

JPEG

From Wikipedia, the free encyclopedia

In computing, **JPEG** (/ˈdʒeɪpeɪɡ/ *JAY-peg*)^[1] (seen most often with the **.jpg** or **.jpeg** filename extension) is a commonly used method of lossy compression for digital images, particularly for those images produced by digital photography. The degree of compression can be adjusted, allowing a selectable tradeoff between storage size and image quality. JPEG typically achieves 10:1 compression with little perceptible loss in image quality.

JPEG compression is used in a number of image file formats. JPEG/Exif is the most common image format used by digital cameras and other photographic image capture devices; along with JPEG/JFIF, it is the most common format for storing and transmitting photographic images on the World Wide Web. These format variations are often not distinguished, and are simply called JPEG.

The term "JPEG" is an acronym for the Joint Photographic Experts Group, which created the standard. The MIME media type for JPEG is *image/jpeg* (defined in RFC 1341), except in older Internet Explorer versions, which provides a MIME type of *image/pjpeg* when uploading JPEG images.^[2]

JPEG/JFIF supports a maximum image size of 65535×65535 pixels,^[3] hence up to 4 gigapixels (for an aspect ratio of 1:1).

Contents

- 1 The JPEG standard
- 2 Typical usage
- 3 JPEG compression
 - 3.1 Lossless editing
- 4 JPEG files
 - 4.1 JPEG filename extensions
 - 4.2 Color profile
- 5 Syntax and structure
- 6 JPEG codec example
 - 6.1 Encoding
 - 6.1.1 Color space transformation
 - 6.1.2 Downsampling
 - 6.1.3 Block splitting
 - 6.1.4 Discrete cosine transform
 - 6.1.5 Quantization
 - 6.1.6 Entropy coding
 - 6.2 Compression ratio and artifacts
 - 6.3 Decoding
 - 6.4 Required precision
- 7 Effects of JPEG compression
 - 7.1 Sample photographs
- 8 Lossless further compression
- 9 Derived formats for stereoscopic 3D
 - 9.1 JPEG Stereoscopic
 - 9.2 JPEG Multi-Picture Format
- 10 Patent issues
- 11 Standards
- 12 Implementations
- 13 See also
- 14 References
- 15 External links

Joint Photographic Experts Group



A photo of a cat with the compression rate decreasing, and hence quality increasing, from left to right.

Filename extension	.jpg, .jpeg, .jpe .jif, .jfif, .jfi
Internet media type	image/jpeg
Type code	JPEG
Uniform Type Identifier (UTI)	public.jpeg
Magic number	ff d8 ff
Developed by	Joint Photographic Experts Group
Type of format	lossy image format
Standard	ISO/IEC 10918, ITU-T T.81, ITU-T T.83, ITU-T T.84, ITU-T T.86
Website	www.jpeg.org/jpeg/ (http://www.jpeg.org/jpeg/)



Continuously varied JPEG compression (between Q=100 and Q=1) for an abdominal CT scan.

The JPEG standard

"JPEG" stands for Joint Photographic Experts Group, the name of the committee that created the JPEG standard and also other still picture coding standards. The "Joint" stood for ISO TC97 WG8 and CCITT SGVIII. In 1987 ISO TC 97 became ISO/IEC JTC1 and in 1992 CCITT became ITU-T. Currently on the JTC1 side JPEG is one of two sub-groups of ISO/IEC Joint Technical Committee 1, Subcommittee 29, Working Group 1 (ISO/IEC JTC 1/SC 29/WG 1) – titled as *Coding of still pictures*.^{[4][5][6]} On the ITU-T side ITU-T SG16 is the respective body. The original JPEG group was organized in 1986,^[7] issuing the first JPEG standard in 1992, which was approved in September 1992 as **ITU-T Recommendation T.81**^[8] and in 1994 as **ISO/IEC 10918-1**.

The JPEG standard specifies the codec, which defines how an image is compressed into a stream of bytes and decompressed back into an image, but not the file format used to contain that stream.^[9] The Exif and JFIF standards define the commonly used file formats for interchange of JPEG-compressed images.

JPEG standards are formally named as *Information technology – Digital compression and coding of continuous-tone still images*. ISO/IEC 10918 consists of the following parts:

E-Watch, Inc.
Exh. 2014
Petitioner - HTC Corporation, et al.

Digital compression and coding of continuous-tone still images – Parts^{[5][7][10]}

Part	ISO/IEC standard	ITU-T Rec.	First public release date	Latest amendment	Title	Description
Part 1	ISO/IEC 10918-1:1994 (http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=18902)	T.81 (09/92) (http://www.itu.int/rec/T-REC-T.81)	Sep 18, 1992		Requirements and guidelines	
Part 2	ISO/IEC 10918-2:1995 (http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=20689)	T.83 (11/94) (http://www.itu.int/rec/T-REC-T.83)	Nov 11, 1994		Compliance testing	rules and checks for software conformance (to Part 1)
Part 3	ISO/IEC 10918-3:1997 (http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=25037)	T.84 (07/96) (http://www.itu.int/rec/T-REC-T.84)	Jul 3, 1996	Apr 1, 1999	Extensions	set of extensions to improve the Part 1, including the SPIFF file format
Part 4	ISO/IEC 10918-4:1999 (http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=25431)	T.86 (06/98) (http://www.itu.int/rec/T-REC-T.86)	Jun 18, 1998	Jun 29, 2012	Registration of JPEG profiles, SPIFF profiles, SPIFF tags, SPIFF colour spaces, APPn markers, SPIFF compression types and Registration Authorities (REGAUT)	methods for registering some of the parameters used to extend JPEG
Part 5	ISO/IEC 10918-5:2013 (http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=54989)	T.871 (05/11) (http://www.itu.int/rec/T-REC-T.871)	May 14, 2011		JPEG File Interchange Format (JFIF)	A popular format which has been the de facto file format for images encoded by the JPEG standard. In 2009, the JPEG Committee formally established an Ad Hoc Group to standardize JFIF as JPEG Part 5. ^[11]
Part 6	ISO/IEC 10918-6:2013 (http://www.iso.org/home/store/catalogue_tc/catalogue_detail.htm?csnumber=59634)	T.872 (06/12) (http://www.itu.int/rec/T-REC-T.872)	Jun 2012		Application to printing systems	Specifies a subset of features and application tools for the interchange of images encoded according to the ISO/IEC 10918-1 for printing.

Ecma International TR/98 specifies the JPEG File Interchange Format (JFIF); the first edition was published in June 2009.^[12]

Typical usage

The JPEG compression algorithm is at its best on photographs and paintings of realistic scenes with smooth variations of tone and color. For web usage, where the amount of data used for an image is important, JPEG is very popular. JPEG/Exif is also the most common format saved by digital cameras.

On the other hand, JPEG may not be as well suited for line drawings and other textual or iconic graphics, where the sharp contrasts between adjacent pixels can cause noticeable artifacts. Such images may be better saved in a lossless graphics format such as TIFF, GIF, PNG, or a raw image format. The JPEG standard actually includes a lossless coding mode, but that mode is not supported in most products.

As the typical use of JPEG is a lossy compression method, which somewhat reduces the image fidelity, it should not be used in scenarios where the exact reproduction of the data is required (such as some scientific and medical imaging applications and certain technical image processing work).

JPEG is also not well suited to files that will undergo multiple edits, as some image quality will usually be lost each time the image is decompressed and recompressed, particularly if the image is cropped or shifted, or if encoding parameters are changed – see digital generation loss for details. To avoid this, an image that is being modified or may be modified in the future can be saved in a lossless format, with a copy exported as JPEG for distribution.

JPEG compression

JPEG uses a lossy form of compression based on the discrete cosine transform (DCT). This mathematical operation converts each frame/field of the video source from the spatial (2D) domain into the frequency domain (a.k.a. transform domain.) A perceptual model based loosely on the human psychovisual system discards high-frequency information, i.e. sharp transitions in intensity, and color hue. In the transform domain, the process of reducing information is called quantization. In simpler terms, quantization is a method for optimally reducing a large number scale (with different occurrences of each number) into a smaller one, and the transform-domain is a convenient representation of the image because the high-frequency coefficients, which contribute less to the overall picture than other coefficients, are characteristically small-values with high compressibility. The quantized coefficients are then sequenced and losslessly packed into the output bitstream. Nearly all software implementations of JPEG permit user control over the compression-ratio (as well as other optional parameters), allowing the user to trade off picture-quality for smaller file size. In embedded applications (such as miniDV, which uses a similar DCT-compression scheme), the parameters are pre-selected and fixed for the application.

The compression method is usually lossy, meaning that some original image information is lost and cannot be restored, possibly affecting image quality. There is an optional lossless mode defined in the JPEG standard. However, this mode is not widely supported in products. Widespread use of the format has stimulated the adoption of simulated HDR modes in inexpensive cameras and smartphones, to correct the loss of shadow and highlight detail.

There is also an interlaced "Progressive JPEG" format, in which data is compressed in multiple passes of progressively higher detail. This is ideal for large images that will be displayed while downloading over a slow connection, allowing a reasonable preview after receiving only a portion of the data. However, support for progressive JPEGs is not universal. When progressive JPEGs are received by programs that do not support them (such as versions of Internet Explorer before Windows 7)^[13] the software displays the image only after it has been completely downloaded.

There are also many medical imaging and traffic systems that create and process 12-bit JPEG images, normally grayscale images. The 12-bit JPEG format has been part of the JPEG specification for some time, but this format is not as widely supported.

Lossless editing

A number of alterations to a JPEG image can be performed losslessly (that is, without recompression and the associated quality loss) as long as the image size is a multiple of 1 MCU block (Minimum Coded Unit) (usually 16 pixels in both directions, for 4:2:0 chroma subsampling). Utilities that implement this include `jpegtran`, with user interface `Jpegcrop`, and the `JPG_TRANSFORM` plugin to IrfanView.

Blocks can be rotated in 90 degree increments, flipped in the horizontal, vertical and diagonal axes and moved about in the image. Not all blocks from the original image need to be used in the modified one.

The top and left edge of a JPEG image must lie on an 8×8 pixel block boundary, but the bottom and right edge need not do so. This limits the possible **lossless crop** operations, and also prevents flips and rotations of an image whose bottom or right edge does not lie on a block boundary for all channels (because the edge would end up on top or left, where – as aforementioned – a block boundary is obligatory).

When using lossless cropping, if the bottom or right side of the crop region is not on a block boundary then the rest of the data from the partially used blocks will still be present in the cropped file and can be recovered. It is also possible to transform between baseline and progressive formats without any loss of quality, since the only difference is the order in which the coefficients are placed in the file.

Furthermore, several JPEG images can be losslessly joined together, as long as they were saved with the same quality and the edges coincide with block boundaries.

JPEG files

The file format known as "JPEG Interchange Format" (JIF) is specified in Annex B of the standard. However, this "pure" file format is rarely used, primarily because of the difficulty of programming encoders and decoders that fully implement all aspects of the standard and because of certain shortcomings of the standard:

- Color space definition
- Component sub-sampling registration
- Pixel aspect ratio definition.

Several additional standards have evolved to address these issues. The first of these, released in 1992, was JPEG File Interchange Format (or JFIF), followed in recent years by Exchangeable image file format (Exif) and ICC color profiles. Both of these formats use the actual JIF byte layout, consisting of different *markers*, but in addition employ one of the JIF standard's extension points, namely the *application markers*: JFIF use APP0, while Exif use APP1. Within these segments of the file, that were left for future use in the JIF standard and aren't read by it, these standards add specific metadata.

Thus, in some ways JFIF is a cutdown version of the JIF standard in that it specifies certain constraints (such as not allowing all the different encoding modes), while in other ways it is an extension of JIF due to the added metadata. The documentation for the original JFIF standard states:^[14]

JPEG File Interchange Format is a minimal file format which enables JPEG bitstreams to be exchanged between a wide variety of platforms and applications. This minimal format does not include any of the advanced features found in the TIFF JPEG specification or any application specific file format. Nor should it, for the only purpose of this simplified format is to allow the exchange of JPEG compressed images.

Image files that employ JPEG compression are commonly called "JPEG files", and are stored in variants of the JIF image format. Most image capture devices (such as digital cameras) that output JPEG are actually creating files in the Exif format, the format that the camera industry has standardized on for metadata interchange. On the other hand, since the Exif standard does not allow color profiles, most image editing software stores JPEG in JFIF format, and also include the APP1 segment from the Exif file to include the metadata in an almost-compliant way; the JFIF standard is interpreted somewhat flexibly.^[15]

Strictly speaking, the JFIF and Exif standards are incompatible because they each specify that their marker segment (APP0 or APP1, respectively) appears first. In practice, most JPEG files contain a JFIF marker segment that precedes the Exif header. This allows older readers to correctly handle the older format JFIF segment, while newer readers also decode the following Exif segment, being less strict about requiring it to appear first.

JPEG filename extensions

The most common filename extensions for files employing JPEG compression are **.jpg** and **.jpeg**, though **.jpe**, **.jif** and **.jif** are also used. It is also possible for JPEG data to be embedded in other file types – TIFF encoded files often embed a JPEG image as a thumbnail of the main image; and MP3 files can contain a JPEG of cover art, in the ID3v2 tag.

Color profile

Many JPEG files embed an ICC color profile (color space). Commonly used color profiles include sRGB and Adobe RGB. Because these color spaces use a non-linear transformation, the dynamic range of an 8-bit JPEG file is about 11 stops; see gamma curve.

Syntax and structure

A JPEG image consists of a sequence of *segments*, each beginning with a *marker*, each of which begins with a 0xFF byte followed by a byte indicating what kind of marker it is. Some markers consist of just those two bytes; others are followed by two bytes indicating the length of marker-specific payload data that follows. (The length includes the two bytes for the length, but not the two bytes for the marker.) Some markers are followed by entropy-coded data; the length of such a marker does not include the entropy-coded data. Note that consecutive 0xFF bytes are used as fill bytes for padding purposes, although this fill byte padding should only ever take place for markers immediately following entropy-coded scan data (see JPEG specification section B.1.1.2 and E.1.2 for details; specifically "In all cases where markers are appended after the compressed data, optional 0xFF fill bytes may precede the marker").

Within the entropy-coded data, after any 0xFF byte, a 0x00 byte is inserted by the encoder before the next byte, so that there does not appear to be a marker where none is intended, preventing framing errors. Decoders must skip this 0x00 byte. This technique, called *byte stuffing* (see JPEG specification section F.1.2.3), is only applied to the entropy-coded data, not to marker payload data. Note however that entropy-coded data has a few markers of its own; specifically the Reset markers (0xD0 through 0xD7), which are used to isolate independent chunks of entropy-coded data to allow parallel decoding, and encoders are free to insert these Reset markers at regular intervals (although not all encoders do this).

Common JPEG markers^[16]

Short name	Bytes	Payload	Name	Comments
SOI	0xFF, 0xD8	<i>none</i>	Start Of Image	
SOF0	0xFF, 0xC0	<i>variable size</i>	Start Of Frame (Baseline DCT)	Indicates that this is a baseline DCT-based JPEG, and specifies the width, height, number of components, and component subsampling (e.g., 4:2:0).
SOF2	0xFF, 0xC2	<i>variable size</i>	Start Of Frame (Progressive DCT)	Indicates that this is a progressive DCT-based JPEG, and specifies the width, height, number of components, and component subsampling (e.g., 4:2:0).
DHT	0xFF, 0xC4	<i>variable size</i>	Define Huffman Table(s)	Specifies one or more Huffman tables.
DQT	0xFF, 0xDB	<i>variable size</i>	Define Quantization Table(s)	Specifies one or more quantization tables.
DRI	0xFF, 0xDD	4 bytes	Define Restart Interval	Specifies the interval between RST n markers, in macroblocks. This marker is followed by two bytes indicating the fixed size so it can be treated like any other variable size segment.
SOS	0xFF, 0xDA	<i>variable size</i>	Start Of Scan	Begins a top-to-bottom scan of the image. In baseline DCT JPEG images, there is generally a single scan. Progressive DCT JPEG images usually contain multiple scans. This marker specifies which slice of data it will contain, and is immediately followed by entropy-coded data.
RSTn	0xFF, 0xD n ($n=0..7$)	<i>none</i>	Restart	Inserted every r macroblocks, where r is the restart interval set by a DRI marker. Not used if there was no DRI marker. The low 3 bits of the marker code cycle in value from 0 to 7.
APPn	0xFF, 0xE n	<i>variable size</i>	Application-specific	For example, an Exif JPEG file uses an APP1 marker to store metadata, laid out in a structure based closely on TIFF.
COM	0xFF, 0xFE	<i>variable size</i>	Comment	Contains a text comment.
EOI	0xFF, 0xD9	<i>none</i>	End Of Image	

There are other *Start Of Frame* markers that introduce other kinds of JPEG encodings.

Since several vendors might use the same APP n marker type, application-specific markers often begin with a standard or vendor name (e.g., "Exif" or "Adobe") or some other identifying string.

At a restart marker, block-to-block predictor variables are reset, and the bitstream is synchronized to a byte boundary. Restart markers provide means for recovery after bitstream error, such as transmission over an unreliable network or file corruption. Since the runs of macroblocks between restart markers may be independently decoded, these runs may be decoded in parallel.

JPEG codec example

Although a JPEG file can be encoded in various ways, most commonly it is done with JFIF encoding. The encoding process consists of several steps:

1. The representation of the colors in the image is converted from RGB to Y'C_BC_R, consisting of one luma component (Y'), representing brightness, and two chroma components, (C_B and C_R), representing color. This step is sometimes skipped.
2. The resolution of the chroma data is reduced, usually by a factor of 2 or 3. This reflects the fact that the eye is less sensitive to fine color details than to fine brightness details.
3. The image is split into blocks of 8×8 pixels, and for each block, each of the Y, C_B, and C_R data undergoes the Discrete Cosine Transform (DCT), which was developed in 1974 by N. Ahmed, T. Natarajan and K. R. Rao; see Citation 1 in Discrete cosine transform. A DCT is similar to a Fourier transform in the sense that it produces a kind of spatial frequency spectrum.
4. The amplitudes of the frequency components are quantized. Human vision is much more sensitive to small variations in color or brightness over large areas than to the strength of high-frequency brightness variations. Therefore, the magnitudes of the high-frequency components are stored with a lower accuracy than the low-frequency components. The quality setting of the encoder (for example 50 or 95 on a scale of 0–100 in the Independent JPEG Group's library^[17]) affects to what extent the resolution of each frequency component is reduced. If an excessively low quality setting is used, the high-frequency components are discarded altogether.
5. The resulting data for all 8×8 blocks is further compressed with a lossless algorithm, a variant of Huffman encoding.

The decoding process reverses these steps, except the *quantization* because it is irreversible. In the remainder of this section, the encoding and decoding processes are described in more detail.

Encoding

Many of the options in the JPEG standard are not commonly used, and as mentioned above, most image software uses the simpler JFIF format when creating a JPEG file, which among other things specifies the encoding method. Here is a brief description of one of the more common methods of encoding when applied to an input that has 24 bits per pixel (eight each of red, green, and blue). This particular option is a lossy data compression method.

Color space transformation

First, the image should be converted from RGB into a different color space called $Y'CbCr$ (or, informally, YCbCr). It has three components Y' , C_B and C_R : the Y' component represents the brightness of a pixel, and the C_B and C_R components represent the chrominance (split into blue and red components). This is basically the same color space as used by digital color television as well as digital video including video DVDs, and is similar to the way color is represented in analog PAL video and MAC (but not by analog NTSC, which uses the YIQ color space). The $Y'CbCr$ color space conversion allows greater compression without a significant effect on perceptual image quality (or greater perceptual image quality for the same compression). The compression is more efficient because the brightness information, which is more important to the eventual perceptual quality of the image, is confined to a single channel. This more closely corresponds to the perception of color in the human visual system. The color transformation also improves compression by statistical decorrelation.

A particular conversion to $Y'CbCr$ is specified in the JFIF standard, and should be performed for the resulting JPEG file to have maximum compatibility. However, some JPEG implementations in "highest quality" mode do not apply this step and instead keep the color information in the RGB color model, where the image is stored in separate channels for red, green and blue brightness components. This results in less efficient compression, and would not likely be used when file size is especially important.

Downsampling

Due to the densities of color- and brightness-sensitive receptors in the human eye, humans can see considerably more fine detail in the brightness of an image (the Y' component) than in the hue and color saturation of an image (the C_b and C_r components). Using this knowledge, encoders can be designed to compress images more efficiently.

The transformation into the $Y'CbCr$ color model enables the next usual step, which is to reduce the spatial resolution of the C_b and C_r components (called "downsampling" or "chroma subsampling"). The ratios at which the downsampling is ordinarily done for JPEG images are 4:4:4 (no downsampling), 4:2:2 (reduction by a factor of 2 in the horizontal direction), or (most commonly) 4:2:0 (reduction by a factor of 2 in both the horizontal and vertical directions). For the rest of the compression process, Y' , C_b and C_r are processed separately and in a very similar manner.

Block splitting

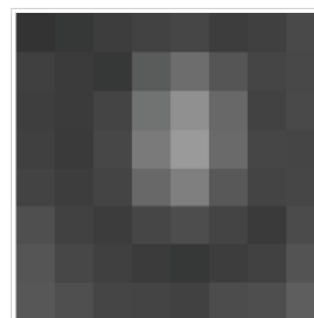
After subsampling, each channel must be split into 8×8 blocks. Depending on chroma subsampling, this yields (Minimum Coded Unit) MCU blocks of size 8×8 (4:4:4 – no subsampling), 16×8 (4:2:2), or most commonly 16×16 (4:2:0). In video compression MCUs are called macroblocks.

If the data for a channel does not represent an integer number of blocks then the encoder must fill the remaining area of the incomplete blocks with some form of dummy data. Filling the edges with a fixed color (for example, black) can create ringing artifacts along the visible part of the border; repeating the edge pixels is a common technique that reduces (but does not necessarily completely eliminate) such artifacts, and more sophisticated border filling techniques can also be applied.

Discrete cosine transform

Next, each 8×8 block of each component (Y , C_b , C_r) is converted to a frequency-domain representation, using a normalized, two-dimensional type-II discrete cosine transform (DCT), which was introduced by N. Ahmed, T. Natarajan and K. R. Rao in 1974; see Citation 1 in Discrete cosine transform. The DCT is sometimes referred to as "type-II DCT" in the context of a family of transforms as in discrete cosine transform, and the corresponding inverse (IDCT) is denoted as "type-III DCT".

As an example, one such 8×8 8-bit subimage might be:

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$


The 8×8 sub-image shown in 8-bit grayscale

Before computing the DCT of the 8×8 block, its values are shifted from a positive range to one centered around zero. For an 8-bit image, each entry in the original block falls in the range $[0, 255]$. The midpoint of the range (in this case, the value 128) is subtracted from each entry to produce a data range that is centered around zero, so that the modified range is $[-128, 127]$. This step reduces the dynamic range requirements in the DCT processing stage that follows. (Aside from the difference in dynamic range within the DCT stage, this step is mathematically equivalent to subtracting 1024 from the DC coefficient after performing the transform – which may be a better way to perform the operation on some architectures since it involves performing only one subtraction rather than 64 of them.)

This step results in the following values:

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.