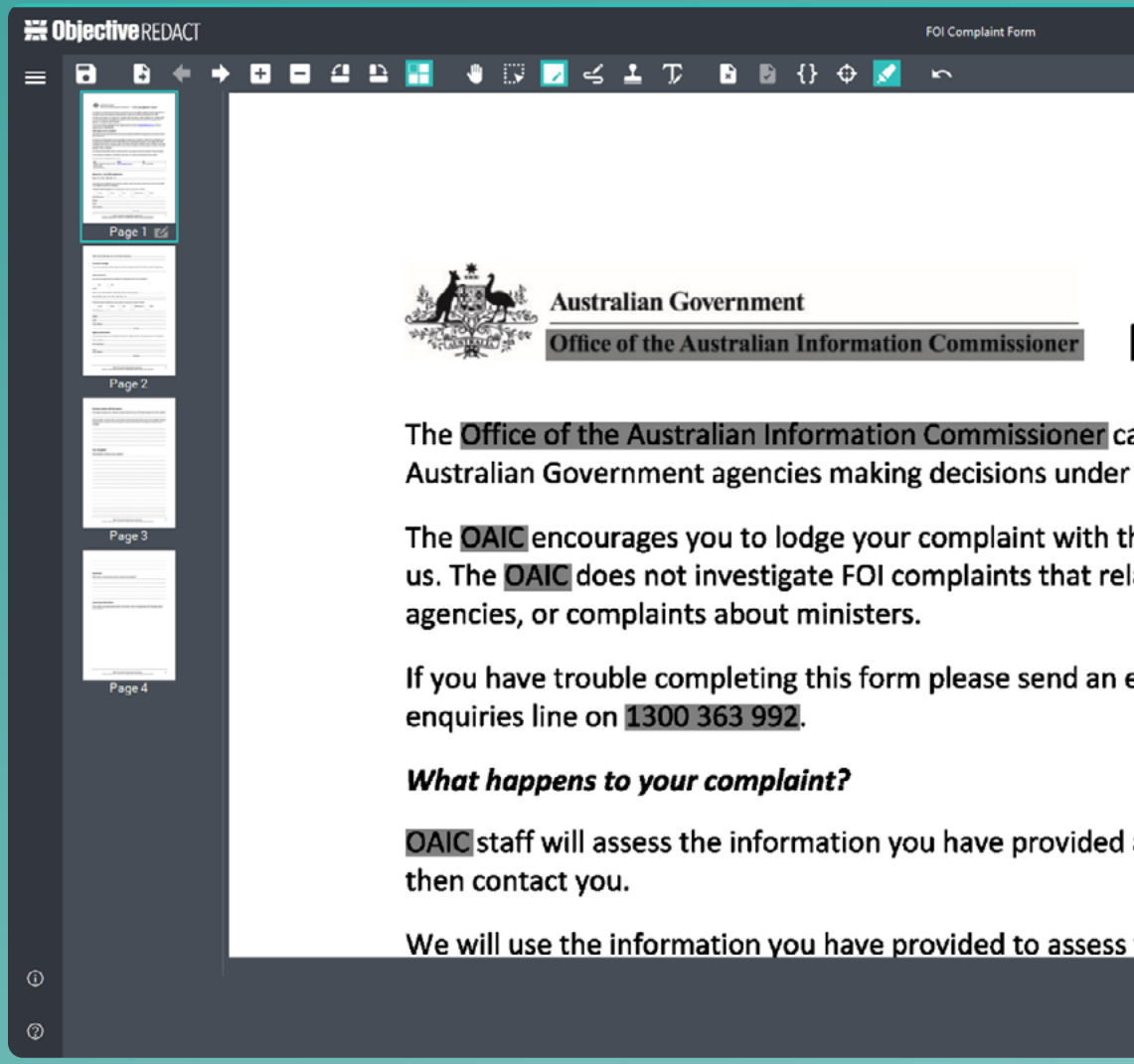
Objective REDACT

GETTING STARTED GUIDE

Advanced Redaction with Objective Redact

Powerful tools to redact a range of document types.



objective.com/redact

Table of Contents

Introduction

Objective Redact comes with powerful tools for redacting structured patterns, words and phrases and logos and images.

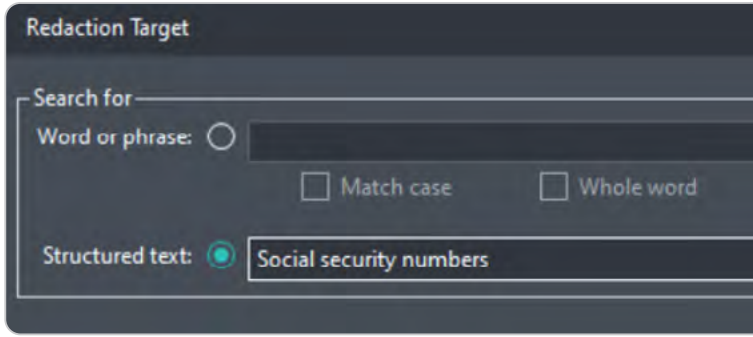
These tools make it easy to redact all document types and most documents easily and quickly. Some users, however, need to redact high volume and complex documents and may come across patterns for which Redact doesn't provide out of the box target expressions to redact.

For these advanced redaction targets, Objective Redact enables users to write regular expressions to create new target redactions specifically for their needs. Just like the target redactions all users are familiar with – social security numbers, email addresses and others – the specific target redactions users can create also leverage the OCR technology for accurate search and automatically remove all associated meta data.

Here we will provide you with an introduction to advanced target redaction using regular expressions along with useful examples from some of our favourites and a syntax summary to get you started. If you find this too challenging, please contact us with your target redaction challenge and we'll work out the regular expression for you.

Let's get started!

Objective Redact enables users to write regular expressions to create new target redactions specifically for their needs.



Regular Expressions and why are they useful?

Regular expressions (Regex) were invented by mathematician Stephen Keene in the 1950s and are used heavily in search engines and document editing applications.

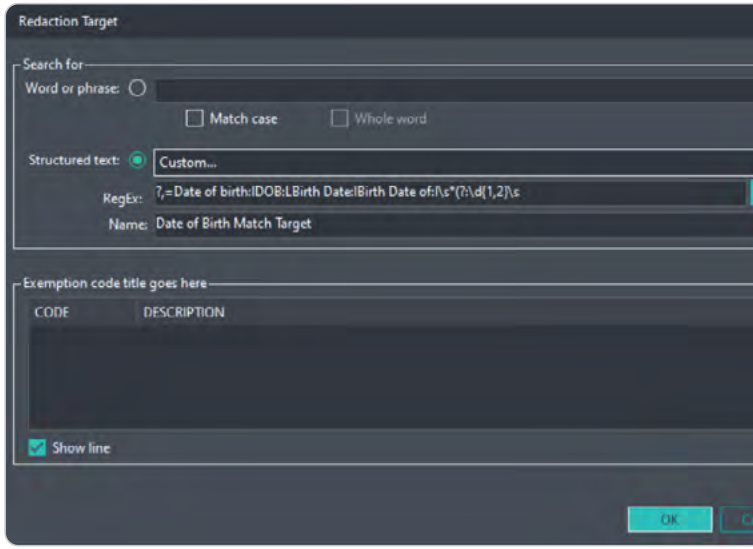
In simple terms Regular Expressions (Regex) are a way for you to describe a pattern of text that you want to find and redact from your documents. For example, email addresses, credit card numbers, SSN are all structured text patterns that Objective Redact includes by default and can easily be added to your redaction targets list. But what if you want to customize these further, or write your own pattern? That's where Regex comes in.

For example, Redact includes a structured text target for automatically finding and redacting email addresses. This will redact all email addresses, but what if you want to only redact emails from a specific company or domain?

E.g. match anything@objective.com but not anything@anyotherdomain.com

That would be a good use case for regex, let's look at how this can be done using regex.

A familiar use of Regex is in "find and replace" technology in many of the applications you use every day.



Regular Expression Syntax

Regex has a reputation for being a tricky topic so don't worry if you find the concepts challenging at first. It will take some practice to come up to speed but armed with our examples and the handy syntax summary that follow, you'll be a regex wiz in no time.

The key to understanding regex is learning how to express the different types of characters that may appear in text, like letters, numbers and symbols.

Believe it or not, a regular expression can be as simple as abc123. Literally the text "abc123" is a valid regular expression that will match the exact text sequence abc123. In regex plain letters and numbers can be used where you want to match those exact literal characters. Also keep in mind the pattern is treated as a whole, so abc123 would match, though abc or 123 alone do not match the entire pattern sequence.

Sometimes we need to match text more generically, maybe we want to match any number not just number 1 2 or 3 specifically. We can do this using a metacharacter \d, which means any digit. Or any alphanumeric character with \w (that includes letters, numbers and underscore _). Metacharacters are the building blocks used in regex for matching a range of characters or special characters, you can find a description for all the common metacharacters in the syntax summary.

We are off to a good start now, we can match any text or digits literally by entering the exact text, and we can match digits, or alphanumeric characters by using \d or \w. Though what if we need to match a sequence of digits, perhaps an account number includes 6 digits, how could we express that?

We could use \d\d\d\d\d\d 6 individual digit characters, or we could make use of quantifiers e.g. \d{6} to match exactly 6 digits. We can also use quantifiers to match a range, between min and max. e.g. \w{3,6} match alphanumeric characters in between 3 and 6 characters long.

The first thing to grasp with regex is that all text is made up of characters, and when regex searches text it checks for matches character by character.

How about if we have a range of letters and numbers that may be valid, but not just any letter or digit. Maybe we want to redact a code only if the last character is any of: f x 6 7 8 9. For this we can use sets [], which define valid characters or ranges within square brackets. E.g. [fx6-9] this means characters f x and numbers 6 through 9. You can also use [a-zA-Z0-9] for all lower- and upper-case letters, and all digits. Or you can even do [^xyz] to match all characters except x y or z, in a set the ^ symbol means negation.

Great, we are starting to build our repertoire for expressing patterns.

Let's circle back to the email address example and examine how this can be approached.

Check out the Regular Expression Syntax Summary at the end of this Getting Started Guide to see all the various regex syntax that is available.

Redacting emails from a specific domain using regex

How do we match anything@objective.com but not anything@anyotherdomain.com?

regex: [a-zA-Z0-9.-]*@objective\.com

Let's examine this regex to better understand how it works.

[a-zA-Z0-9.-] This first set specifies which characters will be matched, you will notice it allows some ranges of alphabetical characters, a to z in lower case, and A to Z in upper case, digits 0 to 9, and the period . and hyphen - characters.

*** the set is followed by a quantifier;** this quantifier says match zero or more characters from the set. This means any length of text is allowed provided the characters are from the specified set. E.g. anne.burton byron-robson would be valid text.

@objective\.com and lastly, literal text characters specify that exactly "@objective.com" must follow the preceding character matches.

It's worth mentioning here all characters that require escaping to use their literal form, rather than the special regex meaning. These are: ^.[$()|*+?{\

Which includes the backslash character, since that itself is used for "escaping" other characters.

To use the backslash character in a literal sense, we would apply a backslash firstly to escape and secondly as a backslash. It would be presented as \\.

Notice how the . period character was prefixed with a backslash \.

This is because in regex the period symbol . is a special character that normally means match any character, though in this case we want a literal period as in .com.

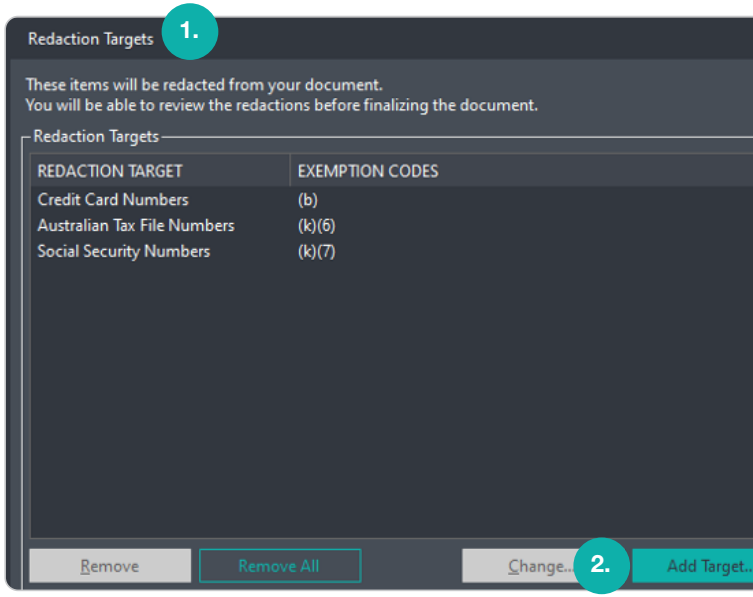
This is often referred to as escaping a character for its literal meaning.

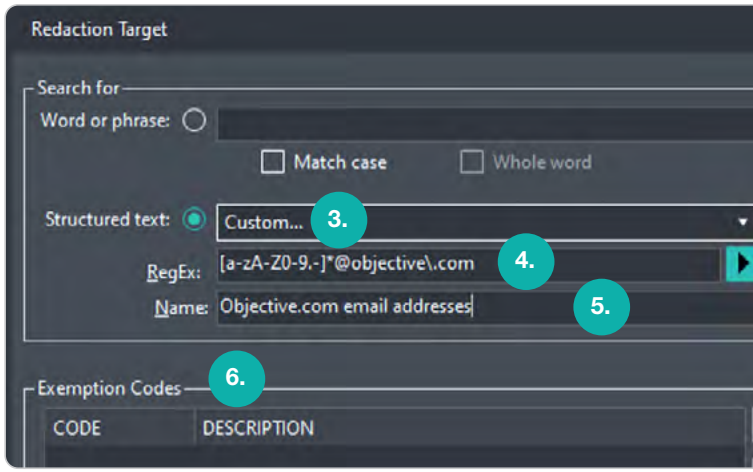
How to enter custom regular expressions in Objective Redact

1. Open a document that you want to Redact, then click on Redaction Targets
2. Click “Add Target”… as pictured
3. Click on Structured text, then from the dropdown list choose Custom…
   Here you are presented with two fields, RegEx and Name.
4. In Regex, enter the regular expression (right-click to paste in copied text).
   E.g. [a-zA-Z0-9.-]*@objective\.com
5. In Name, enter a descriptive name for your redaction target. This is the name that will appear in the redaction target list. E.g. Objective.com email addresses
6. Optionally add an exemption code (reason for redaction).
   We can now see our custom regex target has been added to the list.

Click OK to apply the redaction targets to your document

We can now see in our document the regex has been applied and matches only emails ending @objective.com, mission accomplished!





Useful Advanced Targets using Regular Expressions

Here are a sample of useful advanced targets our customers have challenged us with and may be useful to you.

To use them, simply select the expression closest to achieving your aim and modify content to suit. If there is more you want to achieve with the expression, please refer to the syntax summary that follows to enhance your expression even further.

Regular Expression	Explanation
(?:^|(?<=\s))(?!first\.person@example\.com|second\.person@example\.com|third\.person@example\.com)(\w[\w\.-]*@\w+\.[\w\.-]+)\ba	Match all email addresses except for those specified by a group e.g. exclude first.person@example.com second.person@example.com third.person@example.com Match any other email address
(?<=Date of birth:|DOB:|D.O.B.:|Birth Date:|Birth Date of:|Birthday of:)\s*(?:\d{1,2}\s?[-\\/]?\s?\d{1,2}\s?[-\\/]?\s?\d{4}|(?:January|February|March|April|May|June|July|August|September|October|November|December),?\s?\d{1,2},?\s?\d{4})	Match any dates that are prefixed by Date of birth, in any of the following formats; Date of birth: 1-1-1984 DOB: 1-1-1984 D.O.B.: 1-1-1984 Birth Date: 1-1-1984 Birth Date of: 1-1-1984 Birthday of: 1-1-1984
\d{3,4}\s?-?\d{3,4}\s?-?\d{3,4}\s?-?(?=\d{4})	Credit card numbers, excluding last 4 digits **Note** that using the built-in credit card redaction does perform checksum validation which may provide fewer false positives when matching credit cards.
([a-zA-Z,#/ \.\(\)\-\+*]*[0-9]){7}[0-9a-zA-Z,#/ \.\(\)\-\+*]*	Generic phone numbers regular expression (broader than the default US Phone numbers)
\b(?!finance|account|trademark)[\w\-\.,\\/\(\)]+\b	Redact all words, except those specified In the finance, account, and trademark are excluded from being captured / redacted, all other words are redacted.
(?!12345)\d{5}	Redact any 5 digits, except specifically 12345 Matches 11111, 12346, 54321 Won't match 123456

Regular Expression Syntax Summary

REGEX	MATCHES
abcd	Letters a to z, match the exact characters
0123	Digits 0 to 9, match the exact digits
\d	Any digit
\w	Any alphanumeric character
.	Any character
\.	Literal period character, escaping the . with a \ means a literal period, rather than any character
\D	Any non-digit character
\W	Any non-alphanumeric character
\S	Any character other than whitespace
Quantifiers	**These are used to quantify a character, set or group.**
?	Zero or 1 e.g. cats? makes the s character optional, matches both cat and cats
*****	Zero or more
+	1 or more
{n}	Exactly n characters
{n,}	At least n characters
{n,m}	Between n and m characters, at least n and at most m

REGEX	MATCHES
Escaping	**Escaping means to use the literal character, instead of the special regex meaning of a character. E.g. the . period character normally means "any character", escaping as \. means literal period character**
^.$|*+?\(){}	Characters that must be escaped, prefix with a backslash \ to use the literal character. E.g. \$ literal dollar sign
Specified sets	**Define a set of characters within square brackets [] that will be matched, ranges of characters such as a-z lowercase A-Z uppercase and 0-9 digits can be specified**
[abcd]	Match character any character of abcd, e.g. a matches e does not match, aa does not match, to match aa use a quantifier such as [abcd]{1,2}
[a-zA-Z0-9]	All lower case and upper case alphabetical characters, and numbers, but no other symbol characters such as $ - etc.
[aeiou]	Any characters of a e i o u
[a-z-[aeiou]]	Any alphabetical characters, except a e i o u
[^$-]	Any character except $,
Boundaries	**Boundaries are useful in cases where you don't want to find partial matches within other text, or only want to find matches at the beginning, or ending of words.**
\b	First or last character in a word e.g. cat\b will match cat, but not within catalog
\B	Not first or last character in a word e.g. cat\B will match cat within catalog, but not cat by itself

ABOUT OBJECTIVE CORPORATION

Objective creates information and process governance solutions that are effortless to use and enable organisations to confidently advance their own digital transformation.

Designed for regulated industries, these solutions turn the imperative of compliance, accountability and governance into an opportunity to streamline business processes and deliver the innovative services that customers expect.

With a heritage in Enterprise Content Management (ECM), Objective's expanded solutions extend governance across the spectrum of the modern workplace; underpinning information, processes and collaborative work-spaces.

Through a brilliant user experience, people access the information they need to progress processes from wherever they choose to work.

Objective