Introduction

IPFS Plugin

`IPFS` Plugin is built for *Unreal Engine* to allow game developers make use of the *IPFS* technology. This plugin offers the ability to interact with *IPFS* nodes via HTTP. It offers downloading and uploading files directly in engine.

What Is IPFS

IPFS the InterPlanetary File System is a hypermedia distribution protocol addressed by content and identities. It enables the creation of completely distributed applications, and in doing so aims to make the web faster, safer, and more open.

IPFS is a protocol:

- Defines a content-addressed file system.
- Coordinates content delivery.
- Combines Kademlia + BitTorrent + Git.

IPFS is a file system:

- Has directories and files.
- Is a mountable filesystem (via FUSE).

IPFS is a web:

- Can be used to view documents like the conventional web.
- Files are accessible via HTTP at `https://ipfs.io/<path>`.
- Browsers and extensions can learn to use the `ipfs://` URL or `ipns://` URI schemes directly.
- Hash-addressed content guarantees authenticity.

IPFS is modular:

- Connection layer over any network protocol.

- Routing layer.
- Uses a routing layer DHT (Kademlia/Coral).
- Uses a path-based naming service.
- Uses a BitTorrent-inspired block exchange.

IPFS uses crypto:

- Cryptographic-hash content addressing.
- Block-level deduplication.
- File integrity plus versioning.
- File-system-level encryption plus signing support.

IPFS is p2p:

- Worldwide peer-to-peer file transfers.
- Completely decentralized architecture.
- No central point of failure.

IPFS is a CDN:

- Add a file to the file system locally, and it's now available to the world.
- Caching-friendly (content-hash naming).
- BitTorrent-based bandwidth distribution.

IPFS has a name service:

- IPNS, an SFS-inspired name system.
- Global namespace based on PKI.
- It serves to build trust chains.
- It's compatible with other NSes.
- Can map DNS, .onion, .bit, etc to IPNS.

How IPFS works

Here's what happens when you add a file to IPFS — whether you're storing that file on your own local node or one operated by a pinning service or IPFS-enabled app.

- When you add a file to IPFS, your file is split into smaller chunks, cryptographically hashed, and given a unique fingerprint called a content identifier (CID). This CID acts as a permanent record of your file as it exists at that point in time.
- When other nodes look up your file, they ask their peer nodes who's storing the content referenced by the file's CID. When they view or download your file, they cache a copy — and become another provider of your content until their cache is cleared.
- A node can pin content in order to keep (and provide) it forever, or discard content it hasn't used in a while to save space. This means each node in the network stores only content it is interested in, plus some indexing information that helps figure out which node is storing what.
- If you add a new version of your file to IPFS, its cryptographic hash is different, and so it gets a new CID. This means files stored on IPFS are resistant to tampering and censorship — any changes to a file don't overwrite the original, and common chunks across files can be reused in order to minimize storage costs.
- However, this doesn't mean you need to remember a long string of CIDs — IPFS can find the latest version of your file using the IPNS decentralized naming system, and DNSLink can be used to map CIDs to human-readable DNS names.

✎ Edit this page