Regular Expressions

The Download Overlay component includes the ability to select products with regular expressions. This article gives extensive examples of how to use regex effectively.

On this page...

- The What and Why of Regular Expressions
- Common Regex Syntax
- An Important Distinction: * is not a wildcard
- Example Searches
- For Further Reading..

The What and Why of Regular Expressions

A regular expression (often called a regex) is a powerful way to search text for matches. At the most basic level, it functions similarly to the "control /command + f" search function in most text editors and web browsers; any exact match with the "simple string" you enter will be returned. The advantage of regex is its specialized syntax, which allows for complex queries beyond simple 1:1 matching. This page will list the syntaxes available to you in the search form and walk through some examples of regex queries.

Regex is the filter of last resort!

It is likely that the filter you are attempting to apply can be found within the Download Overlay itself; you should start there.

Before you attempt to use advanced regex syntax in your search for files, you should try searching as you would in a browser or document. Often, a more complex query is not necessary to match your target filenames.

Common Regex Syntax

Not all regex syntax is valid for a filename search. The most useful syntaxes are included in the table below.

Syntax	Meaning
-	Matches any character
*	The preceding character appears zero or more times
+	The preceding character appears one or more times
\$	Matches the end of a filename
[aeiou]	Matches any character in the listed set
[^XYZ]	Matches any character not in the listed set
[a-z0-9]	The set of characters can include a range or multiple ranges

If you want to use one of the special characters literally, you must 'escape' it. For example, if you actually wanted to include a period in your search query, you should enter "\."

An Important Distinction: * is not a wildcard

One of the most common mistakes in regex is using * as a generic wildcard character. As is shown in the example table below, the * character is always interpreted in conjunction with the preceding character.

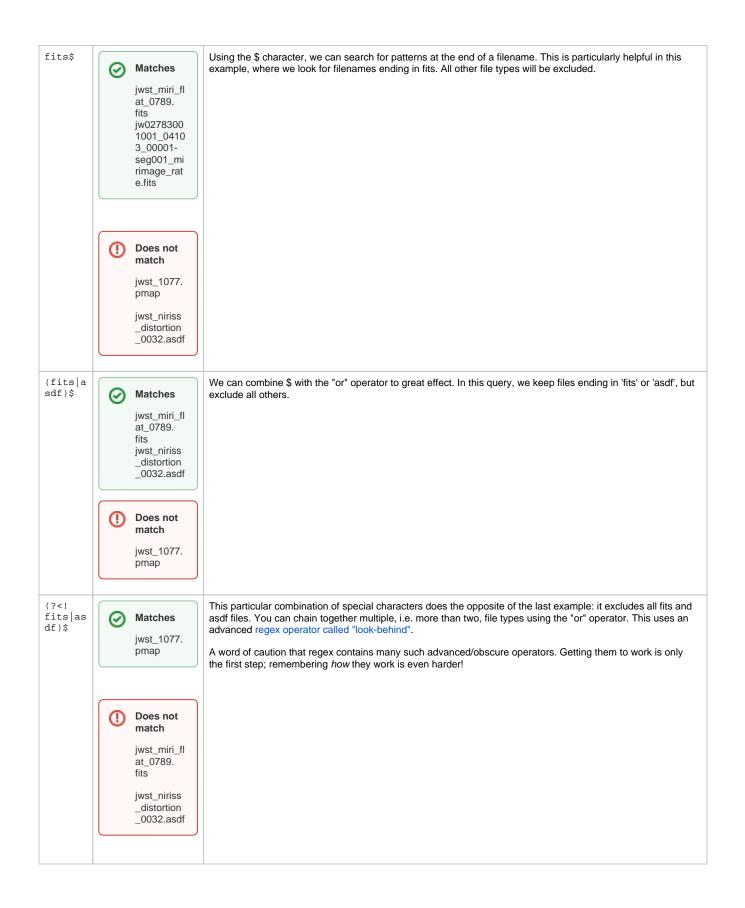
The correct syntax for a wildcard is either ".*" or ".+", depending on whether the absence of a character should be included. See the last two columns of "regex input" for help clarifying this subtle point.

	Regex input			
Filename	jwst*file	jwst.file	jwst.*file	jwst.+file
jwstfile.fits	Match	No match	Match	No match
jwsttttfile.fits	Match	No match	Match	Match
jwstXfile.fits	No match	Match	Match	Match

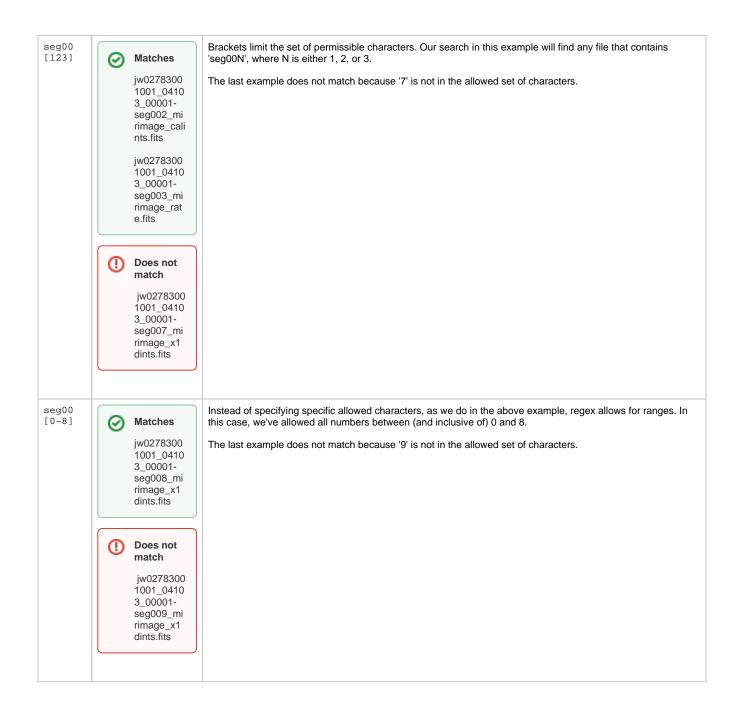
Example Searches

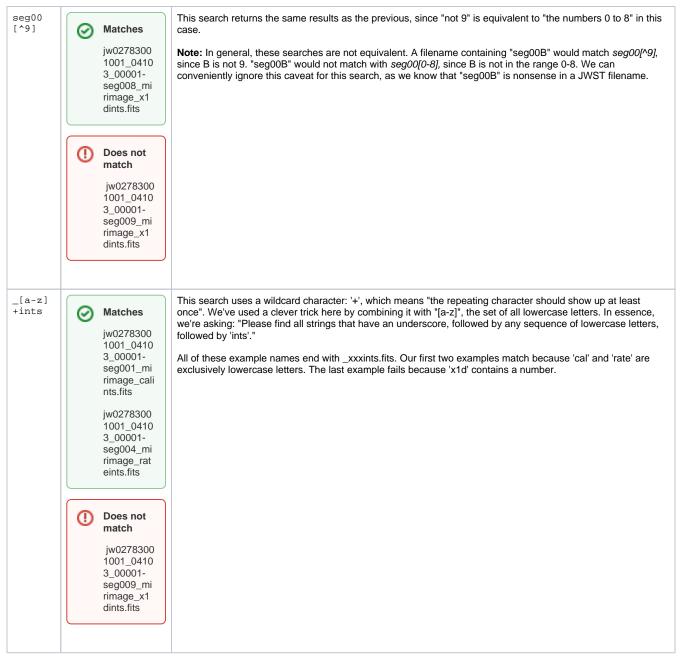
These examples use real filenames produced by the JWST mission. All text in the Regex Input column is "raw"; that is, you should enter it into the search bar exactly as you see it. Where special characters are used, they either take advantage of regex functionality or are part of the filename.

Regex Match? Input		Notes			
rate	 Matches jw0278300 1001_0410 3_00001- seg001_mi rimage_rat e.fits jw0278300 1001_0410 3_00001- seg001_mi rimage_rat eints.fits 	A regex search iterates through the filenames and returns matching strings. In this basic case, we've asked for all files that contain the phrase 'rate' somewhere within the name. It does not matter where this appears within the name; a simple string search is equivalent to a search in most text editors and internet browsers. The last example does not match because it does not have "rate" anywhere in its name.			
rate\. fits	Matches jw0278300 1001_0410 3_00001- seg001_mi rimage_rat e.fits Does not match jw0278300 1001_0410 3_00001- seg001_mi rimage_rat eints.fits jwst_miri_fi lteroffset_0 006.asdf	Although it's recommended to filter file types using the UI, it's also possible to filter using regex. The period is a special input that matches any character in our search string, so we escape it here using a backslash. It is good practice, but unnecessary in this case, to escape the period. There are no files with names like "rate9fits" that would cause an accidental match.			



 \odot





For Further Reading...

• Regex101 is an excellent resource to "practice" these queries. The download overlay uses JavaScript syntax.