needs a Production Supervisor . See Products .
Pass/Fail (PF) result	IntelliTrack's quality convention: 1 = Good , 0 = In Process , negative = Reject . Any signed-in user may reject a product; changing a reject back, or any other result change, is a release decision .
Recipe	The definition of how a product is made and the limits its measurements must meet.
Recipe Value	The target and limits for a process attribute within a recipe. Measurement limits are recipe-driven .
Control Instance	A recipe revision prepared for a specific equipment module, with its own frozen copy of the limits. One prepared on the Control Instances page must be reviewed and released before a machine downloads it; a web Check-Out activates one directly from the recipe the operator chooses.
Campaign	A named run of one recipe on a workcell. The control instances of that run are grouped under it, and check-outs and machine downloads join it.
Recipe Context S/N / Dynamic BOM Context S/N	Optional Check-Out fields: the product a control instance's calculated values are worked out from, and the assembly a part is going into (the part is admitted only if the dynamic BOM pairs the two).
Deviation	A recorded difference between a control instance's actual values and the recipe. The Recipe role and Production and above may comment on one; the Recipe role, a Production Supervisor or an administrator may accept it; the Recipe role or an administrator resolves it.
Override	A change-controlled adjustment to a control instance's values, with a reason code and the approvals its signature policy requires. It takes effect when it is next applied (at the machine's confirmation, or the next web check-out), not at approval; a revert puts the master recipe's values back. A Production Supervisor or administrator requests one; the Recipe role or an administrator — never the requester — approves or rejects it; Production and above may revert it.

Term	Meaning
ECO	Engineering Change Order; a recipe's ECO value is validated against an ECO key pattern.
GTIN	Global Trade Item Number assigned to a material.
Bill of Material (BOM)	The list of components for a finished product. Static BOMs list component materials; dynamic BOMs list specific produced→consumed product pairs. Composition shows a BOM with its nested BOMs, as a tree or a list, and Used In shows where it is itself a component.
Incomplete BOM	The reject reason a check-in gives when the station's static BOM components are not all present. Such a part is never scrapped automatically.
Genealogy	The record of what a product was built from and what it became, using the produced (assembly) and consumed (component) convention. A product's first entry, its birth record , points at itself; every trace starts from it, so it cannot be deleted.
Work Order	An instruction to produce a planned quantity of a material; moves through Draft → Scheduled → Released → Running → Completed → Closed (or Canceled).
Overbuild tolerance	How far past its planned quantity a work order may build, as a percentage; the order's Maximum .
Hand-off	Moving a unit to another work order while the order it left keeps the credit for it.
Equipment enforcement	One machine's own setting for the work-order and recipe enforcement parameters (System Parameters 90–93), in place of the site's, set by the Engineering role or an administrator. See Work Orders .
Check-Out	Choosing the equipment and recipe and identifying the product (its serial number, or a carrier and position) to begin processing it; the OK-to-Work gate decides whether it may proceed.
OK to Work / Not OK to Work	Whether a product is permitted to be processed at the chosen equipment under the chosen recipe.
OK to Ship	Whether a finished product — and everything in its genealogy — has passed all requirements and may be shipped; a passing check shows its Shipping Verification Code .
Skip token	A value a machine receives at check-out and echoes back to say it intentionally skipped an attribute it cannot measure on that cycle.
Queue	The list of products waiting at a process or equipment.
Carrier	A reusable pallet, tray, magazine, or fixture that holds products at numbered positions.
Group (tray) operation	Checking a whole carrier of parts out or in together, by entering the carrier's name where a serial number would go; every part must pass, and all get the same result.

Term	Meaning
Audit cycle	An equipment module's audit schedule: when the next audit is due, the grace period after it, and whether the line is being held.
Audit pull (audit sample)	A part the equipment checks in as an audit sample. It opens an audit record and resets the audit cycle, releasing the line.
Pending Audits	The QA backlog of open audit records waiting to be measured and closed.
Audit override	Skipping or postponing a due audit within the process's limits. A release decision : made by a Production Supervisor or administrator, and, with GxP Mode on, signed by someone who holds Production Supervisor.
Destructive audit	An audit whose pulled part may never re-enter production: it gets the destructive-audit status and is refused at every later check-out.
Quarantine Incident	A quality hold that finds suspect products (by blacklist, equipment, genealogy, or performance) and, once confirmed, moves them to a chosen status.
Staged (quarantine)	The parts a quarantine search has found and set aside for an incident. Nothing changes until the incident is confirmed.
History-only part	A part a quarantine search found in the Reporting archive that is no longer held at this site; it is only counted on the confirmation page.
Issue product	In an equipment incident, a part that was actually processed on the suspect equipment, rather than one caught through its genealogy.
Acknowledgement (quarantine)	The recorded outcome for one held part. Only <i>Scrapped</i> and <i>Destroyed</i> change the part's status. Marking a part <i>Contained</i> is Production's; every other type is a release decision for a Production Supervisor or administrator.
Blacklist	Serial numbers a Black List quarantine incident lists as suspect: a new product with one of them is refused when it is created or received, or given the incident's status when that status still allows work. The entries last as long as their incident.
OEE	Overall Equipment Effectiveness = Availability × Performance × Quality .
Mode / State	OEE classifications: a mode sets the operating context (its <i>scheduled</i> flag marks downtime planned vs. unplanned); a state classifies activity (running, faulted, blocked, starved).
MTBF / MTTR	Mean Time Between Failures / Mean Time To Repair.
Active Alarms board	The OEE page listing the machines' standing alarms, unanswered first, where Production can acknowledge an alarm or raise a maintenance work request from it. See OEE .
First-out alarm	The alarm that started a cascade — the one to fix; the others followed from it.
Shelved alarm	An alarm deliberately set aside for a stated time, with a reason; left off the Active Alarms board unless you choose to include it.

Term	Meaning
Alarm rationalisation	Recording, for each fault code, its cause, its consequence, the operator's action and the response time — what turns a list of messages into an alarm system.
Alarm health measures	Floods (more than ten alarms in ten minutes), chattering alarms (re-raised within a minute of clearing), stale alarms (standing more than 24 hours) and fleeting alarms (gone within five seconds), reported on Alarm Health .
Machine time	The timestamp a machine's controller reported for an alarm, shown beside IntelliTrack's own time for reference only.
Alarm archive	Copies of alarm records kept after the purge removes them, for the number of days in System Parameter 17.
Downtime reason	The configured reason given for a stop. It carries whether the stop was planned and which of the six big losses it belongs to; downtime with no reason recorded is unexplained downtime .
PPD	Process/performance data captured during production (and audits); the basis of performance reporting.
SPC	Statistical Process Control. The chart's control limits are worked out from the process data (the mean $\pm$ 3 sigma); the recipe's alarm limits are drawn as the specification lines that capability is measured against.
Western Electric rules	Four run rules the SPC chart applies to flag a process that is out of statistical control.
Cp / Cpk, Pp / Ppk	Process capability against the recipe's alarm limits, using within-subgroup variation (Cp, Cpk) or overall variation (Pp, Ppk).
First Pass Yield (FPY)	The share of products that were good at their first attempt at a module.
Rolled Throughput Yield (RTY)	The product of the first-pass yields of the selected steps: the chance a product passes every one of them first time.
WIP	Work In Process — product currently in production. A product counts as WIP while it has an open processing event (checked out and not yet checked in).
Dashboard	A saved, auto-rotating slideshow of reports for unattended display.
Publication / Subscription / Profile	A notifiable event, a link from a recipient to that event, and a notification recipient (with email/phone addresses).
Publication type	The kind of event or condition a publication watches, chosen as its Publication Type Code and fixed once saved. See What you can be notified about .
Notification channel	How a site's notifications leave IntelliTrack: its mail server, its Twilio account for text messages, or a Teams or Slack webhook. Set up by administrators; see Set up notification channels .

Term	Meaning
Inbox	Where the notifications sent to you In-app appear inside IntelliTrack: the Inbox button under your name in the menu bar, with a red bubble counting the ones you have not read (1 to 9, then 9+). Yours alone. See Check your notifications (Inbox) .
Cooldown / suppression window	Cooldown Minutes keeps a publication quiet about the same item for a while after it notifies; a subscriber's own Suppression Window Minutes skips repeats from that publication for them.
Channel permissions	The channels a notification profile accepts. A message goes out on a channel only when the subscription enables it, the profile permits it, and the profile has an address for it.
Private publication	A publication visible only to its creator and administrators. It does not stop anyone who knows its Id from subscribing.
Approval lifecycle	The shared Draft → Under Review → Approved → (Pending Effective) → Active workflow for materials, recipes, and BOMs; an item can later be retired (Inactive , Obsolete , Archived — the retired statuses, kept for the record but out of use). Each forward step collects the number of approvals its signature policy sets, whether or not GxP Mode is on.
Pending status change	A lifecycle move waiting for the approvals its signature policy requires. The record is locked until the change is approved, rejected, or cancelled.
Report History	A saved record of a report run, reopenable via a shareable link and replayed in dashboards.
Reporting Index	The Reports → Reports landing page, with a tile for each area of reports.
Where-Used / Recall Impact	A report listing every downstream product whose genealogy contains a given serial number or lot — the scope of a recall.
Include Extended Data	A Product Report option that adds the process data of the unit's genealogy relatives.
API Client	A holder of an API key (X-API-Key) used by external systems to call IntelliTrack's APIs. It acts as a service identity .
Service identity	The identity that background work and API clients act under. It never signs in at the sign-in page, never holds the Administrator, Recipe, Engineering or Production Supervisor role, and cannot give an electronic signature.
ERP ledger	The record of what IntelliTrack has sent to the ERP: one entry per event, ERP system and transaction, marked Pending , Sent , Failed or DeadLettered . A dead-lettered entry is not retried until someone selects Retry .
Watermark (ERP)	How far through IntelliTrack's history the ERP worker has read for a transaction; the first run starts from the beginning.
Field mapping (ERP)	A translation of an IntelliTrack value (for example a unit of measure or an item) to the value the ERP expects.

Term	Meaning
System parameter	A numbered setting kept per site, in bands of ten by subject; parameters numbered 1000 and up are ones your site created. See System Configuration .
Log Search	The page (Configuration → System Errors & Warnings) that searches the site's error and warning logs together.
Branding	Your company's name, logo, colours, browser-tab icon and contact details, set by an administrator on Branding and shown throughout the application.
Session	Your signed-in time in IntelliTrack. It ends after a period without activity (15 minutes as shipped), after its maximum length (12 hours as shipped), or when you sign out.
Role / role mapping	IntelliTrack's eight roles — General , Production , Production Supervisor , Recipe , Engineering , Quality , Audit , and Administrator (key <code>Admins</code>) — decide what a person may do. Production Supervisor includes Production, and Administrators pass every Production, Production Supervisor, Engineering and Quality check (except signing a release decision under GxP Mode); the other roles each open only their own pages. Which directory groups, identity-provider roles, or identity-provider groups give each role is the role mapping , kept by an administrator on Configuration → Additional Configuration → Security & Administration → Security → Role mapping ; a person who matches no mapping has no access. See Security and Sign-In .
Production Supervisor	The role for shift leads and supervisors: everything the Production role (line operators) may do, plus the acts that release, correct, override or authorize — releasing product from Hold, changing a pass/fail result, renaming or correcting a product, scrapping, splitting, merging and consuming, work-order administration, skipping or postponing audits, dispositioning and rolling back quarantine, requesting recipe overrides. It includes Production without a second mapping, signs release decisions , and is never held by a service identity. Shipped group: <i>IntelliTrack Production Supervisor</i> . See Security and Sign-In .
Engineering	The role for controls, process and manufacturing engineers who maintain the plant and equipment model: enterprises, sites, areas, work cells, equipment and locations; processes and routes; attributes and attribute maps; OEE modes, states and user-defined fields; equipment enforcement; and each machine's alarm definitions. It gives that work without security, system parameters, licensing or signature policy, which stay with Administrators — who can also do everything Engineering can. It includes no production role, can be named as the signer in a signature policy (none does as shipped), and is never held by a service identity. Shipped group: <i>IntelliTrack Engineering</i> . See Security and Sign-In .
Quality	The role for quality technicians who verify production: they close a process audit record with its measurements and verdict (Product Audits). It includes no production role and no production role includes it, because the audit checks the line; Administrators can also do everything Quality can. It signs a closed audit as shipped and is never held by a service identity. Shipped group: <i>IntelliTrack Quality</i> . See Security and Sign-In .

Term	Meaning
Release decision	An act that lets stopped or suspect product continue, or waives a quality check. There are seven: changing a reject to a pass (or any result change other than to reject), releasing product from Hold or another restrictive disposition, dispositioning quarantined product, rolling back a quarantine incident, and skipping a due audit or postponing it by time or by units. Made by a Production Supervisor or administrator; with GxP Mode on, it must be signed by someone who holds Production Supervisor , so an administrator without that role cannot sign one. See Who may sign: roles .
Identity provider (OpenID Connect)	Your organization's sign-in service, offered on the sign-in page as Sign in with its name . It checks who you are (including any extra verification it asks for) and sends you back; its roles or groups are mapped to IntelliTrack roles. IntelliTrack stores no password for any way of signing in. A person signed in this way cannot give electronic signatures in this release (see Which password you sign with).
Sign-in review	The administrators' list of refused sign-ins waiting for a decision — for example someone whose groups map to no role, or who matches an existing user who signs in another way. See Security and Sign-In .
Provider link	An administrator-approved link from an existing IntelliTrack user to their account at a new sign-in service, so their history continues under one person.
Security audit trail	The record, kept on the Security pages, of every change made there (role mappings, settings, links); entries are only ever added.
My Access	The page your name in the menu bar opens (also My access on pages that refuse you): how you are signed in and, for every role, whether you hold it and which of your groups or roles grants it.
Display date format	The one site-wide way dates and times appear on screens, reports, chart axes and export headers, chosen by an administrator from presets (Date and Time Format on Configuration → Additional Configuration ; System Parameter 130). Not a personal setting, and not tied to the interface language. Signature and audit-trail pages, and a few others, keep a fixed year-month-day form. See Date and time format .
Audit trail	The permanent, unalterable record of every change to configuration and quality records: who (and in which roles), when, what changed from what to what, and why. Always on. See Electronic Signatures & Audit Trail .
Audit trail review	A documented review of the audit trail for a period (optionally one record type) by someone who changed none of the records in scope, recorded with its Findings and closed by the reviewer's electronic signature (meaning <i>Verified</i>) — always signed, even with GxP Mode off.

Term	Meaning
Audit role	The IntelliTrack role that reads the audit trail and decides about its findings, and changes nothing — not configuration, not production, not the audit trail's own settings. Held separately from Administrators so that the people most able to change a record outside the application are not the only people who can look for evidence that someone did.
Audit trail workspace	The page a reviewer starts from: findings needing attention, when the last check ran and what it found, whether a review is due, and whether the tamper evidence is intact.
Check (for changes made outside IntelliTrack)	The comparison of IntelliTrack's own record of a change with SQL Server's history of the same table. A change with only the database's half did not come through the application. Runs on a schedule (System Parameter 124) and on demand from the workspace.
Finding	One change the check could not account for. Not an accusation: most have an innocent explanation, and the record needs somebody independent to write down which.
Decision (about a finding)	Explained — accounted for, naming an approved change control or work order — or Escalated . A decision is never replaced; changing your mind adds a second and keeps the first.
Degraded check	A check that ran with part of its detection unavailable — usually because SQL Server Audit is not installed, so direct writes to the production tables were not being watched. It is never presented as a clean result.
Entry-by-entry review	An audit trail review worked one entry at a time, each marked Reviewed or Flagged , which can be left and resumed and cannot be signed off while any entry is undecided or any finding in the period is still open. The alternative is a single attestation over the whole period; a site's procedure decides which.
SQL Server Audit	The database's own record of who wrote what, which covers the two things the audit trail comparison cannot: the tables production writes on every part, and an edit made within a few seconds of a legitimate change to the same record. Offered when IntelliTrack is installed; until it is on, every check reports a degraded result and says so.
Archive (audit trail)	A dated, counted and checksummed copy of audit trail entries older than the retention period (System Parameter 126). A record still being changed keeps its whole history, however old its earliest part. Archiving copies and verifies; it does not remove anything.
Tamper evidence	Each audit trail entry and each signature carries a fingerprint over every one of its own stored values and over the previous row's fingerprint, so altering, removing or inserting one is detectable. Verify now re-derives them all and keeps the result.
Electronic signature	A person's password, given on a change to record that they performed, reviewed or approved it; stored with their printed name, the date and time, and its meaning , and linked to the record so it cannot be removed or copied. Asked for in an Electronic signature panel above the form's buttons; a page showing a signed record lists its Electronic signatures .

Term	Meaning
GxP Mode	The switch that turns electronic signatures on (System Parameter 110, Configuration → Additional Configuration → Compliance → GxP Mode). Off: no change asks for a signature except the two that are always signed — an audit trail review and a change to GxP Mode itself; the audit trail runs either way.
Signed by default / Tier 2	Whether an act is signed as shipped once GxP Mode is on. Most acts are signed by default; Tier 2 acts (marked so on Signature policies) are signed only if an administrator turns their policy on. See Which acts can be signed .
Signature meaning	What a signature signifies — for example <i>Authored</i> , <i>Reviewed</i> , <i>Approved</i> , <i>Disposition Decision</i> — shown on the panel and recorded as shown.
Signature policy	The rule that says whether an act needs a signature, with which meaning, how many signers, in which role, and whether a reason is asked for; set per Enterprise, site, area, workcell, process, material group, recipe type or object, and the most specific one applies.
Reason code	A predefined reason, grouped by category, chosen when a change is signed; some also require a comment. A reason code is retired, never deleted, because signatures and audit-trail entries cite it.
Inactivity lock	Locks a browser left untouched for the set number of minutes (System Parameter 114; 0, the default, turns it off) until its own person enters their password again. Unlike a session that ends, it returns you to the page you were on. Someone signed in through an identity provider cannot unlock it and must close the browser and sign in again.
Tamper-evidence verification	A check, run on demand (Verify now), that recomputes every signature and audit-trail chain, reports any entry changed, removed, or inserted outside IntelliTrack, and confirms their protections are in place; each run is kept.
Signature readiness	The page that shows whether this server can verify signatures made with domain accounts (the Kerberos preflight, each failing check with its remedy), together with the last signature, the last verification run, and the settings that govern signing.
URS (User Requirements Specification)	The customer's document stating what the system must do. The Package Generator extracts its requirements and keeps each one's source ID, wording and location unchanged. See Chapter 21 .
FRS (Functional Requirements Specification)	The optional package document stating how IntelliTrack meets the URS obligations. Its statements come only from the library's released function descriptions; a requirement no standard function satisfies is a Custom item, and values the URS does not give are placeholders. See Chapter 21 .
IQ (Installation Qualification)	The protocol verifying that IntelliTrack is installed as specified: prerequisites, databases, application files, configuration and the optional components the site uses. Its product facts are printed from the release. See Chapter 21 .
OQ (Operational Qualification)	The protocol verifying that the functions the URS needs work as specified, with the operational and integration tests the Package Generator selects.

Term	Meaning
PQ (Performance Qualification)	The protocol demonstrating the system in routine production. The customer owns it: it sets every acceptance criterion, rate and period, and executes it in production.
Obligation	One individually assessable part of a URS requirement, with its technical classification, responsible parties, functions and tests. A compound requirement is split into several obligations under the unchanged requirement.
Test instance	One execution of a library test in a package, with its own instance ID and parameter fields: for example, the same test once per equipment module.
Parameter set	A file (JSON) of parameter values, saved from Download with values and kept by the customer; uploaded again, it fills in a values form or supplies a Package Generator run's Site and Test values. IntelliTrack never stores one. See Parameter sets .
Parameterized instance	A test document downloaded with at least one user-supplied value. Its title page and header say so, with its instance ID; the master document carries no values. See The parameterized document .
Library baseline	The signed file of the validation library, built by FDS for one IntelliTrack version and installed with it; every library page and generated package works from it. A library update replaces it with a newer one. See Library updates .
Change Notice	The list of the tests a library baseline adds, revises, supersedes or withdraws, each with its change class (Editorial, Correction or Product) and reason.

IntelliTrack User's Guide — Chapter Outline

This outline is derived from the application's own navigation menu (`Views/Shared/_Menu.cshtml`) and the controller/View inventory in `_feature-inventory.md` (in the sibling `UserGuideAuthoring` folder, for documentation maintainers). Chapters are ordered by the user's likely journey: getting started, the data model that everything else depends on, the core daily production workflows, quality/disposition, then periodic reporting and administration, followed by the reporting data dictionary (18), security and sign-in (19), electronic signatures and the audit trail (20), and validation documentation (21).

Navigation map (exact menu labels)

The top navigation bar exposes seven menus. Each title only opens its menu, on a click (hovering does not open it, and the title is not a link); the page a title used to open is an item in its menu. The menus group the pages by the activity areas of the ISA-95 (IEC 62264-3) standard that most manufacturing execution systems use — **Production**, **Quality**, **Product Definition** (what is made) and **Inventory** — after **Home** and **Reports** and before **Configuration**. Within a menu the items run in the order the work happens. The menus are the same for every signed-in user, with two exceptions, both for pages only Production users and above can use: the **Production** menu lists **Check-Out** only to them, and the **Inventory** menu (**Receiving** and **Shipping Verification**) is shown only to them. Otherwise the gate is on the destination: a page whose role the person lacks renders **Unauthorized**, naming the role keys it needs, with a **My access** link. Items whose landing page needs more than `General` are marked with their roles below.

Menu (label)	Items (label → destination)
Home	Home → <code>Home/Index</code> ; Help → <code>Help/Index</code> ; About → <code>Home/About</code> ; Contact → <code>Home/Contact</code>
Reports	Reports → <code>Reports/Index</code> ; Product Report → <code>Reports/ProductReportSearch</code> ; Performance Reports → <code>PPDReports</code> ; Quality → <code>QualityReports</code> ; Audit Results → <code>Audit/Results</code> ; OEE Reports → <code>OEE</code> ; Inventory Reports → <code>InventoryReports</code> ; Supplier Reports → <code>SupplierReports</code> ; Carrier Reports → <code>CarrierReports</code> ; AI Insights → <code>AiInsights</code> (Admins); Dashboards → <code>Dashboard</code>
Production	Work Orders → <code>WorkOrders</code> ; Production → <code>Production</code> ; Check-Out → <code>Production/CheckOut</code> (Production and above; hidden from others); Products → <code>Products</code> ; OEE Events → <code>OEEEvents</code>
Quality	Product Audits → <code>Audit</code> ; Quarantine → <code>Quarantine</code>
Product Definition	Materials → <code>Materials</code> ; Recipes → <code>Recipes</code> ; Bills of Material → <code>BOM</code>

Menu (label)	Items (label → destination)
Inventory (Production and above; hidden from others)	Receiving → Products/Receive ; Shipping Verification → Production/OKToShipCheck
Configuration	Equipment Model → Enterprises ; Processes → Processes ; Notifications → Notifications , a page of tiles: Inbox → Inbox , Subscriptions → Subscriptions , Publications → Publications , User Profile → Profile , and for Admins Delivery Log → NotificationLog and Notification Channels → NotificationChannels ; API Clients → ApiClients (Admins); Licensing → Licensing/Overview ; System Errors & Warnings → SystemLogs (the Log Search page); and last, Additional Configuration → Home/Configuration , which sorts the setup pages without a menu item into groups (see below): Notification Channels (NotificationChannels , Admins) is in its Connections & Notifications group, and its Compliance tile leads to the audit trail and electronic signature pages

Right of the menus (the header), three short rows: **Current User:** *name* (the session's display name; hovering **Current User:** shows the account name), the name being the link to My access → Access (open to anyone, even without a role); then **Inbox** → Inbox , a bell pill (the person's own in-app notifications, with the unread count from Inbox/UnreadCount asked once as the page loads, shown 1 to 9 then **9+** in a red bubble, none when nothing is unread), **Sign out** (a POST to Account/SignOut), or **Sign in** → Account/SignIn?method=choose when there is no session, and the language flags (one per supported language); then **Site Name:** *site* (or **Site Not Set**).

Additional Configuration (Home/Configuration , page heading **System Configuration**) has no menu items of its own; it holds the **Items per Page** (per user) and **Date and Time Format** (site-wide, Admins to change) cards, then one tile per group of setup pages. A group's tile opens Home/ConfigurationGroup/{group} , a page with one tile per setup page. A tile is shown only to a role its page admits, and a group none of whose pages the person may open is not offered (its address answers 403; an unknown group, 404):

- **Equipment & Processes** (Equipment): Equipment Model Definition → Enterprises ; Process Configuration → Processes ; Attributes → Attributes ; OEE Configuration → Oee/ConfigurationIndex (its Modes, States and UDF Definitions tiles open OeeModes , OeeStates and OeeUdfDefinitions , all Engineering or Admins).
- **Materials & Logistics** (Materials): Carriers → Carrier ; Locations → Locations ; Units of Measure → UoM ; Suppliers → Suppliers .
- **Quality & Change Control** (Quality): Disposition Codes → Disposition ; Approval Statuses → ApprovalStatus ; ECO RegEx Definitions → ECORegexKeys (Admins).
- **Connections & Notifications** (Connections , all Admins): System Connections → Configuration ; Email Server (SMTP) → Configuration/Smtp ; Notification Channels → NotificationChannels .

- **Security & Administration** (Administration): Security → Security (Admins: Role mapping, Security settings, Sign-in review, Users, Provider links, Connection test, Security audit trail); System Parameters → Parameters; Branding → Configuration/Branding (Admins).

The last tile, Compliance → Compliance (Admins), is a page of tiles, one per audit trail and electronic signature page (all Admins): Audit trail → AuditTrail; Audit trail reviews → AuditTrail/Reviews; Tamper-evidence verification → AuditTrail/Verification; Signature policies → SignatureAdmin/Policies; GxP Mode → SignatureAdmin/GxpMode; Reason codes → SignatureAdmin/ReasonCodes; Signature attempts → SignatureAdmin/Attempts; Signature readiness → SignatureAdmin/Readiness. Until 2026-09-22 those eight were a **Compliance** group at the foot of the Configuration menu.

Until 2026-10-08 there were five menus: **Home**, **Reports**, **Production** (which also held Receiving, Shipping Verification, Quarantine and the product audit page, listed twice as **Quality Audits** and **Product Audits**), **Recipes** (Recipes, Materials, Bills of Material) and **Configuration**.

Role model

Access is controlled by the `[AuthorizeRole(...)]` attribute, which checks the IntelliTrack role set of the person's session — resolved at sign-in from their groups or identity-provider roles, and re-resolved when the mapping changes. It never asks the directory itself. Which Windows groups, identity-provider application roles, or identity-provider groups give each role is set on **Configuration** → **Additional Configuration** → **Security & Administration** → **Security** → **Role mapping** (Security/RoleMappings, Admins; the old Configuration/SecurityRoles address redirects there). The mapping is kept in the Config database; it was derived once, at first start, from the `Security:Roles:*` settings, whose shipped values are:

Role key	Name shown on My Access	Shipped Security:Roles group	Guide wording
General	General	Authenticated Users (every Windows sign-in); for an identity provider, map an all-users role or group	"any signed-in user"
Admins	Administrators	IntelliTrack Admins	"the Administrator role"
Production	Production	IntelliTrack Production	"the Production role" (line operators)
ProductionSupervisor	Production Supervisor	IntelliTrack Production Supervisor	"the Production Supervisor role"
Recipe	Recipe	IntelliTrack Recipe	"the Recipe role"

Role key	Name shown on My Access	Shipped Security:Roles group	Guide wording
Engineering	Engineering	IntelliTrack Engineering	"the Engineering role" (controls, process and manufacturing engineers: the plant and equipment model)
Quality	Quality	IntelliTrack Quality	"the Quality role" (quality technicians: closing process audit records)
Audit	Audit	IntelliTrack Audit	"the Audit role"

Where a screen requires more than one role (e.g. "Admins, Recipe") the user needs **any** one of the listed roles. Production Supervisor includes Production, and Administrators pass every Production, Production Supervisor, Engineering and Quality gate; that inclusion is built into the product, not made by mapping. The one exception: with GxP Mode on, the seven **release decisions** must be signed by a holder of Production Supervisor, which their shipped signature policies name as the required signer role. Other roles do not include one another: Engineering includes no production role, and none includes it; Quality (closing a process audit record, Audit/Complete POST, gated Quality, Admins) includes no production role and no production role includes it, because the audit checks the line. The plant hierarchy and locations, processes, routes, attributes and attribute maps, OEE modes, states and UDFs, equipment enforcement and alarm definitions are gated Engineering, Admins ; security, system parameters, licensing, branding, integrations, signature policy, approval statuses and master data (units of measure, suppliers, carriers, GTINs, ECO keys, dispositions) stay Admins . Any signed-in user may reject a product. An attribute on an action replaces the controller's, which is how many Index pages on Admins and Engineering controllers are viewable by General . **My access** (Access) shows a person every role, whether they hold it, and which of their groups or roles grants it.

Chapter plan

#	File	Scope (one line)	Primary inventory rows
1	01-getting-started.md	Signing in and out, roles and My access , reading the menu bar (current user, site), status and license messages, top-menu navigation, the home page and product search, the in-app Help center, date/time and language preferences, About/Contact.	Home, Help, Access

#	File	Scope (one line)	Primary inventory rows
2	02-plant-and-equipment-model.md	Building the equipment model: Enterprises → Sites → Areas → Workcells → Equipment (including equipment fault codes), plus Processes and Process Attributes.	Enterprises, Sites, Areas, Workcells, Equipment, Processes, Attributes
3	03-products.md	Looking up products; status, history and genealogy; creating, editing, rejecting and deleting; receiving incoming inventory; split, merge and consume.	Products
4	04-materials-and-suppliers.md	Managing raw/component materials and the shared approval lifecycle, suppliers, and units of measure.	Materials, Suppliers, UoM
5	05-recipes.md	Authoring, versioning, and approving recipes; site namespaces; control instances, values, attribute maps, deviations, overrides, GTIN, ECO keys.	Recipes, ControllInstances, Values, AttributeMaps, Deviations, Overrides, GTIN, ECORegexKeys
6	06-bills-of-material.md	Static and dynamic bills of material, BOM items, their approval lifecycle, and the BOM-versus-genealogy check.	BOM, StaticBomItems, DynamicBom
7	07-work-orders.md	The work-order lifecycle; creating, editing and cancelling orders; assigning and moving products; quantity enforcement, retention, and the work-order API.	WorkOrders, WorkOrdersApi
8	08-production-execution.md	Running production: the OK-to-Work gate, Check-Out, machine (PLC/AOI) runs, the queue, shipping verification, routing, and carriers.	Production, Route, Carrier
9	09-product-audits.md	How audits come due and hold the line, the audit station, recording audit results, skipping or postponing a due audit, the QA backlog, reviewing audit results, and audit configuration.	Audit
10	10-quarantine-and-disposition.md	Raising, confirming, acknowledging and rolling back quarantine incidents; product status (disposition) codes; shipping verification.	Quarantine, Disposition, Production/OKToShipCheck
11	11-oee.md	OEE reports, the state timeline, alarms, OEE event review, downtime, modes/states, and OEE configuration.	OEE*, Downtime, OeeEvents, OeeModes, OeeStates, OeeUdfDefinitions

#	File	Scope (one line)	Primary inventory rows
12	12-reports.md	The shared report mechanics (history, share links, export); the Reporting Index; Product, Where-Used / Recall Impact, Performance (PPD) and SPC, Production, Quality, Inventory, Supplier, Carrier, Material, Curve and WIP reporting.	Reports, PPDReports, QualityReports, InventoryReports, SupplierReports, CarrierReports, MaterialReports, CurveReports, Wip
13	13-dashboards.md	Building, viewing, playing back, archiving and restoring dashboards.	Dashboard
14	14-ai-insights.md	AI Insights, AI memory, AI settings, connections and routing, budgets, rates and usage.	AiInsights, AiMemory, AiSettings, AiConnections, AiBudgets
15	15-notifications.md	Notification publications, subscriptions and publication audiences, user profile addresses and SMS consent, the in-app Inbox, the site's notification channels (email, SMS, Teams, Slack), and the delivery log.	Notifications, Inbox, Publications, PublicationAudience, Subscriptions, Profile, ProfileAddress, NotificationChannels, NotificationLog
16	16-erp-integration.md	ERP connections, field mappings, transaction enablement, and diagnostics.	ErpConnections, ErpFieldMappings, ErpTransactions, ErpDiagnostics
17	17-system-configuration.md	Additional configuration, system parameters, the site date and time format, connection/SMTP setup, the three database synchronization jobs (Config→Site, Site→Reporting, WIP), units of measure and dimensions, locations, approval statuses, branding, API clients, licensing, and Log Search (errors and warnings).	Home (Configuration), Configuration, Parameters, UoM, UomDimension, Locations, ApprovalStatus, Branding, ApiClients, Licensing, SystemLogs, ErrorLogs, Warnings
18	18-reporting-data-dictionary.md	Reference: the read-only Reporting database views (including the carrier views) and the carrier-activity function — columns, data types and descriptions — for people building their own reports.	Reporting database views

#	File	Scope (one line)	Primary inventory rows
19	19-security-and-sign-in.md	Signing in and out, the ways in (Windows, a directory password, an identity provider), sessions, My access , roles and the role mapping, emergency access when the domain is down, and the administrators' Security pages (settings, sign-in review, users and provider links, the security audit trail, connection test).	Account, Access, Security
20	20-electronic-signatures-and-audit-trail.md	GxP Mode and what it switches on, which acts need a signature and how that is decided, signing a change and what the panel asks for, reading a record's signatures, reason codes, the inactivity lock, the audit trail and its review, tamper-evidence verification and signature readiness.	Compliance, SignatureAdmin, AuditTrail, Session, Parameters
21	21-validation-documentation.md	The validation library included in IntelliTrack, reached from the Validation Documentation tile on Additional Configuration; the Document Library (search, filters, preview, single and batch Word downloads, downloads with values and parameter sets) and the coverage register; the Package Generator from URS upload (with an optional parameter set) to the generated package (IQ, OQ and PQ protocols, traceability matrix, optional FRS) and its review in Word; templates uploaded with a download or run; library updates and their Change Notices.	ValidationLibrary, ValidationGenerator

Reader-facing reference files produced in later phases: [README.md](#) (table of contents), [Glossary.md](#).

[NEEDS REVIEW] Chapter grouping merges several small admin controllers (Parameters, UoM, Locations, ApiClients, Licensing, SystemLogs/ErrorLogs/Warnings, ApprovalStatus) into Chapter 17. If reviewers prefer one chapter per menu item this can be split. Logged in `_open-questions.md`.