

Granite WMS What's New V 6.0

New Applications & Tools Overview

- [Business API](#) - An upgraded replacement for the Webservice
- [Jaeger](#) - Your new favorite way to look at logs

Webdesktop Changes Overview

- [Data Capture](#) - A new feature that gives you the ability to easily create input forms to streamline your users' experience capturing repetitive data.
- [Data Grid Improvements](#) - Data Grids now support conditional formatting and configurable page sizes
- [Process Management](#) - You can now easily copy and delete processes right from within the Webdesktop
- [StockTake Session Auditing](#) - Stocktake session creation criteria are now audited, giving you better visibility
- [User Management](#) - User permissions are now managed at the User Group level. You can also copy users from within the Webdesktop

Security & Permissions changes Overview

- [Security Settings](#) - Opt-in system wide settings that allow you enforce policies like password strength and user account lock outs
- [Api Key support](#) - a new way of authenticating against our Repo API and Business API
- [Changes to Permissions](#) - Permissions are no longer columns on the user table

New Integration Providers

- [Sage 100](#)
- [Acumatica](#)
- [Sage Intacct](#)

Custodian / Util

- Custodian merge with Util
- SQLObjects allow adding ad-hoc views, stored proc or tables.

Business API

In this first release of the Business API, our aim is that you shouldn't notice any major differences between it and the Webservice. However, this doesn't mean nothing has changed. Under the hood we've moved away from the legacy technology the Webservice was built on. This modernization enhances our ability to adapt to new requirements, allowing us to respond to business needs more quickly and efficiently. With this new API, we are better positioned to support innovation and deliver improved services to our users.

While the aim is to remain as consistent as possible, there are some changes to the behaviour in the Business API. The most notable change is that creating a pallet will require a location. This is to ensure better

transactional consistency throughout the lifetime of the pallet. This means that you will need to have a Location step *before* any CarryingEntity step that creates pallets. This affects the following process types:

- TakeOn
- Manufacture
- Receiving
- Return
- Palletize

Aside from that, the other changes are fixes that require no changes to processes:

- StocktakeHold (Release) requires the TrackingEntity to be on hold.
- QCHOLD Pallet writes ContainableEntity_id to transaction
- QCRELEASE Pallet writes ActionQty to transaction
- RECEIVING & TAKEON of a MasterItem with DirectOnHold writes ProcessName to QCHOLD transaction
- RECEIVING a document with the same item with multiple ToLocations correctly validates the ToLocation
- MANUFACTURE validates against document detail TO Location (Webservice incorrectly validates against FROM Location)
- PICKING a pallet writes ContainableEntity_id to transaction
- BarcodeMaster will skip barcodes for failed transactions more often than the Webservice - this is part of the strategy to avoid assigning barcodes twice.

System observability with Jaeger

In V6 we're aiming to make it easier than ever to track down issues. To achieve this, we've added OpenTelemetry instrumentation into the Business API and the Process App. This allows us to collect all the information that you would normally see in each application's log file (and much more), and correlate the information to view a complete picture of what happened when a user did something.

All of this information (called trace data) is sent to Jaeger's Elasticsearch database, where it is stored for as long as you need it.

The Jaeger UI allows you to query this data using any of the tags that are stored with the trace, giving you visibility of the whole action like this:

TODO add image showing trace data

Webdesktop Changes

Datacapture Feature

- **new** data capture feature

Data Grid changes

- **new** support configurable page size for data grids [Page Size](#)
- **new** support for formatting data grids [Row Formatting](#)
- **new** standard styles for improved look across application [Styling Guidelines](#)

Process Management changes

- **new** support renaming processes
- **new** support deleting processes

Stocktake Session management improvements

- **new** refresh button for stock take session
- **new** support for stock take session audit

Changes to User management

- **new** support for copying users
- new permissions management in user groups

Minor

- **new** MasterItem Alias SQL view preview. Allow user to preview the SQL view configured for MasterItem Alias.
-

Security & Permissions changes

Security Settings

Security Settings are option-in System Settings that allow you to configure Granite's security policy to meet your customers' requirements. These settings allow you to configure things like password strength requirements, and lock out of user accounts after a certain number of failed attempts. For all the details see the dedicated documentation for [Security Settings](#).

Api Keys

As part of the security overhaul, we've introduced api keys. These can be used in place of a user name and password to authenticate against our Repo API and Business API. This change will allow third-parties a much more seamless experience in working with our apis. SQLCLR is also making use of this new authentication method in the background.

Changes to Permissions

User permissions have changed significantly in V6. We no longer have permissions columns on the **Users** table, instead we have two tables called **SystemPermissions** and **UsersPermissions**.

- **SystemPermissions** - this is where permissions are defined. For a user or group to have a permission, the permission must exist in this table.
- **UsersPermissions** - This is where permissions are assigned to UserGroups. An entry linking a user or user group to a system permission grants the user / group that permission.

This change from fields to records to manage permissions will enable smoother migrations in the future, new permissions will no longer require schema changes.

If a User Group is assigned a permission, all of the users that belong to that group automatically have that permission as well. A user can never have less permissions than the group that they belong to.

Our V6 migration fully converts old user permissions to the new format. For a mapping of old permissions to new permissions see [user permissions](#).