

WebKit

What does Lockdown Mode/Captive Portal Mode disable?

- WebAssembly
- MP3 Playback
- MathML
- Web Audio API
- Gamepad API
- JPEG 2000
- WebGL
- Speech Recognition API
- RTCDataChannel
- MediaDevices.getUserMedia()
- PDF Viewer
- SVG Fonts
- JIT
- ServiceWorkers

This can be seen in the function [adjustSettingsForLockdownMode](#) in the file [Source/WebKit/WebProcess/WebPage/WebPage.cpp](#)

```
static void adjustSettingsForLockdownMode(Settings& settings, const WebPreferencesStore& store)
{
    // Disable unstable Experimental settings, even if the user enabled them for local use.
    settings.disableUnstableFeaturesForModernWebKit();
    Settings::disableGlobalUnstableFeaturesForModernWebKit();

    settings.setWebGLEnabled(false);
#ifdef ENABLE(WEBGL2)
    settings.setWebGL2Enabled(false);
#endif
#ifdef ENABLE(GAMEPAD)
    settings.setGamepadsEnabled(false);
#endif
#ifdef ENABLE(WIRELESS_PLAYBACK_TARGET)
    settings.setRemotePlaybackEnabled(false);
#endif
    settings.setFileSystemAccessEnabled(false);
    settings.setAllowsPictureInPictureMediaPlayback(false);
#ifdef ENABLE(PICTURE_IN_PICTURE_API)
    settings.setPictureInPictureAPIEnabled(false);
#endif
    settings.setSpeechRecognitionEnabled(false);
#ifdef ENABLE(SPEECH_SYNTHESIS)
    settings.setSpeechSynthesisAPIEnabled(false);
#endif
#ifdef ENABLE(NOTIFICATIONS)
    settings.setNotificationsEnabled(false);
#endif
#ifdef ENABLE(SERVICE_WORKER)
    settings.setPushAPIEnabled(false);
#endif
#ifdef ENABLE(WEBXR)
    settings.setWebXREnabled(false);
    settings.setWebXRAugmentedRealityModuleEnabled(false);
#endif
#ifdef ENABLE(MODEL_ELEMENT)
    settings.setModelElementEnabled(false);
#endif
#ifdef ENABLE(MEDIA_STREAM)
    settings.setMediaDevicesEnabled(false);
#endif
#ifdef ENABLE(WEB_AUDIO)
    settings.setWebAudioEnabled(false);
#endif
    settings.setDownloadableBinaryFontAllowedTypes(DownloadableBinaryFontAllowedTypes::Restricted);
#ifdef ENABLE(WEB_CODECS)
    settings.setWebCodecsEnabled(false);
    settings.setWebCodecsAV1Enabled(false);
#endif
#ifdef ENABLE(WEB_RTC)
    settings.setPeerConnectionEnabled(false);
    settings.setWebRTCEncodedTransformEnabled(false);
#endif
#ifdef ENABLE(MATHML)
    settings.setMathMLEnabled(false);
#endif
#ifdef ENABLE(PDFJS)
    settings.setPdfJSViewerEnabled(true);
#endif
#ifdef USE(SYSTEM_PREVIEW)
    settings.setSystemPreviewEnabled(false);
#endif
    settings.setEmbedElementEnabled(false);
```


WebKit

JIT?

I knew [JIT](#) was disabled, but had a hell of a time determining where in the source code that was being enforced until I back tracked ALL the way up to the top to [ProcessLauncher::launchProcess\(\)](#) in [ProcessLauncherCocoa.mm](#)

There we can see:

- [launchProcess\(\)](#)
 - [serviceName\(m_launchOptions, m_client\);](#)
 - [webContentServiceName\(launchOptions.nonValidInjectedCodeAllowed, client\);](#)

```
static const char* webContentServiceName(bool nonValidInjectedCodeAllowed, ProcessLauncher::Client* client)
{
    if (client && client->shouldEnableLockdownMode())
        return "com.apple.WebKit.WebContent.CaptivePortal";
    return nonValidInjectedCodeAllowed ? "com.apple.WebKit.WebContent.Development" : "com.apple.WebKit.WebContent";
}
```

If [Lockdown Mode](#) is enabled then the WebKit launcher will start a [com.apple.WebKit.WebContent.CaptivePortal](#) process instead of the normal [com.apple.WebKit.WebContent](#) process.

You can also see what entitlements these different services get assigned in the file [Source/WebKit/Scripts/process-entitlements.sh](#)