Sage 300 SDK

Global Search

August 2018

The MIT License (MIT)

Copyright © 2018 The Sage Group plc or its licensors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Contents

[1. Introduction 4](#_Toc520741487)

[2. Solr by Apache 5](#_Toc520741488)

[3. Installation 6](#_Toc520741489)

[3.1 Modules Folder 7](#_Toc520741490)

[4. Module Configuration Files 8](#_Toc520741491)

[4.1 Elements 9](#_Toc520741492)

[4.1.1 Document Element 9](#_Toc520741493)

[4.1.2 Drill Down 15](#_Toc520741494)

[4.1.3 Extra Element 16](#_Toc520741495)

[4.2 Delta Imports 19](#_Toc520741496)

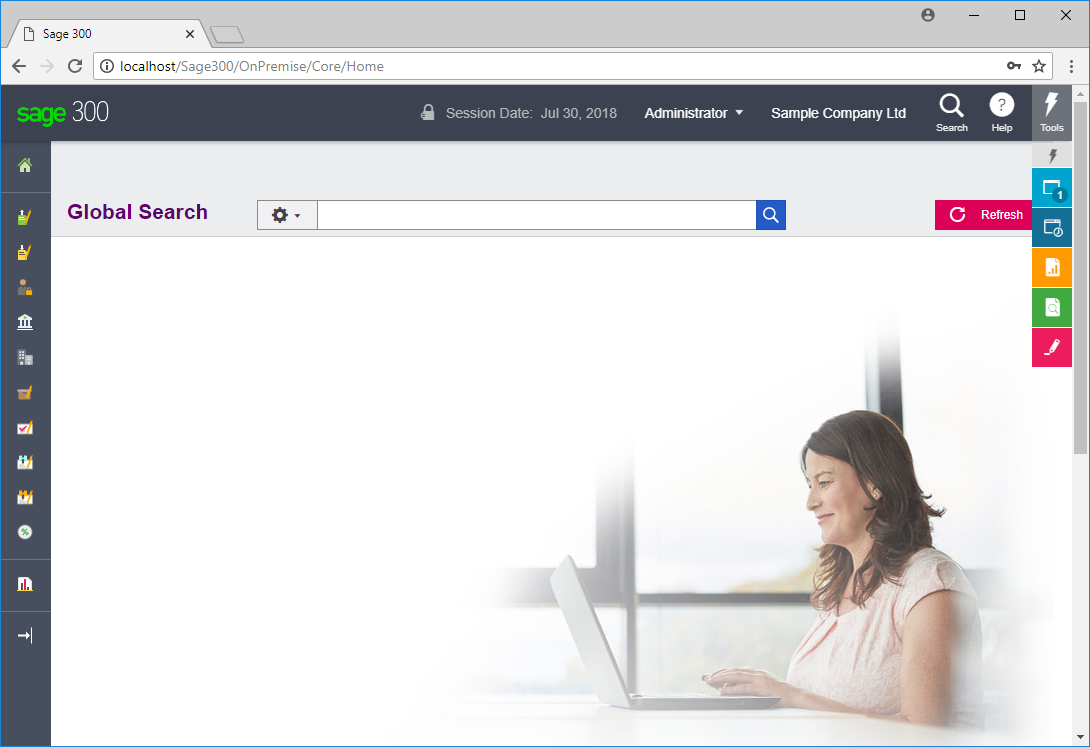
[4.2.1 Delta Import Frequency 19](#_Toc520741497)

[4.2.2 Delete Importing 19](#_Toc520741498)

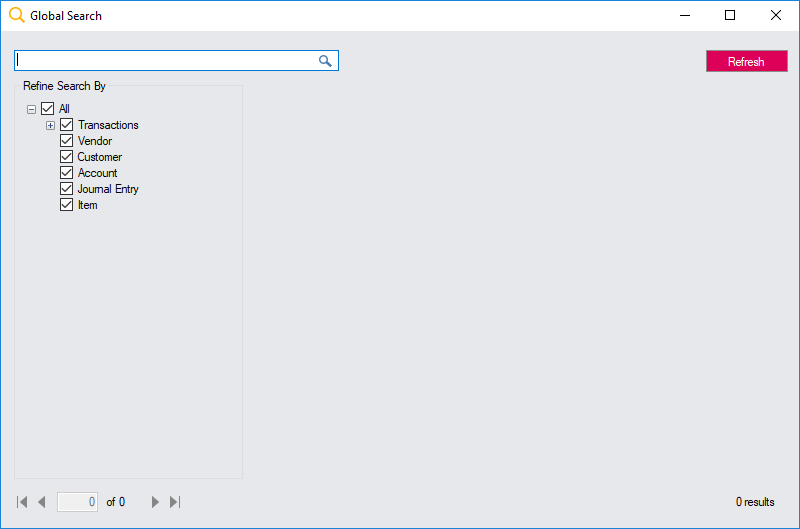
[5. Results Ordering 21](#_Toc520741499)

1. Introduction

Global Search is a new feature for Sage 300 2019. It has been implemented in both the Sage 300 Web Screens and the Desktop.



Web Screen Implementation



Desktop Implementation

It has been implemented with partners in mind. This means that Sage 300 exposes the ability for partners to participate in the feature.

Global Search uses the Solr full-text search server from Apache.

It is assumed that readers have basic programming knowledge with Visual Studio C# and an understanding of SQL and XML.

This document will explain the implementation as well as providing insight into how a partner participates in the feature.

1. Solr by Apache

**Solr** (pronounced "solar") is an [open source](https://en.wikipedia.org/wiki/Open_source) [enterprise search](https://en.wikipedia.org/wiki/Enterprise_search) platform, written in [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), from the **Apache**[**Lucene**](https://en.wikipedia.org/wiki/Lucene) project. Its major features include [full-text search](https://en.wikipedia.org/wiki/Full_text_search), hit highlighting, [faceted search](https://en.wikipedia.org/wiki/Faceted_search), real-time indexing, dynamic clustering, database integration, [NoSQL](https://en.wikipedia.org/wiki/NoSQL) features[[2]](https://en.wikipedia.org/wiki/Apache_Solr#cite_note-2) and rich document (e.g., Word, PDF) handling. Providing distributed search and index replication, Solr is designed for scalability and [fault tolerance](https://en.wikipedia.org/wiki/Fault_tolerance).[[3]](https://en.wikipedia.org/wiki/Apache_Solr#cite_note-3) Solr is widely used for enterprise search and analytics use cases and has an active development community and regular releases.

Solr runs as a standalone full-text search server. It uses the [Lucene](https://en.wikipedia.org/wiki/Lucene) Java search library at its core for full-text indexing and search, and has [REST](https://en.wikipedia.org/wiki/REST)-like [HTTP](https://en.wikipedia.org/wiki/HTTP)/[XML](https://en.wikipedia.org/wiki/XML) and [JSON](https://en.wikipedia.org/wiki/JSON) APIs that make it usable from most popular programming languages. Solr's external configuration allows it to be tailored to many types of application without Java coding, and it has a plugin architecture to support more advanced customization.

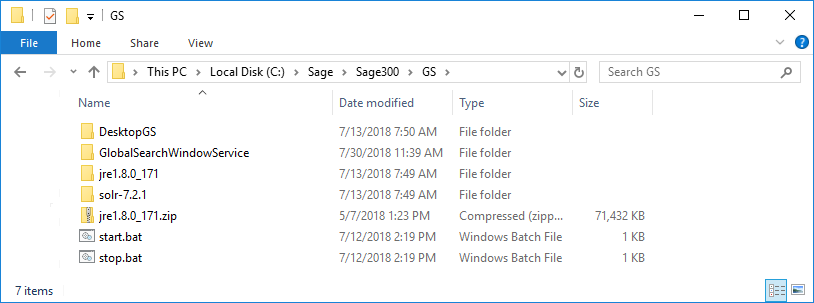
Apache [Lucene](https://en.wikipedia.org/wiki/Lucene) and Apache Solr are both produced by the same [Apache Software Foundation](https://en.wikipedia.org/wiki/Apache_Software_Foundation) development team since the two projects were merged in 2010. It is common to refer to the technology or products as Lucene/Solr or Solr/Lucene.



(<https://en.wikipedia.org/wiki/Apache_Solr>)

1. Installation

When Sage 300 is installed, all Global Search related folders and files will be found in the ***GS*** folder of the Sage 300 root directory.

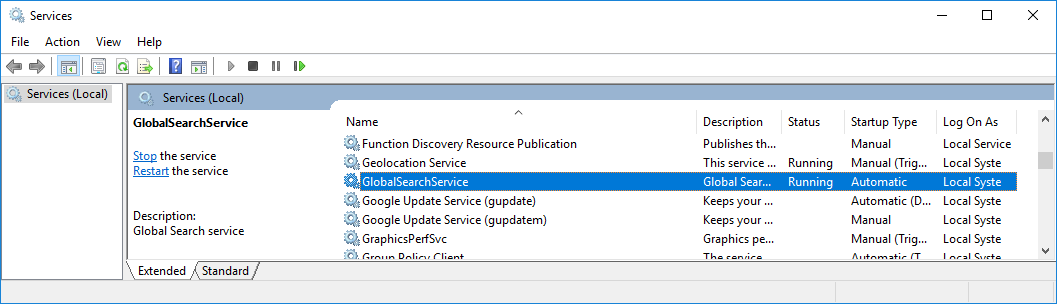


Example Installation Folder

This document will only address the contents of the ***GlobalSearchWindowService*** folder as this folder contains all the folders and files consumed by the service. This is the only area that will require modifications for affecting the Global Search feature.

Changes to Sage 300 metadata is not recommended for any folder or file ***except*** the files located in the modules folder. And, these should be modified with caution.

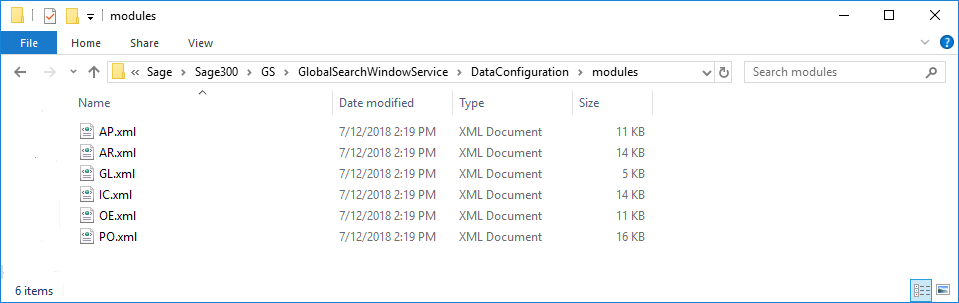
Additionally, the installation also installs the ***GlobalSearchService***.



Global Search Service

* 1. Modules Folder

The ***modules*** folder is where the metadata resides for the module configurations.



Module Configuration Files

These metadata files contain the Global Search configurations per module. Modifying these files will affect the searchable contents for each entity in the module.

Sage 300 module configuration files should be copied elsewhere for safety before modification.

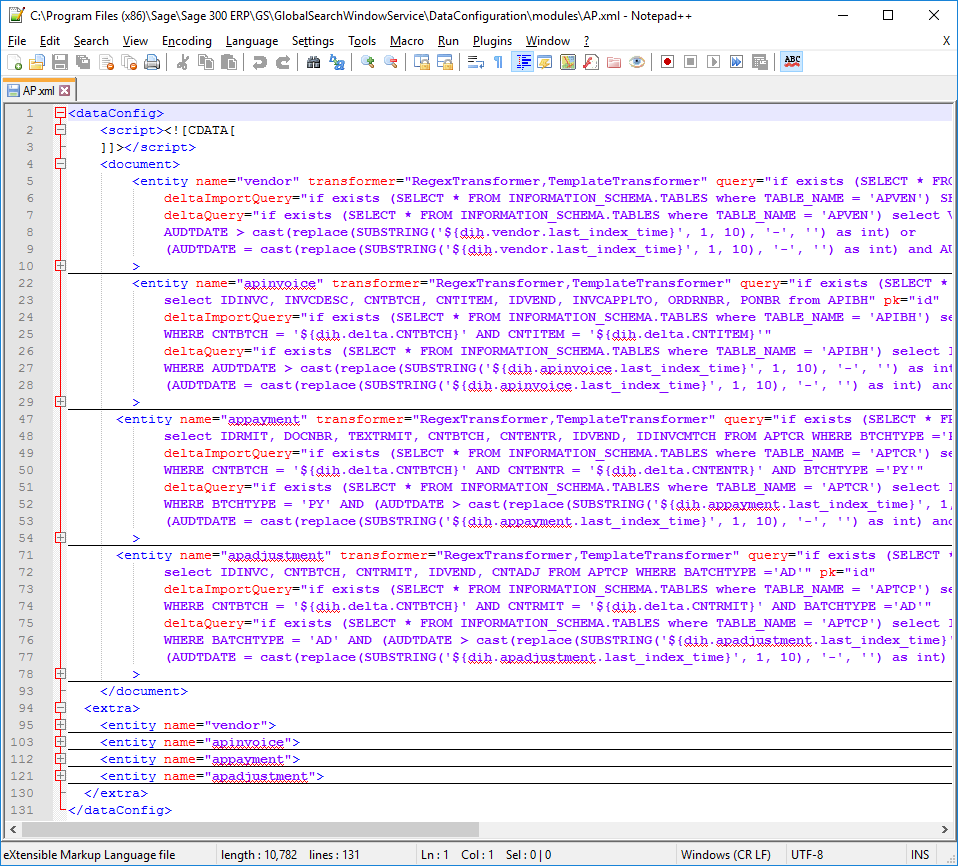
This is where a partner or ISV will add their module configuration file for their module to participate in the feature.

1. Module Configuration Files

In the modules folder, each XML file is associated to a module in Sage 300 for which there is a Solr search entity. This is just a naming convention and the name of the file is not important.

The file format is loosely based on the Solr data configuration file with extra contents that Sage 300 uses for security and display. This separation from the main Solr configuration file allows partners and ISVs to participate in the feature as well as simplifying the metadata that must be created or modified.

Let’s start by opening the ***AP.xml*** to show its contents.



AP.xml

* 1. Elements

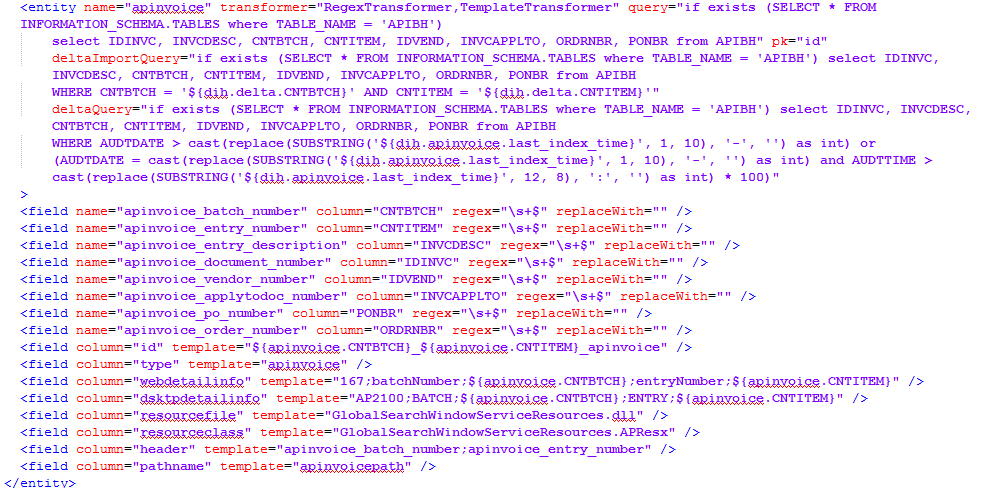
All module configuration xml files will have the root ***<dataConfig>*** element. Within that, there are the ***<script>***, ***<document>*** and ***<extra>*** elements:

* + script
    - Support using JavaScript to add or modify data after getting from database

Although using JavaScript to manipulate data is supported in Solr, the current design of Global Search does not see the need at this point. However, third parties or system integrators might be able to make use of it. Please refer to this link for additional details: [https://wiki.apache.org/solr/DataImportHandler#ScriptTransformer](https://wiki.apache.org/solr/DataImportHandler)

* document
  + All entities defined in this file
* extra
  + Contains all matching entities defined in the ***<document>*** element to support security, ordering, and localization.
    1. Document Element

The ***<document>*** element contains an ***<entity>*** element that uniquely defines an item to be searched:



apinvoice Entity

An Entity is composed of a list of attributes and fields.

Attributes

* name
  + A unique identifier ***across all entities*** in the Global Search metadata.

It is suggested that it start with the module abbreviation combined with its function

* transformer
  + For Global Search, only the ***RegexTransformation*** and ***TemplateTransformer*** are used.

Please refer to this link for additional details: <https://wiki.apache.org/solr/DataImportHandler#Transformer>

* query
  + An SQL statement to retrieve data for the full or initial import
* deltaImport
  + An SQL statement determine what data needs to be updated to the Solr database since last update (delta import)
* deltaImportQuery
  + An SQL statement to get the data that is returned from the deltaImport

Fields

Each field represent a piece of information about this entity. The value of the field will be for display or internal use.

* name
  + The name of the field. If not specified, the column name will be used
  + Suggested naming convention is ***{entity name}\_{unique descriptive name}*** (i.e. ***apinvoice\_vendor\_number***)
  + Some fields will not have the ***name*** attribute because they are mandatory fields for Global Search
    - id
      * Unique identifier for this record
    - type
      * The type of this record
    - webdetailinfo
      * Information for launching web screen
    - Dsktpdetailinfo
      * Information for launching desktop screen
    - resourcefile
      * Name of the resource file for localization
    - resourceclass
      * Name of the class within the resource file to be used
    - header
      * Value of header
    - pathname:
      * Value of the path
* column
  + The column name in the database table or a new field value as part of the entity
* template
  + To overwrite or modify any existing field. It is used to assign value (see below)
* regex
  + Regular expression to match trailing spaces
* replacewith
  + Based on the regex file to replace any trailing space with a value (empty string)

Template Field

The ***template*** field noted above requires some additional explanation.

It is used to transform or assign value to a field. Its value could either be derived from a static value or from other field(s).

In the ***apinvoice*** example, the field “***header***” combines the static value “***apinvoice\_batch\_number***” and “***apinvoice\_entry\_number***” to form the value “***apinvoice\_batch\_number;apinvoice\_entry\_number***”

Another example, the field “***id***” combines the value from batch number, entry number and the static value “\_***apinvoice***” to form the template ***“${apinvoice.CNTBTCH}\_${apinvoice.CNTITEM}\_apinvoice***”. The result in the field would look like “***1\_1\_apinvoice***”.

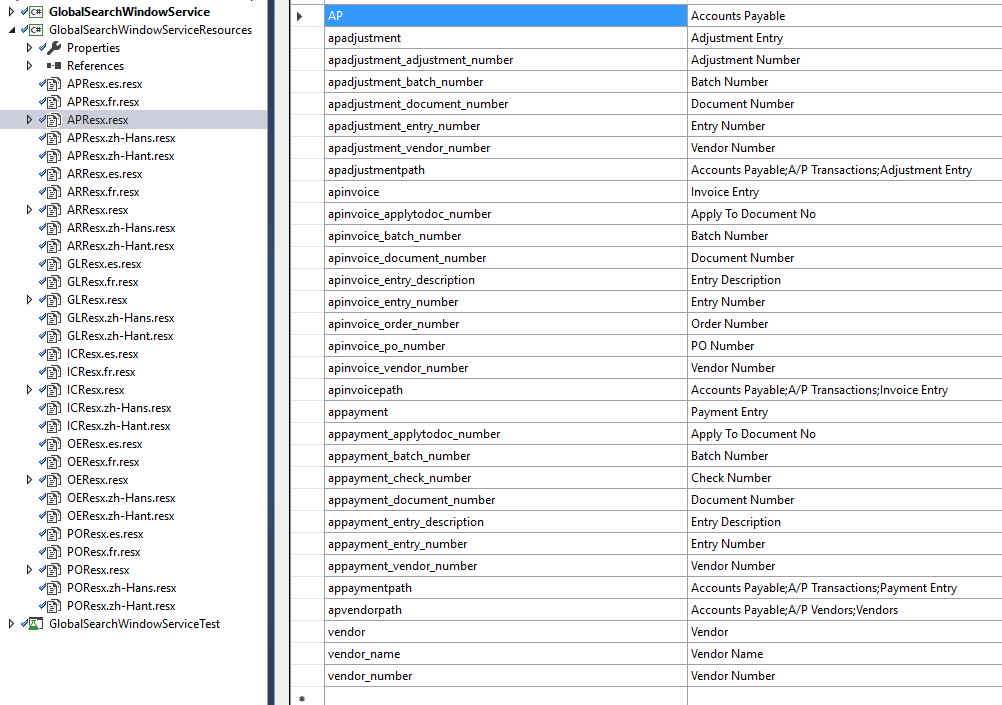
To illustrate, here is the AP Invoice data:



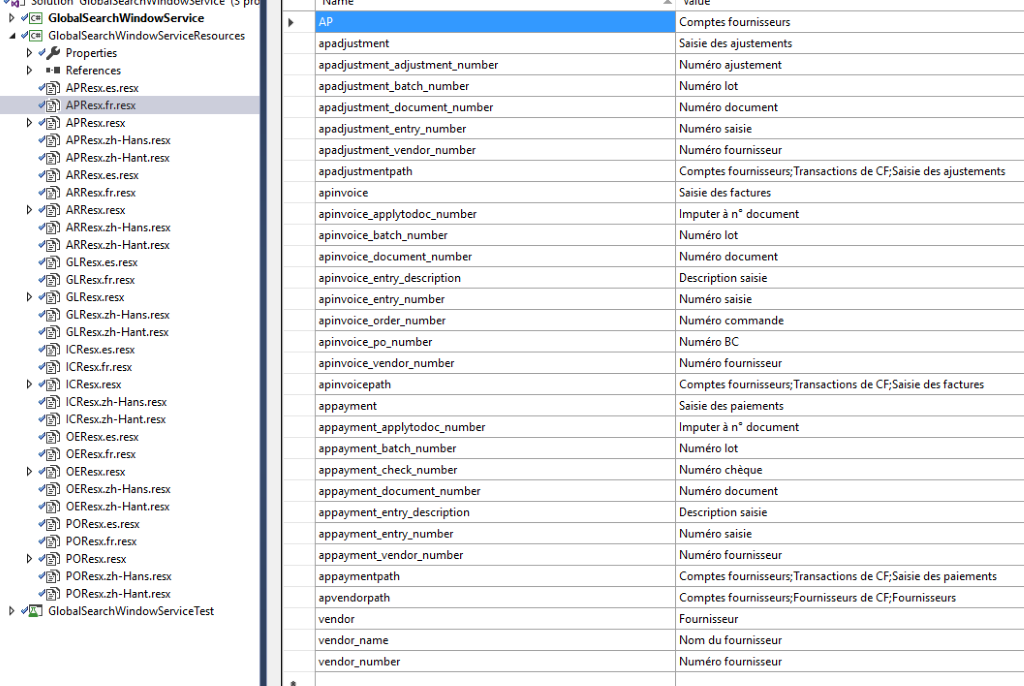
apinvoice Data

Resource Files

All label values will be translated and should be stored in separate resource files for different languages. If a label could not be found in the resource file specified, it will be defaulted to use the label value.

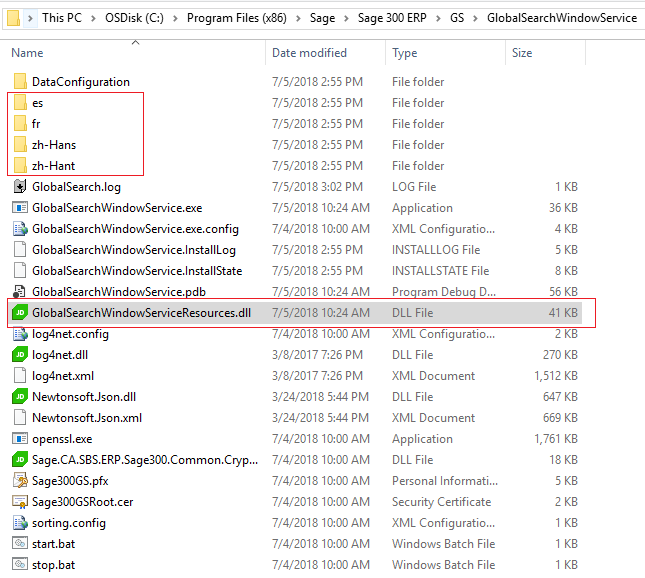


AP Resource File



AP Resource File (French)

The resource assembly needs to be placed in the ***GlobalSearchWindowService*** folder and language sub-folders for the non-English versions:



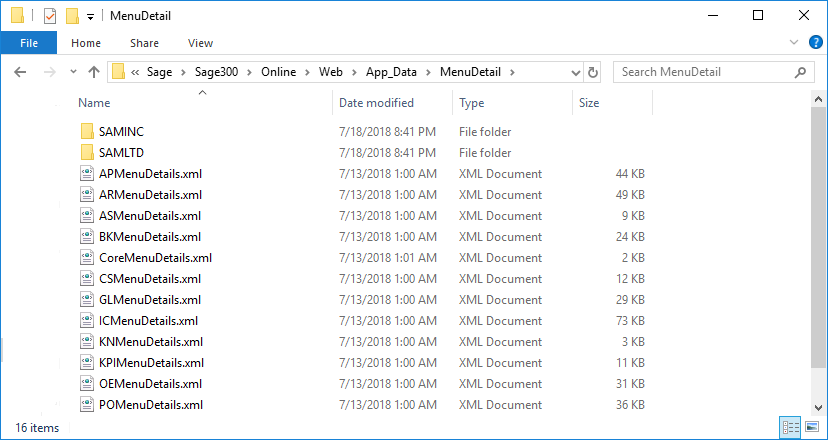
Resource File Locations

* + 1. Drill Down

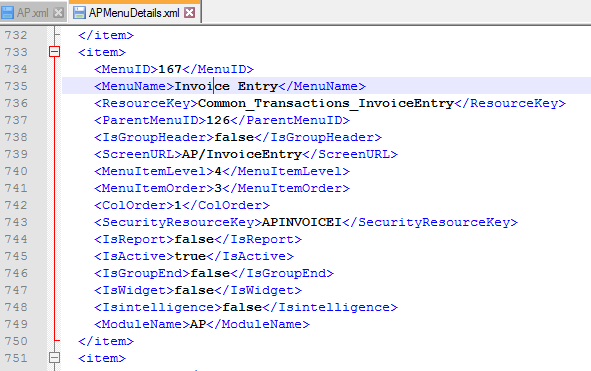
Launching the related screen for an entity’s result is supported by both the Web Screens and the Desktop.

Web Screens

The ***webdetailInfo*** attribute provides information on how to launch the screen once the search results are returned. It will start with the screen id specified in the menu file for the module followed by the identifier for a record.



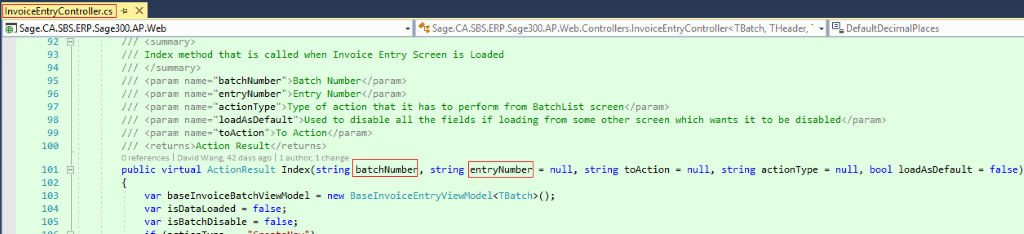
Menu Detail Folder Location



AP Invoice Menu Example

The Sage300 web screen must be able support how to open the screen with a given value(s).

The controller of the destination screen must be able to accept the arguments provided. In the AP Invoice example, ***webdetailinfo*** has the value ***167;batchNumber;1;entryNumber;1***. This means the value **1** and **1** will be send to the ***InvoiceEntryController’s*** ***Index*** method as the ***batchNumber*** and ***entryNumber***.



AP Invoice Controller’s Index Method

Desktop

The ***dsktpdetailInfo*** attribute provides information on how to launch the screen once the search results are returned. It will start with the Roto Id of the UI Object followed by the document keys.

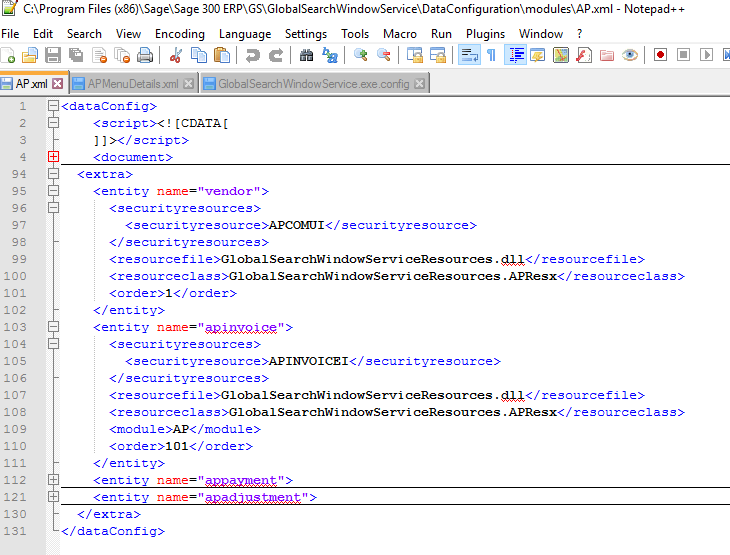
For more information on how drilldown works for desktop screens, please refer to the Drilldown document on the DPP Wiki (Appendix L).

* + 1. Extra Element

Each entity under the ***<extra>*** element will have a corresponding ***<entity>.***. This extra element allows for additional information in which to organize the value, manage security and filter display order.

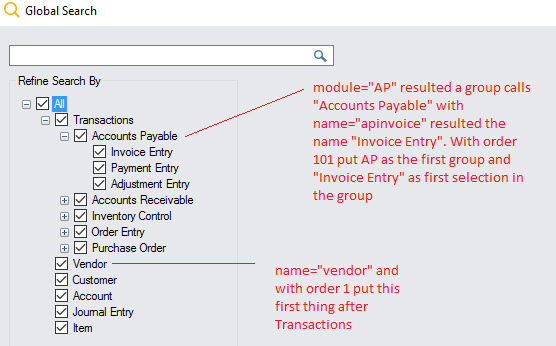
Values

* securityresource
  + Sage300 security access name.
  + Supports multiple resources within ***securityresources***. Must satisfy all the specified security resources to see the results
* resourcefile
  + Name of the resource file with localization
* resourceclass
  + Name of the class within the resource file to be used
* module
  + To define the name of the group (optional)
* order
  + To define the sequence how it should be displayed for the Global Search filter



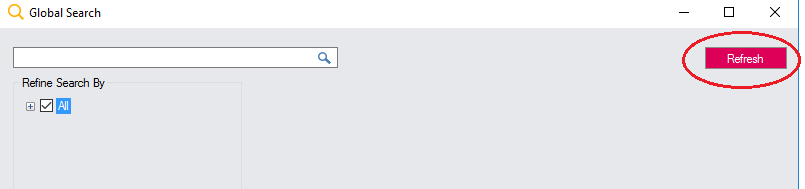
AP Extra Element Example

Therefore, the resulting Filter display will be altered simply by changing the values in the module configuration file.



Desktop Filter Display Example

To see the effect of either introducing a new configuration file or modifying existing ones, a refresh (i.e. full import) is needed. The is done by pressing the “Refresh” button on the Global Search screen.



Desktop Refresh Button

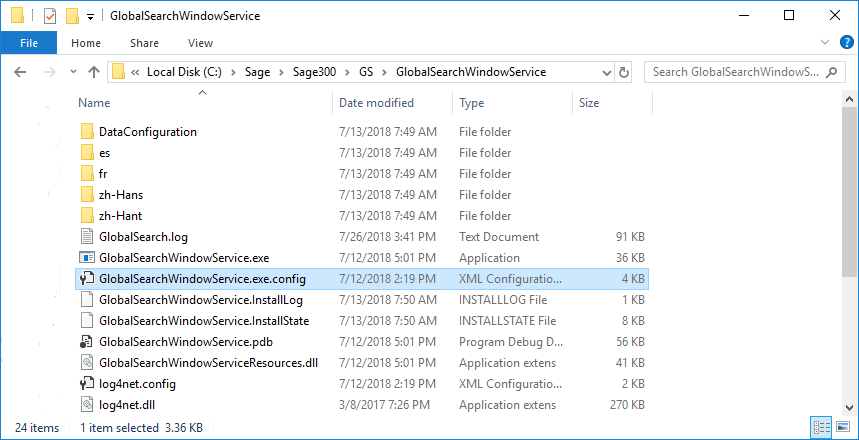
* 1. Delta Imports

Once the Solr database has been populated from the Sage 300 database, the Solr database will needed to be kept in-sync with changes made to the Sage 300 database.

* + 1. Delta Import Frequency

Global Search will periodically update its cached value to keep in sync with the Sage 300 database.

This value is stored in the ***GlobalSearchWindowService.exe.config*** file.



Config File Location

The update period is defaulted to ***10*** minutes or ***600*** seconds and it is configurable.

<!-- Value is in seconds -->

<add key="deltaUpdatePeriod" value="***600***"/>

Delta Import Frequency Configuration

This value will only take effect after the window service is restarted.

* + 1. Delete Importing

Delta importing consists of two aspects:

* Discovering the data that has been added or updated since the last delta import was performed
  + The SQL statement in the ***deltaQuery*** attribute of the ***entity*** tag is used
* Updating the added or updated data into the Solr database
  + The SQL statement in the ***deltaImportQuery*** attribute of the ***entity*** tag is used

Between each update (delta import), the last updated time stamp of this entity is stored in  ***dih.<entity name>.last\_index\_time*** (e.g. dih.apinvoice.last\_index\_time). It has the format of “YYYY-MM-DD HH:mm:ss” (e.g. “2018-07-05 16:04:48”)

This information is used against the Sage 300 database tables AUDDATE and AUDTIME columns to determine the data that has changed since the last updated was performed.

if exists (SELECT \* FROM INFORMATION\_SCHEMA.TABLES where TABLE\_NAME = 'APIBH')

select IDINVC, INVCDESC, CNTBTCH, CNTITEM, IDVEND, INVCAPPLTO, ORDRNBR, PONBR

from APIBH

WHEREAUDTDATE > cast(replace(SUBSTRING('${dih.apinvoice.last\_index\_time}', 1, 10), '-', '') as int)

or

(AUDTDATE = cast(replace(SUBSTRING('${dih.apinvoice.last\_index\_time}', 1, 10), '-', '') as int)

And

AUDTTIME > cast(replace(SUBSTRING('${dih.apinvoice.last\_index\_time}', 12, 8), ':', '') as int) \* 100)

AP Invoice Delta Query

This SQL statement is interpreted as: “If table ‘APIBH’ exist, then get all records with AUDTDATE greater than the timestamp. If AUDTDATE is the same as the timestamp, get ones that have greater AUDTTIME”

The result will be stored in dih.delta internally and will be used by the ***deltaImportQuery***.

if exists (SELECT \* FROM INFORMATION\_SCHEMA.TABLES where TABLE\_NAME = 'APIBH')

select IDINVC, INVCDESC, CNTBTCH, CNTITEM, IDVEND, INVCAPPLTO, ORDRNBR, PONBR

from APIBH

WHERE CNTBTCH = '${dih.delta.CNTBTCH}' AND CNTITEM = '${dih.delta.CNTITEM}'

AP Invoice Delta Import Query

1. Results Ordering

Global Search provides the ability to order the results based on their types.

In the ***GlobalSearchWindowService*** folder, the ***sorting.config*** file allows a field, a key value, to be specified from each entity thereby controlling the results based upon entity.

<appSettings>

<add key="searchResultOrder" value="apinvoice\_batch\_number,appayment\_batch\_number,apadjustment\_batch\_number,arinvoice\_batch\_number,arrefund\_batch\_number, arreceipt\_batch\_number,aradjustment\_batch\_number,gljournal\_batchnumber,icreceipts\_receiptnumber,icshipments\_shipmentnumber,icinternalusage\_internalusagenumber, icadjustments\_adjustmentnumber,ictransfers\_documentnumber,oeorder\_ordernumber,oeshipment\_shipmentnumber,oeinvoice\_invoicenumber,oecreditdebitnote\_documentnumber, porequisitions\_requisitionnumber,popurchaseorder\_ponumber,poreceipts\_receiptnumber,poinvoices\_invoicenumber,poreturns\_returnnumber,pocreditdebitnote\_documentnumber, item\_number,customer\_number,account\_formattednumber"/>

</appSettings>

sorting.config File