



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# GVP Deployment Guide

Fetching Module and Squid

5/11/2025

# Fetching Module and Squid

## Fetching Module Now Integrated

The Fetching Module (the executable file `pwproxy.exe` in earlier versions of GVP) is integrated with the Media and Call Control Platform Installation Packages[IP]). The 8.1.2 and later releases of the Fetching Module support HTTP/1.1-compliant caching and is responsible for fetching VoiceXML and CCXML files, as well as HTTP/HTTPS resources.

The Fetching Module efficiently passes fetch results and other information back to the NGI (on the Media Control Platform host) or CCXMLI (on the Call Control Platform host).

## Squid Caching Proxy

The third-party Squid software acts as a caching proxy for the Fetching Module. Like any other caching mechanism, the Squid Caching Proxy caches frequently used files so that fetching a copy of a file needs to occur only once; after that, the file is retrieved from the cache.

In GVP 8.1.2 and above, the Squid proxy is optional to provide more flexibility in the deployment. For example, when it is deployed with a web server, multiple Media Control Platform instances can share the same Squid proxy to optimize caching.

For information about how to install Squid Caching Proxy and the Fetching Module, see [Specifications and Standards](#).

## GVP Caching

Audio and video recordings are common in VoiceXML documents, and they can be very large. Because their content is also mostly static, using cached content significantly improves performance.

GVP can perform the caching function itself (through the Fetching Module and Squid), or you can add another server a caching appliance, or a web proxy server.

## External Caching

External cache servers can be beneficial. For example, if you have a site with 10 GVP servers, and an audio file expires, each server must fetch a new copy of the audio file. If there is an external cache server, fetching a new copy of the audio file occurs only once. Also, the external cache servers typically have very robust cache management tools to purge and refresh content.

## Fetching Module Caching

The Fetching Module performs caching, as follows:

1. The 8.1.2 and above Fetching Module itself performs in-memory caching, which is HTTP/1.1-compliant.

(GVP 8.1.1 and earlier versions of the Fetching Module are not HTTP/1.1-compliant.)

2. If the Fetching Module determines that it cannot serve the request from its in-memory cache, it goes to the Squid Caching Proxy to try to fetch the content. The Squid Caching Proxy performs HTTP/1.1-compliant caching.
3. If Squid determines that it cannot serve the content from its cache, it goes to the web server to try to fetch the content.

### Tip

It is important that the clocking between the HTTP server and client be synchronized, so that the caching policies such as max-age and max-stale work properly.

The Media and Call Control Platforms support clearing of the Fetching Module in-memory cache at runtime. In Genesys Administrator a custom command (CLEARFMCACHE) can be triggered and sent to the Media or Call Control Platform through Management Framework's CCLib.

For more information about how the Fetching Module performs caching, see [Caching](#).

## Page Collector Caching

Page Collector is the caching mechanism within GVPI. It fetches XML pages, determines the document format, and passes the pages on to the appropriate parser (VoiceXML or Transportation Extensible Markup Language (TXML)). The cache consists of two types of files an index file and the actual cached files.

### Index File

The index file stores thumbnail data for each cached file in binary format, which facilitates a fast search. The thumbnail data consists of the URL, the local file location, response headers, and the last access time. At process start up, the index file is read into memory. After that, the in-memory image is persisted to the index file at periodic intervals and during shutdown.

At process start up, the index file is read using best effort. If a file is missing or corrupt, the process starts up with whatever data it can retrieve. During an HTTP operation, the index file is searched for the entry. If the entry is not found, the request is sent to the web server. If the entry is found, it is validated (using the RFC 2616 standard) to determine whether the cached entry should be returned to the client or whether a request should be sent to the web server.

### Caching Files

Each response that can be cached is stored in a local file. The file location is <MCP Install\_Dir>/data/CnCache/<server>/CACnnnn.tmp, where <server> is the name of the web server (as it appears in the URL) from which the response was received, and *nnnn* is a random number. Each time a response is cached, a corresponding thumbnail entry is created in the index file.

For information about how to use HTTP caching to improve performance, see [Optimizing GVP Performance Through HTTP Caching](#).