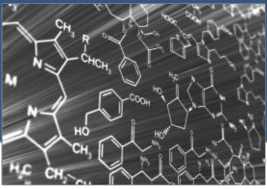
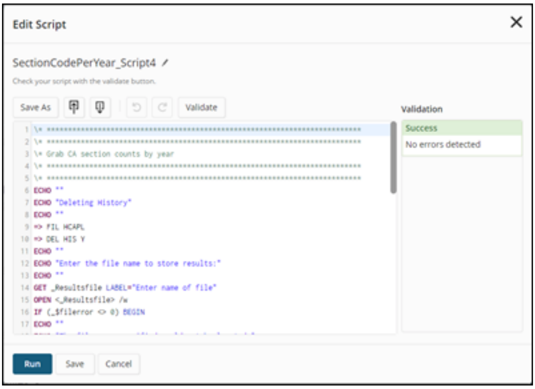
Using Scripts in CAS STNext

Streamline or automate parts of your search strategy

- Save time
- Stay up to date
- Share with colleagues





Edit Script

SectionCodePerYear_Script4

Check your script with the validate button.

```
\* **************************************************************************
\* **************************************************************************
\* Grab CA section counts by year
\* **************************************************************************
\* **************************************************************************
ECHO ""
ECHO "Deleting History"
ECHO ""
=> FIL HCAPL
=> DEL HIS Y
ECHO ""
ECHO "Enter the file name to store results:"
ECHO ""
GET _Resultsfile LABEL="Enter name of file"
OPEN <_Resultsfile> /w
IF (_$filerror <> 0) BEGIN
ECHO ""
```

Validation

Success

No errors detected

FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

CAS
A division of the
American Chemical Society

What are Scripts?

STN scripting is a mini programming language used to create repeatable and adaptable series of search commands. Scripts can be written offline in a simple text editor such as Notepad, or created directly in STNext.

Scripts can be useful if you:

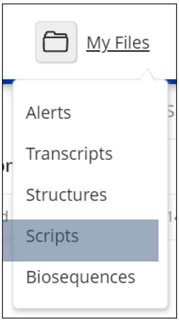
+ Conduct the same searches on a regular basis
+ Have a consistent overall search strategy, but substitute one or more variables when running the queries
+ Maintain a list of search terms (e.g., keywords, CAS Registry Numbers®) that are frequently used

Basic script commands

*	introduce a comment in the script
=>	introduce an executable command
ECHO	display text on screen while script running
\>_LNUM#	assign specific L-numbers to a search command within the script
DEL HIS Y	delete any previous L-numbers in your session. Resets to L1
OPEN	access previously saved data file from Scripts folders
READ	retrieve values or content from open data file
GET	present dialog box for user to enter value

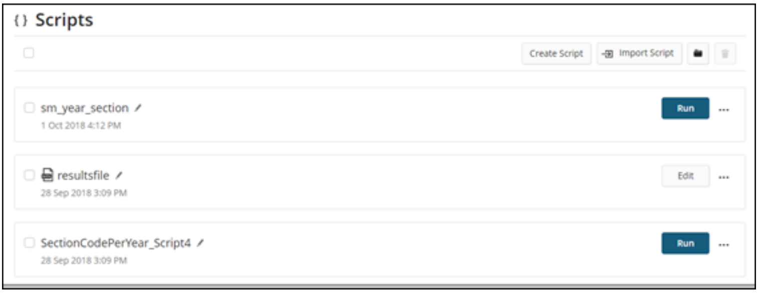
Managing Scripts and Data Files

STNext has a folder-based system for organizing files.



Scripts and related data files are stored in the My Files > Scripts folder.

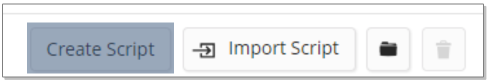
The Scripts section allows you to access saved scripts, edit them, run them or rename them. From here you can import a new script with .sc or .scb file extension. Or you can create a script in the editor.

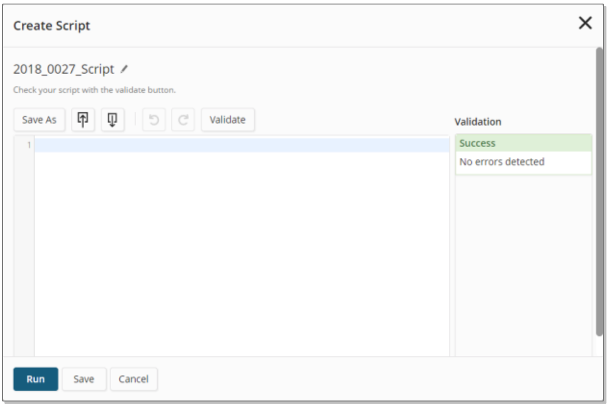


sm_year_section
1 Oct 2018 4:12 PM

resultsfile
28 Sep 2018 3:09 PM

SectionCodePerYear_Script4
28 Sep 2018 3:09 PM





2018_0027_Script

Check your script with the validate button.

Success

No errors detected

The Script Editor includes a validation tool to check the script syntax.

Basic Search with Synonyms

Any search you create in STN can be saved as a script. Identifying search terms can require a significant investment of your time. Preserve the knowledge you have gained by saving the variations in scripts (or data files) that you can then re-use or share.

```
\* This script provided without warranty
\* Example script with synonyms

=> DEL HIS Y

=> FILE HCAPLUS

=> S "TISSUE ADHESIVES" OR
"MEDICAL ADHESIVES" OR "SURGICAL
ADHESIVES" OR "SURGICAL TISSUE
ADHESIVES" OR "MEDICAL TISSUE
ADHESIVES" OR "ADHESIVES, BIOL.
TISSUE" OR "HEMOSTATIC ADHESIVES"
OR "MUCOADHESIVES"\>_LNUM1

=> S "Orthopedic prosthetics" OR
"Prosthetic implants"\>_LNUM2

=> S _LNUM1 AND _LNUM2\>_LNUM3

=> S _LNUM3 AND P/DT AND
PY.B>=2012\>_LNUM4
```

Recommend L-number reset at beginning of script.

Use " " quotes around terms that contain multiple words.

Note there is a character limit of 255 characters per command line.

You can group terms into multiple L-numbers and then combine the results for additional processing.

Search with Truncated Synonyms in Specific Fields

STN allows word stems and truncation variables when choosing search terms. This provides you the ability to retrieve search terms with multiple word endings or spelling variations.

Truncation options

?	zero to any number of characters at end of term
#	zero or one character at end of term
!	exactly one character within a term or at end of term

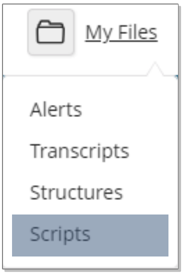
```
\* This script provided without warranty
\* Example script with truncation


=> FILE HCAPLUS

=> S (ETCH? OR PHOTOETCH? OR
CHASE# OR CHASING# OR ENCHAS? OR
ENGRAV? OR PHOTOENGRAV? OR EMBOSS?
OR PHOTOEMBOSS? OR INCISE# OR
INCISING# OR IMPRINT? OR IMPRESS?
OR ENCAUSTIC?)/TI,AB
```

You can specify search field(s) in the query, or you can allow STN to search the file's default field(s). In this example both the title (TI) and abstract (AB) will be searched for the terms.

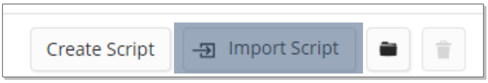
Using Data Files

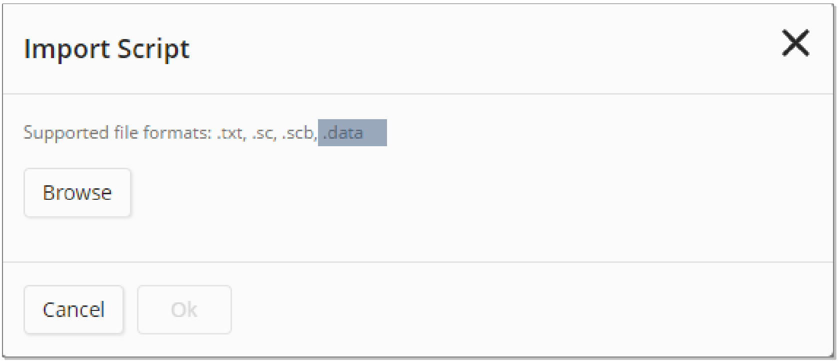
Store frequently used (or standardized) search terms in data files to keep the search values organized, easy to read into a script, and for sharing with others in your organization.

Create a file in a text editor such as Notepad, putting one value per line, and save the file with a .data extension. Then import the file into STNext for reference by scripts.



Scripts and related data files are stored in the My Files > Scripts folder.





Supported file formats: .txt, .sc, .scb, .data

NOTE: Scripts and related data files can be grouped and organized in STNext folders.

Using Structure Images

Scripts can access saved structure images from the My Files > Structures folder and upload them into your session.

Use the Structure Editor to name and save the structure images.
Options include:
+ Import a .cxf, .mol or .str file
+ Convert CAS Registry Number, SMILES or InChI into a drawing
+ Draw the substance structure using the edit tools

In your script, use the UPLOAD command to specify the structure image you want to use:

```
=> FILE REGISTRY
UPLOAD LNUM _lnum1 <pantoprazole>
```

The result of the UPLOAD is an L-numbered structure query. In this instance, LNUM is indicating the beginning of variables being assigned, and _lnum1 indicates that this upload will be referred to as _lnum1 from this point forward in the script. This allows the uploaded L-number to be utilized again in the script, for example:

```
=> s _lnum1 sss full
```

If you have organized your structure images into folders, include the folder name in the command. For example, if the folder name was “drugs,” the command would include that folder name before the substance structure name:

```
UPLOAD <drugs/pantoprazole>
```

Basic Loops and Writing Results

Scripts can save you time by automating repetitive commands, compiling the results, and writing those results to a data file that you can review later. This example loops through multiple years and multiple CA sections (e.g., Section 1 is Pharmacology) to monitor trends over years.

```
 \* This script provided without warranty
 \* Example script with loop and datafile


=> DEL HIS Y

=> FILE HCAPLUS

ECHO “Enter filename to store
results:”

GET _Resultsfile LABEL=”Enter
name of file”

OPEN <_Resultsfile> /w
```

GET commands present a dialog box to the user to collect the variable information.

```
IF (_$filerror <> 0) BEGIN
   ECHO “The file name specified
   could not be located.”
   ECHO “Exiting script.”
   EXIT
END
```

Optional error detection routine with ECHO messaging helps expose problems when running script.

```
_year = 2008
_yearlimit = 2017
@YEARLOOP
 => SET RANGE =_year
 IF (_year = _yearlimit) BEGIN
   ECHO “Processed all years.”
   EXIT
 END
```

Script continued...

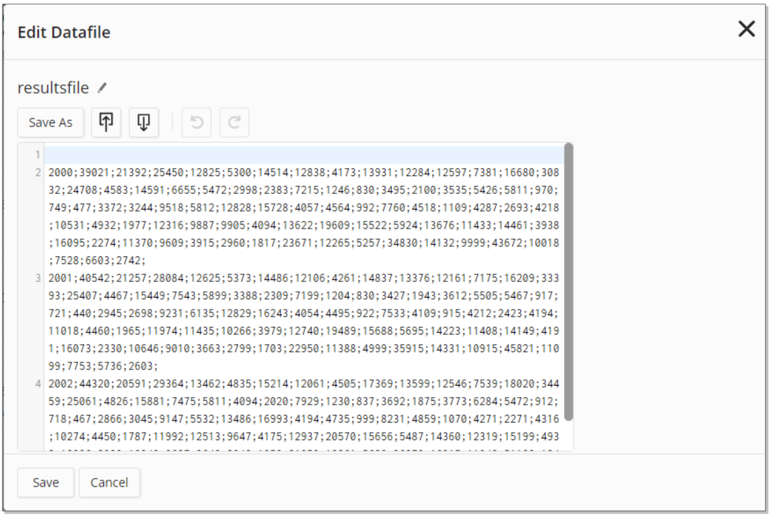
```
 WRITE _year NOCR
 WRITE “;” NOCR

 _section = 1
 _sectionlimit = 81
@SECTIONLOOP
 IF (_section = _sectionlimit) BEGIN
   ECHO “Processed _section sections.”
   GOTO @JUMP
 END
```

NOCR prevents the script from jumping to a new line for each entry.

```
 => S _section/SC
 WRITE _$LANS NOCR
 WRITE “;” NOCR
 _section = _section + 1
 GOTO @SECTIONLOOP

@JUMP
 => DEL HIS Y
 _year = Year + 1
GOTO @YEARLOOP
```

After searching the section number, the record count is written to the data file, and the section number is incremented.

This script writes the record count to the data file for later review.



Edit Datafile

resultsfile

```

2000;39021;21392;25450;12825;5300;14514;12838;4173;13931;12284;12597;7381;16680;308
32;24708;4583;14591;6655;5472;2998;2383;7215;1246;830;3495;2100;3535;5426;5811;970;
749;477;3372;3244;9518;5812;12828;15728;4057;4564;992;7760;4518;1109;4287;2693;4218
;10531;4932;1977;12316;9887;9905;4094;13622;19609;15522;5924;13676;11433;14461;3938
;16095;2274;11370;9609;3915;2960;1817;23671;12265;5257;34830;14132;9999;43672;10018
;7528;6603;2742;
2001;40542;21257;28084;12625;5373;14486;12106;4261;14837;13376;12161;7175;16209;333
93;25407;4467;15449;7543;5899;3388;2309;7199;1204;830;3427;1943;3612;5505;5467;917;
721;440;2945;2698;9231;6135;12829;16243;4054;4495;922;7533;4109;915;4212;2423;4194;
11018;4460;1965;11974;11435;10266;3979;12740;19489;15688;5695;14223;11408;14149;419
1;16073;2330;10646;9010;3663;2799;1703;22950;11388;4999;35915;14331;10915;45821;110
99;7753;5736;2603;
2002;44320;20591;29364;13462;4835;15214;12061;4505;17369;13599;12546;7539;18020;344
59;25061;4826;15881;7475;5811;4094;2020;7929;1230;837;3692;1875;3773;6284;5472;912;
718;467;2866;3045;9147;5532;13486;16993;4194;4735;999;8231;4859;1070;4271;2271;4316
;10274;4450;1787;11992;12513;9647;4175;12937;20570;15656;5487;14360;12319;15199;493
```

Retrieve Monomer RNs and then Search References

Scripts can help make your searches more comprehensive. This example uses POLYLINK to retrieve relevant CAS Registry Numbers and then use them to search across multiple files.

```
 \* This script provided without warranty
 \* Example script with multiple files and
    variable search terms


=> DEL HIS Y

=> FILE REGISTRY

\* Search known RNs to find all
   relevant RNs

=> S 24980-41-4 or 26100-51-6\>_lnum1
=> POLYLINK _lnum1\>_lnum2
=> DEL SEL Y
=> SEL _lnum2 RN
=> D SEL
=> QUE E1-_$ENUM\>_lnum3
```

_$ENUM variable represents the last E-number and allows dynamic inclusion of all E-numbers from the EXPAND.

```
\* Take RNs and search multiple files

=> FILE CAPLUS EMBASE

=> S (3D OR THREE DIMENSION? OR
ADDITIVE DEPOSIT? OR BIOACTIVE INK OR
BIOINK OR BIOPRINT OR BIO PRINT OR
BIOADDITIVE MANUF?)/TI,AB \>_lnum4

=> S _lnum4 and _lnum3\>_lnum5
```

Script continued...

```
=> S _lnum5 and (BONE OR TISSUE OR
SCAFFOLD OR ORGAN OR MUSCLE OR HEART OR
REGENERAT? OR IMPLANT OR CARTILAGE)\
>_lnum6

=> S _lnum6 AND P/DT AND PY>=2017\>_lnum7

=> S _lnum6 NOT _lnum7\>_lnum8

=> S _lnum8 and PY>=2017\>_lnum9

=> DUP REM _lnum9\>_lnum10
```

Assigning L-numbers to subsets of the results (using \>_lnum#) allows you to isolate records based on characteristics of interest, and also return to any of the answer sets for new analysis.

About POLYLINK

The POLYLINK command retrieves the monomer-based and SRU (Structural Repeating Unit)-based CAS Registry Numbers for a given condensation polymer or set of condensation polymers in a CAS REGISTRY answer set.

POLYLINK may be used on:

+ An individual CAS Registry Number
+ An E-number containing a CAS Registry Number
+ Any REGISTRY answer set
+ Any ANALYZE list of CAS Registry Numbers

Discovering Search Term Variations

Scripts can be written to assist in monitoring literature across multiple resources. This example verifies a preferred drug name, supplements the search with CAS Registry Number and synonyms, and then removes duplicates from the results.

```
 \* This script provided without warranty
 \* Example script with multiple files and
    variable search terms

=> DEL HIS Y
=> FILE HCAPLUS
=> SET PLU OFF
=> SET EXPANDER RENUMBER
=> SET DUPORDER FILE
```

Scripts can edit user preferences such as handling plurals or restarting the E-numbers.

```
ECHO “Please enter earliest Entry Date
(YYYYMMDD)”
GET _Timeframe Label= “Enter earliest
Entry Date in YYYYMMDD format”

ECHO “Please enter a drug name”
GET _CAPLUSNAME Label= “Enter drug name”

ECHO “Retrieving drug names from HCAplus”
=> e _CAPLUSNAME+use/ct
=> e _$ENUM+UF/CT
=> s e1-_$ENUM/BI AND ED>=_Timeframe\
>_lnum2

ECHO “Retrieving the CAS Registry Number
for your drug name”
=> FILE REGISTRY
=> S _CAPLUSNAME/CN
=> SEL RN\>_lnum3
```

Script continued...

```
=> FILE HCAPLUS
=> S (_lnum2 OR _lnum3) and ED
>=_Timeframe\>_lnum4

=> FILE EMBASE
ECHO “Retrieving drug names from Embase”
=> e _CAPLUSNAME+use/ct
=> s _$ENUM
=> e _$ENUM+UF/CT
=> s (e1-_$ENUM/BI OR _lnum3) AND
ED>=_Timeframe\>_lnum5

=> FILE MEDLINE
ECHO “Retrieving drug names from Medline”
=> e _CAPLUSNAME+xuse/ct
=> s e1
=> s (e1-_$ENUM/BI OR _lnum3) AND
ED>=_Timeframe\>_lnum6

=> FILE HCAPLUS EMBASE MEDLINE
=> s _lnum4 OR _lnum5 OR _lnum6\>_lnum7

=> SET PLU LOGIN
=> SET EXPAND LOGIN
=> DUP REM _lnum7\>_lnum8

ECHO “Enter DISPLAY command including L-
number, display format (e.g., BIB ABS)
and number of records (enter 1- for all)”
```

Read Search Terms from a Data file

Scripts can read values out of data files to streamline creating queries with large numbers of known search terms (such as CAS Registry Numbers, patent numbers or common chemical names). This example reads CAS Registry Numbers from a file and parses them into workable groups.

```
\* This script provided without warranty
\* Example script reads data file and groups
   search terms


=> DEL HIS Y

=> FILE REGISTRY

_maxRNs = 15
_innerCount = 0
_storeSearch = “S”
_storeFinal = “S”

ECHO "OPEN Solvent RN file"
OPEN <SOLVENTRNS> /R
```

This script uses variables and counters to include hundreds of CAS RNs in a search query by creating groups of 15 values.

```
@MAINLOOP
 READ _searchterm
 ECHO “Read _searchterm”
 IF _searchterm = “*****” THEN
 GOTO @FINALSEARCH
```

The data file has a string used to signal the end of reading search terms.

```
 _join = “”
 IF _innercount > 0 _join = “OR”

 _storeSearch = _storeSearch + _join
 _storeSearch = _storeSearch + _searchterm
```

Script continued...

```
 _innerCount = _innerCount + 1
 IF _innerCount = _maxRNs THEN
 BEGIN
   => _storeSearch

   _innerCount = 0
   _storeSearch = “S”
   IF _storeFinal <> “S” THEN
   BEGIN
   _storeFinal = _storeFinal + “OR”
   END
   _storeFinal = _storeFinal + _$LNUM
 END
 GOTO @MAINLOOP

@FINALSEARCH
 IF _storeSearch <> “S” THEN
 BEGIN
   => _storeSearch

   IF _storeFinal <> “S” THEN
   BEGIN
   _storeFinal = _storeFinal+ “OR”
   END
   _storeFinal = _storeFinal + _$LNUM
 END

 => _storeFinal
 => D CN 1-
 :Y

 CLOSE
 ECHO "Script complete"
 EXIT
```

Here we are programmatically adding the “OR” operator between L-numbers.

For more information...

CAS
help@cas.org

Support & Training:
www.cas.org/support

FIZ Karlsruhe
helpdesk@fiz-karlsruhe.de

Support & Training:
www.stn-international.de