GSRS 3.0.2 Release Notes

August 2022

Table Contents

[Overview: 3](#_Toc111454603)

[Highlighted New Features 3](#_Toc111454604)

[New Feature: Save-as-Draft 3](#_Toc111454605)

[New Feature: Registering Nucleic Acid Fragment CV Terms 4](#_Toc111454606)

[New Feature: Allowing Structure Searches in Advanced Search 6](#_Toc111454607)

[New Feature: Improving Global Search Behavior Part 1: Allow “search within” an existing search or filter 7](#_Toc111454608)

[New Feature: Improving Global Search Behavior Part 2: More intuitive wildcard searches 9](#_Toc111454609)

[New Feature: Improving Global Search Behavior Part 3: More intuitive free-text searches 11](#_Toc111454610)

[Other Issues Addressed 13](#_Toc111454611)

[Improvement: Corrected ORM settings for greater RDBMS compatibility, particularly with PostgreSQL 13](#_Toc111454612)

[Improvement: Improved auto-generated SQL indexes 13](#_Toc111454613)

[Other selected Improvements: 13](#_Toc111454614)

# Overview:

GSRS 3.0.2 software release is a maintenance release which includes several new registrar and searching features, improved performance, and fixes for some installation and user interface defects.

Highlighted new features include:

* saving drafts of records during edit and registration
* registering structural fragments (sugars and linkages) for nucleic acids
* allowing structural searches in advanced search page
* improving global search behavior, including better wildcard and more intuitive search results

Highlighted improvements and bug fixes include:

* more consistent RDBMS table creation and indexing
* improved searching performance
* improved user interface styling
* improvements to entity service deployments

# Highlighted New Features

## New Feature: Save-as-Draft

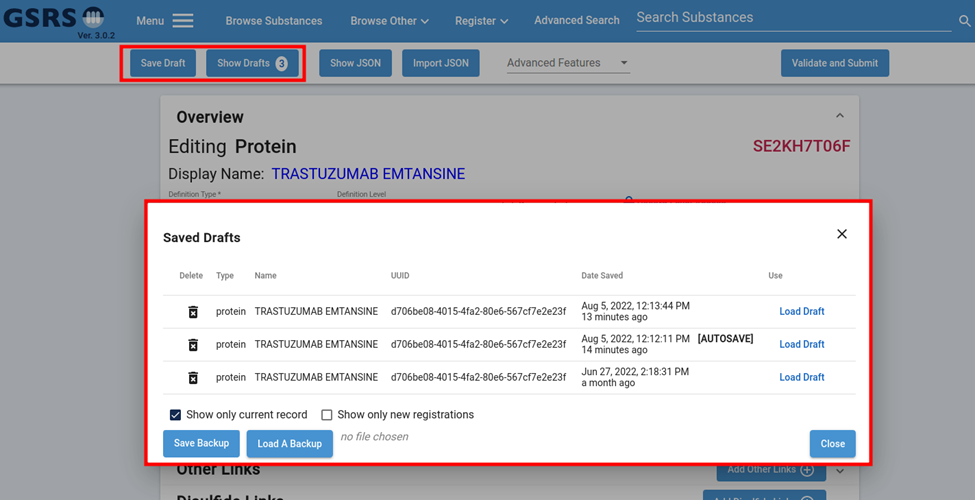


Figure 1: Save Drafts / Show Drafts

**Purpose and Motivation:**

The new Save-as-Draft feature allows users to save and load the current state of any substance edit form. This allows users to easily undo or redo bulk changes, return to a closed edit form without concern for lost effort. These drafts are stored in the user’s local browser files and do not require communication with the back-end to function.

**How it works:**

From the Edit page, two new buttons can be found at the top menu bar: ”Save Draft” To manually save a draft of the current state of the edited record, and “Show Drafts” with a count of current drafts for that record or substance type for new registrations. Clicking “Show Drafts” will open up a new dialog where all locally stored drafts can be filtered, deleted, or loaded into the form. You do not need to be on the edit page of a specific record to load a draft of it, but please note that doing so will overwrite everything in the current form.

Along with manual saves, the form will also automatically create a save every minute after a change is detected, noted by the phrase [AUTOSAVE] next to the date. Up to three of these saves will be stored. The time interval between auto saves can be adjusted in the front-end by creating an “*autoSaveWait*“ property in the config file with a numeric value representing the delay before checking and saving in milliseconds.

At the bottom of the saved drafts dialog, there is also an option to save and load a copy of your drafts in the event you need to clear your browser cache or switch computers.

**Notes for the future:**

Our intention with this feature is to improve the auto-save features and eventually expand the functionality to allow users to view a history of individual changes made to a given form and allow users to load the form at any of those states or undo specific changes without scanning the entire form. This improvement will also help in the general history audit of records.

## New Feature: Registering Nucleic Acid Fragment CV Terms**Graphical user interface Description automatically generated with low confidence**

Figure - Fragment Structure Wizard

**Purpose and Motivation:**

The structure of controlled vocabulary fragments can now be added and edited in the CV forms using the Jsdraw tool and an interactive wizard that was only previously available as a limited undocumented advanced feature. This allows users to save the time and effort of manually generating the smiles of a particular configuration of connection points elsewhere to edit this portion of the CV.

**How it works:**Graphical user interface, text, application

Description automatically generated

Figure - Add Term to CV

Each of the Fragment CV domains – Amino acid residues, nucleic acid sugars, links, and bases – now have a button next to each entry in the admin panel Controlled Vocabulary Management dialog to open the fragment structure wizard. Once star atoms have been set as potential connection points, clicking “Choose Connection Points” will generate all possible combinations for that structure. Clicking one will then set the fragment and simplified structure for that vocabulary term.

Graphical user interface

Description automatically generated with medium confidence

Figure - Sugar Add to CV

The fragment structure wizard is also incorporated in the edit form “Add to CV” option whenever a new term is entered in a field for one of the four fragment vocabulary domains.

The backend vocabulary API service also now uses a few validation rules to protect against some common issues with molecular fragments of this nature. Specifically, there is a new default validation rule that protects against direct chemical duplicates, and chemically uninterpretable strings.

**Notes for the Future:**

The backend mechanisms to detect duplicates, handle stereochemistry, and link more concretely to existing substances that leverage the fragments will be improved in future releases.

## New Feature: Allowing Structure Searches in Advanced Search

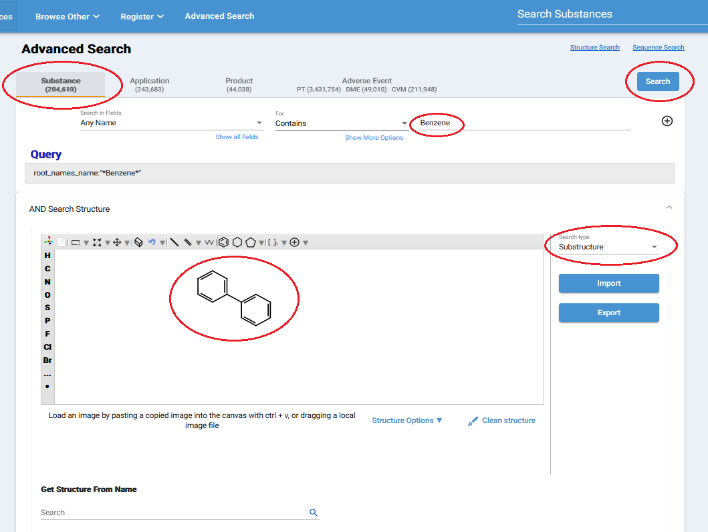


Figure -Advanced Search page with text and structure search

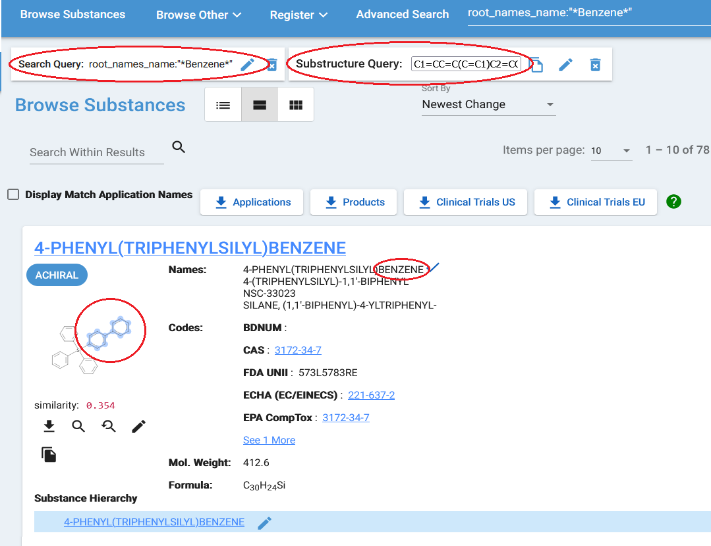


Figure -Advanced Search Results for Substructure search

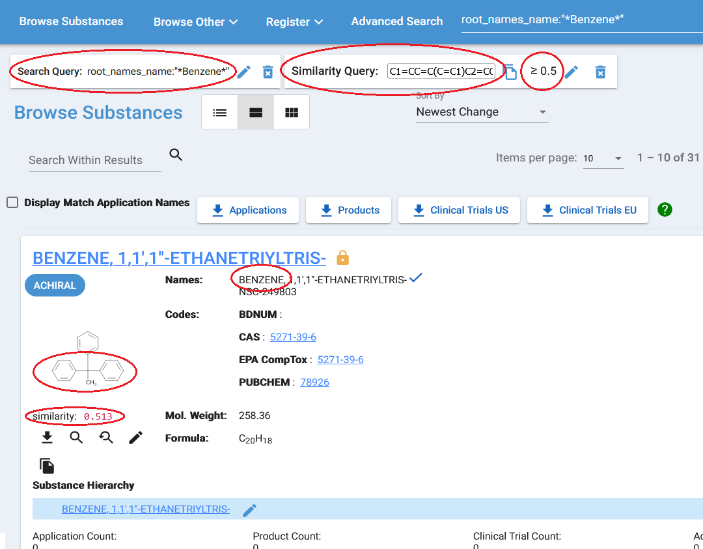


Figure -Advanced Search Results for Similarity search

**Purpose and Motivation:**

In the Advanced Search, in Substance tab, added a structure search with the combination of text search. The combination of text and structure search allow users to do more robust search based on their criteria and needs. This feature is currently available in Substance search only and will be available for other entities in the future release.

**How it works:**

The structure search works by selecting text criteria in the first row and drawing any structure in the structure tool. The users can also select the type of structure search such as :”Substructure:”, :”Similarity:”, :”Exact:”, or :”Flex:” search. When selecting the :”Similarity:” search, a Similarity Cutoff (Tanimoto) slider will appear, where can be selected between 0.4 to 1. See the screenshots above for the Advanced Search page and search results.

**Notes for the future:**

In the future release, the structure search will be available for other entities such as Application, Product, Clinical Trials, and Adverse Events in the Advanced Search.

## New Feature: Improving Global Search Behavior Part 1: Allow “search within” an existing search or filter

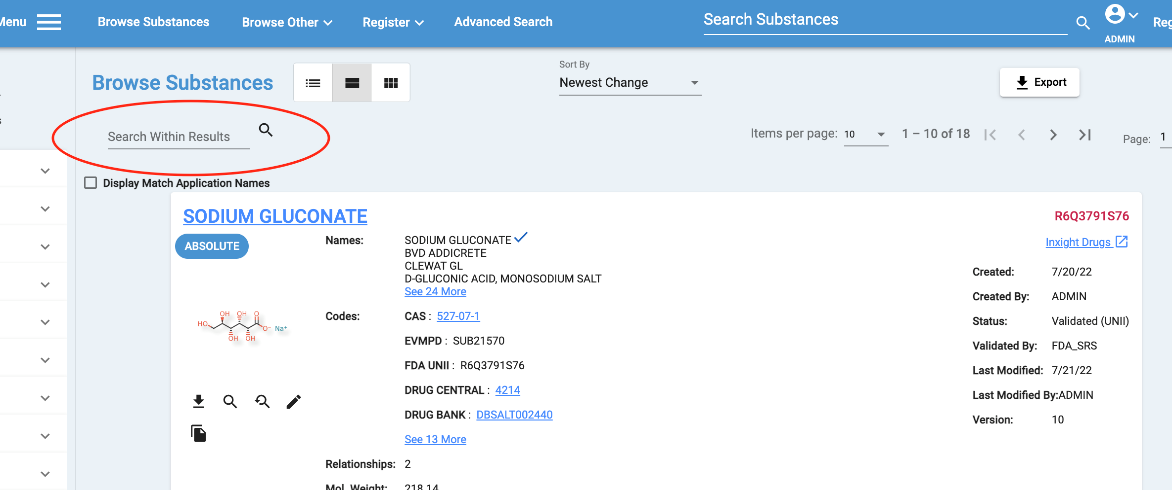


Figure - Search Within Results

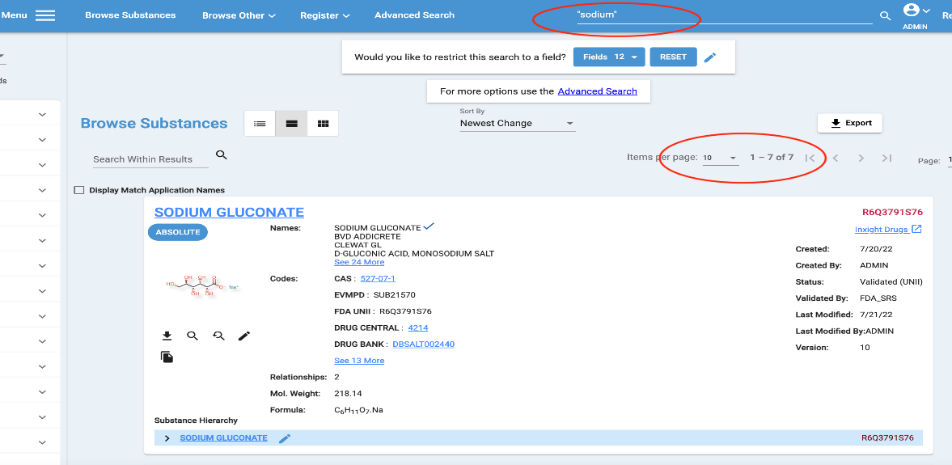


Figure -Search view items per page

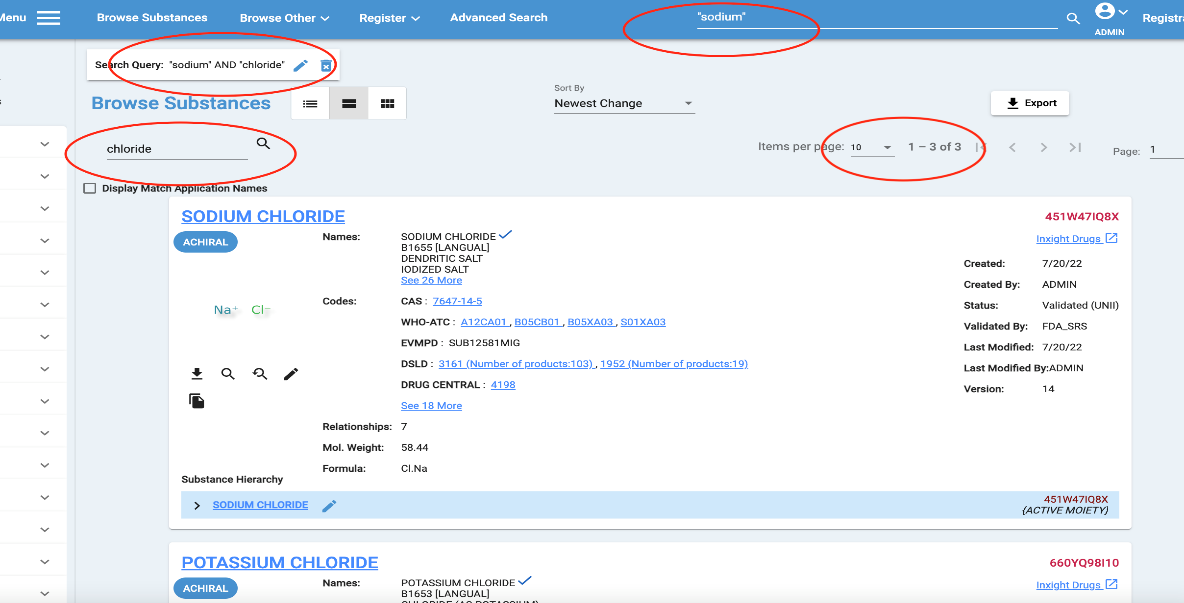


Figure - Search Locations and Search Query and Items Per page

**Purpose and Motivation:**

With this feature, the user is able to search within the results in the substance search, thus letting them make specific searches within the results.

**How it works:**

Say the user makes a search as in the second screenshot, where they search for :”sodium:” in the top search box, in this example, it produces 7 results. If the user wants to make a more specific search and uses the search within results box and inputs :”chloride:” as in the third screenshot, the application searches within the 7 results to see which of the substances contain sodium AND chloride, which brings the result down to 3 results.

**Notes for the future:**

In the future, the ability to do an additional chemical search within a search result is planned. Currently this can be done by using the advanced search features, but the user would have to redo the searches and filters there.

## New Feature: Improving Global Search Behavior Part 2: More intuitive wildcard searches

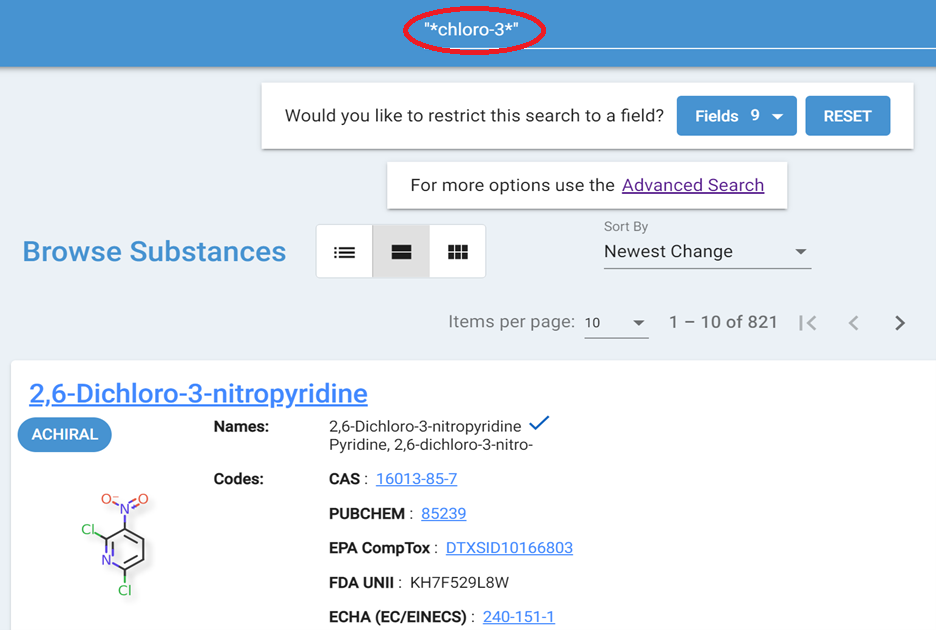


Figure - Global Searching

**Purpose and Motivation:**

With wildcard search, users can search successfully with query strings consist of letters and numbers besides the wildcard, such as "AQUILARIA\*" or "\*2019 CVR\*". However, before GSRS release 3.0.2, searching with a query string that contains hyphen, period, and ampersand would either cause error or get no results e.g., searching for "APT-101\*" will get an error. Searching with whitespace works fine when the string parts split by whitespace all have at least two characters. E.g., if you search with "\*ELICA IMP\*", you will mostly get the correct results. But if you search with "\*ELICA I\*", most of the time you will get an error.

Hyphen, whitespace, period and ampersand are used very often in the indexed fields of the substances. It would be very helpful to users to be able to search with these in the search strings. Starting GSRS release 3.0.2, wildcard searches contain :”- :”, :”.:”, :”&:” and whitespace(s) are supported. Now users can search fields of interests with :”- :”, whitespace, :”.:” and :”&:” in the values of the fields and get the right results.

**How it works:**

Both indexing process and searching process are changed to handle these special characters in wildcard search. Since the indexing process has been changed, users need to reindex the existing data first to make the search work as expected.

Some examples of simple forms:

**"APT-101\*"; "\*OAT.test"; "OAT&test\*"; “\*-OATTEST”; “^OATTEST\*”; “\*OAT Test$“**

Some examples of more complex forms:

**root\_names\_name:"APT-101\*"**

**root\_names\_name:"\*ELICA IMPERFECTA\*"**

**root\_names\_name:"\*ELICA IMPERFECTA\*" AND root\_codes\_code:"N0000171131"**

**Note:** The “AND” or “OR” that connects searching clauses needs to be capitalized.

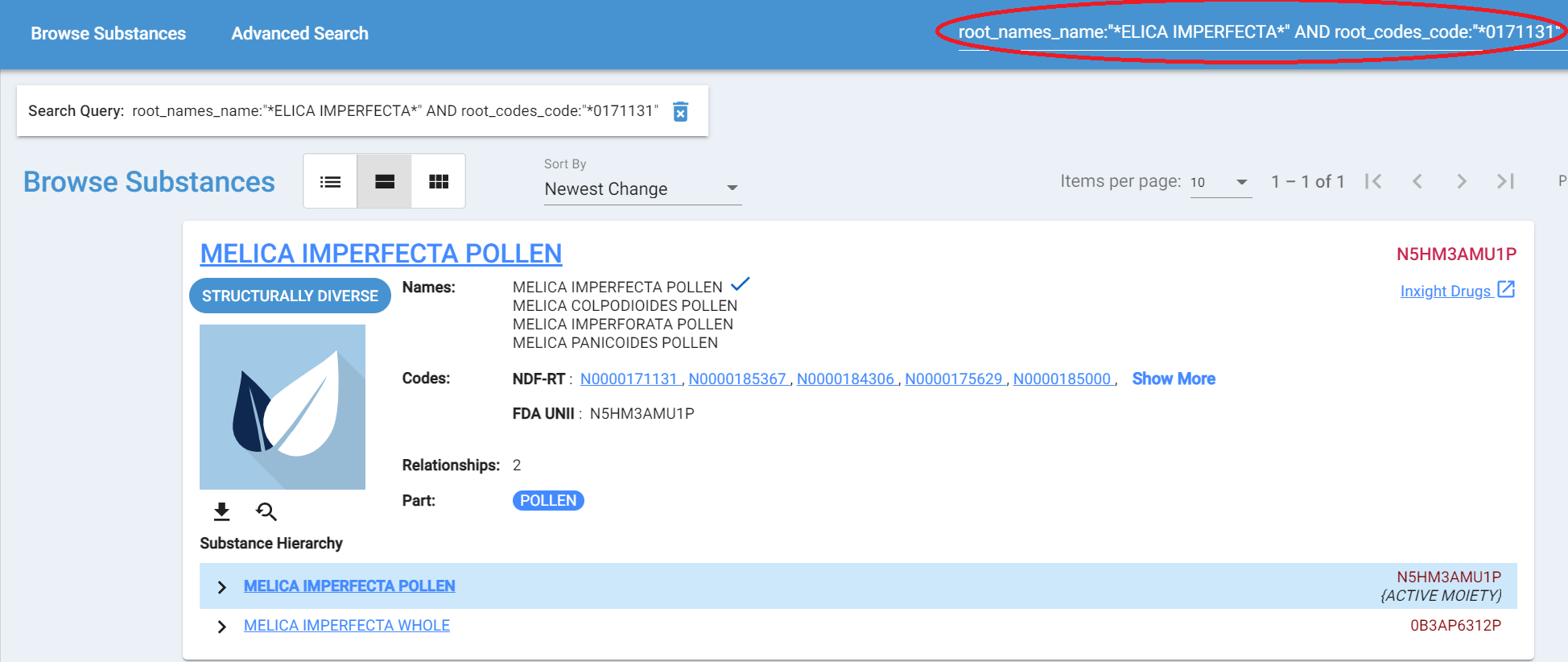


Figure - Using AND / OR

**Notes for the future:**

Certain protected characters that have special meaning or are ignored in indexing may be re-evaluated for indexing in the future. There may eventually be a supported feature of “string searches” which do a more verbatim form of indexing than the default word-based inverted indexes. This will be evaluated based on use cases brought to our attention.

## New Feature: Improving Global Search Behavior Part 3: More intuitive free-text searches

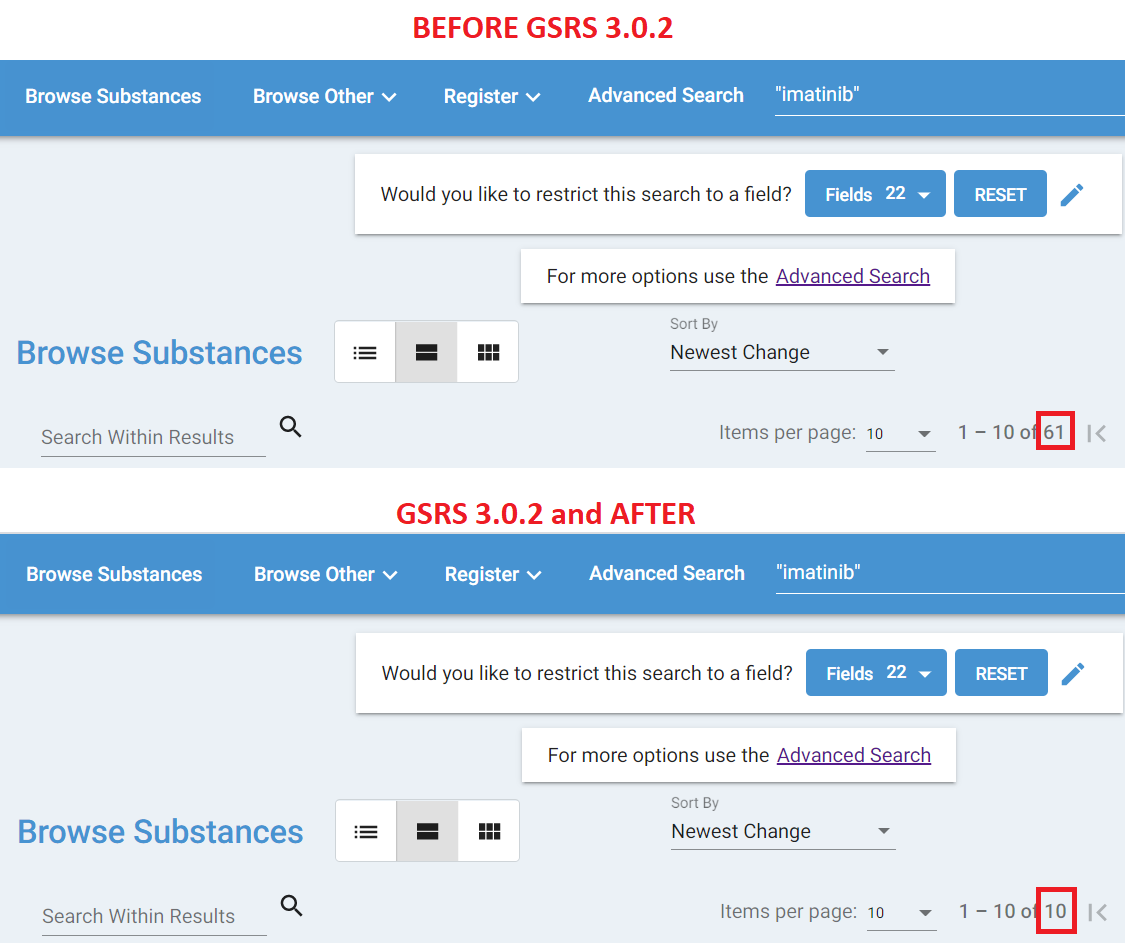
****

Figure - Multiple Results vs Streamlined Results

**Purpose and Motivation:**

Unqualified global searches return too many unexpected results. In previous versions of GSRS, all unqualified searches in the top substance search bar and the REST API would default to searching *all* text fields indexed for a substance record. A given substance record can have many text fields indexed, including fields for comments, notes, relationships, and references. All these fields are included in the default search. However, most global searches are motivated by users searching for substances based on a full or partial substance name or an identifying code. For searches like “ASPIRIN” or “IMATINIB” users tend to expect all results to have those terms in their names. For searches like “50-00-0” or “LR99Q6X62V” users tend to expect only records that directly contain these codes as identifier codes. Instead, GSRS would return all records that mention the text strings in any field, including references, notes, validation notes, and relationships. In practice this resulted in a lot of unexpected and unexplained results on a search.

The new search behavior is to preferentially search “identifier fields” over the more general text fields on unqualified global searches. This will return a more targeted set of results.

**How it works:**

The configuration file for an entity service deployment comes with a default set of identifier fields for each entity specified in the “ix.core.exactsearchfields” property. This property can be overwritten to specify other priority search properties as-needed. At indexing time all fields of this type are indexed a second time under an “identifiers” indexing field. If the new configuration property “gsrs.search.substances.restrictDefaultToIdentifiers” is set to true (default to true in 3.0.2 substance service) any unqualified global search will search the “identifiers” field rather than the more global “text” fields.

For users of the default substance configuration this means that text put into the top search bar in the UI will search only within names, certain codes, InChI keys, UUIDs, structure hashes and molecular formulas. However, field-targeted search suggestions built from matches on all other fields will still be displayed in the UI fields drop-down (the “narrowSearchSuggestions” section of the REST API response):

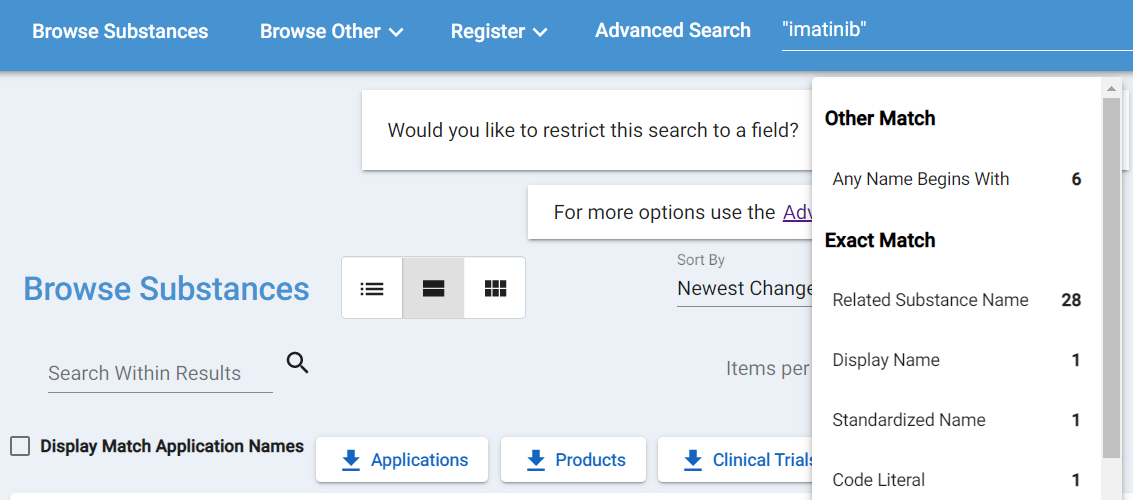


Figure - NarrowSearchSuggestions

For users that would like to have searches work as they did before, they must either change the configuration settings mentioned above, or they can qualify their searches with the keyword “text”. For example, `text:"imatinib"` will initiate the same search that was default in GSRS 3.0.1 and prior.

**Notes for the Future:**

As of GSRS 3.0.2 *only* the substances entity service default configuration is set to use this feature. However, this feature may be implemented for other entity services (products, applications, clinical trials, etc). In addition, this new indexing mechanism provides for some potentially significant performance improvements which may be explored in later releases.

# Other Issues Addressed

## Improvement: Corrected ORM settings for greater RDBMS compatibility, particularly with PostgreSQL

**Description:** Allow existing new and existing PostgreSQL databases to be used without cryptic console errors about “bytea” and “oid” in the vocabulary tables. In 3.X prior to 3.0.2, the dialect used for connecting to a PostgreSQL database was configured to store large binary objects in a way that was inconsistent with previous GSRS-generated tables. To fix this, DBAs would sometimes need to do some custom modifications for things to work as expected on PostgreSQL instances. GSRS 3.0.2 changes the dialect used when connecting to PostgreSQL databases in such a way that they remain backwards compatible with tables generated in 2.X and also avoid cryptic errors in the console. More information available in the “Database connection requirements for our RDBMSs” document on the git repository.

## Improvement: Improved auto-generated SQL indexes

**Description:** In GSRS 2.X, a set of hard-coded SQL commands for each database flavor was stored on the application and applied at initialization time to create certain useful database indexes. GSRS 3.X did not support this mechanism, and instead suggested in the RDBMS setup documentation that DBAs run these commands manually if desired. GSRS 3.0.2 includes standard JPA annotations that will allow schema generation to create these database indexes at initialization time. To apply the indexes to an existing database, the “spring.jpa.hibernate.ddl-auto” setting can be set to “update”. More information available in the “Database connection requirements for our RDBMSs” document on the git repository.

## Other selected Improvements:

* The facet filter side menu on substance browse is now expanded if a facet filter has been applied. Applied facets also show on top, as they had in GSRS 2.X.
* The term “Display Name” is now standard for the primary selected substance name. Previously the property had sometimes been called “Preferred Name,” “Preferred Term,” “Accepted Name,” and “PT” in various contexts. It is now always rendered as “Display Name” or “DN”.
* Other highlighted priority names, called “Additional Listing Name” in the edit forms, are now rendered in the details and browse views with special checkboxes.
* Names and codes on the browse view now allow users to quickly see and expand to all names and codes, rather than just showing the first few
* The substance selector widget on edit pages now allows searches for names with dashes, brackets, and special characters to work more intuitively. They had been returning strange results or failing to return anything.
* Products, Applications, and Adverse events UIs now allow for type-ahead in searches on the browse pages
* Products, Applications and Adverse Events entity services now allow for query-breakdown field suggestions as with substances
* Product export files now include all ingredients as individual rows rather than concatenated together
* Product-based indexes on substances based on ingredient roles have been fixed to properly capture the ingredient role in the products. They had previously been indexing many substances as active ingredients when they were not.
* Chemical moiety formulas sometimes rendered with extra unexpected characters, now rendered in standard format
* JSON import behavior made more intuitive
* Structure zoom popup view now shrinks to fit in browser page
* Matching Application count bubble (in the substance browse page) appears correctly now
* Warn users when display name changes, even in-place
* Able to add dropdown terms in CV in “impurity” module
* Standardized name algorithms improved to avoid strange character encoding cases
* Added scheduled task to report on / regenerate standardized name fields as needed