HARMAN
HARMAN Connected Services Corporation
636 Ellis Street
Mountain View, CA 94043, USA
+1 650 623 9400
+1 650 623 9401

January 2021

HARMAN - "Packaged Browser" solution for extended Flash Player support

Architecture and Security

Introduction

As one of HARMAN's offerings for extended Flash Player support, the "packaged browser" solution wraps up a browser engine together with a captive, customized version of the Adobe Flash Player, resulting in a new application that can be separately installed and distributed by our licensees.
This document describes the architecture of this solution and covers a little of the security aspects, both of the solution approach as a whole and more specifically around the development and architecture of the provided executable files.

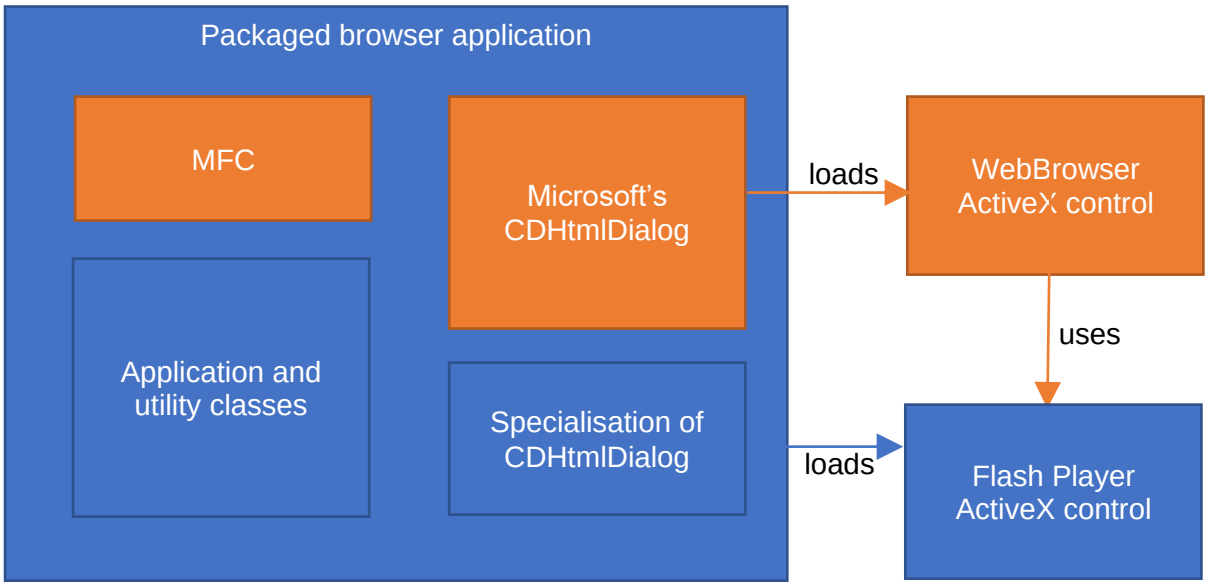
Architecture

The packaged browser solution comes in two variants:

1) Windows/ActiveX: this uses a custom-built Windows executable that loads in the 'WebBrowser' ActiveX control that is provided by Microsoft and that contains the same browser engine as used by IE, and by the new Edge browser's "IE Mode". This engine then creates an instance of the Flash Player also using ActiveX, from the HCSFP.ocx file.
2) Electron/Chromium: this is a cross-platform option that uses the open source "Electron.js" package to create the application for Windows, MacOS or Linux. This contains the Chromium browser engine and will load the Flash Player via the "Pepper" or "PPAPI" mechanism, using the appropriate Flash Player library provided by HARMAN.

Note that HARMAN also offer the Flash Player binary files to be used both within third party applications (including customers' equivalent 'packaged browser' applications, or C++/C# apps that load in the Flash Player ActiveX control, etc) and as standard browser plug-ins – particularly for Internet Explorer. Much of the below architecture information doesn't apply when the Flash Player binaries are considered separately from the "packaged browser" solution, but the security section does contain some details that would still apply.

ActiveX Version

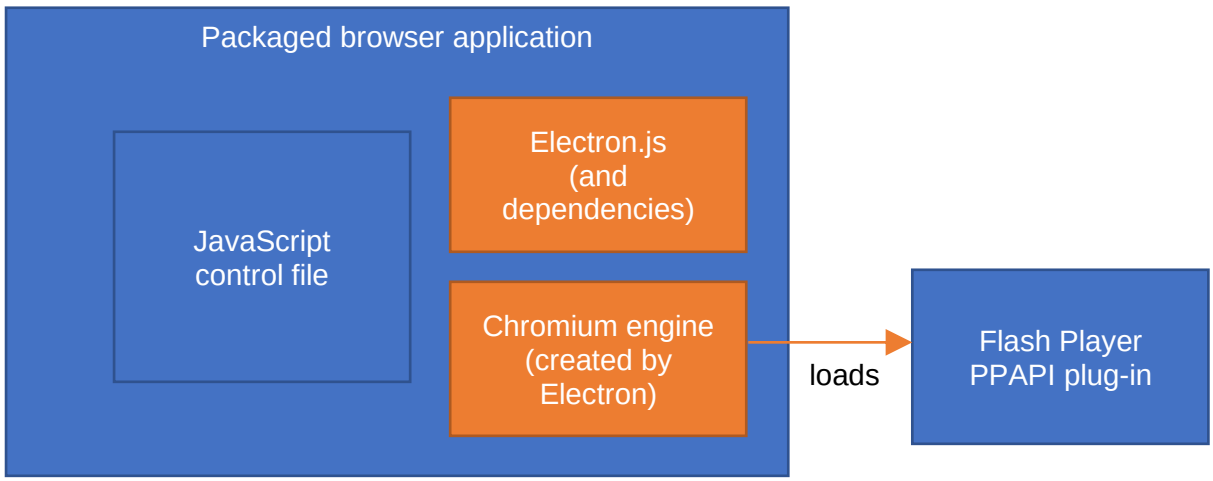


The packaged browser application uses Microsoft's 'MFC' library for the application and window framework, with the main user interface being provided via a specialisation of the CDHtmlDialog class. This class loads in the 'WebBrowser' ActiveX control provided by Microsoft within their MSHTML.dll library, and this uses the customised Flash Player that HARMAN create.

The functionality provided within the packaged browser application/utility classes is as follows:

1) Load the customized Flash Player and perform local COM registration for this such that when the Microsoft WebBrowser control creates an instance of the Flash plug-in object, this library is used rather than any system-registered version that may be present.
2) Optionally checks for command-line parameters or configuration file settings, as requested by customers (for example to pass in a URL to be loaded at start-up).
3) Optionally provides a console output log (only enabled during evaluation builds) to capture information/error events from the JavaScript console into a file
4) Hooks into the Windows notifications and WebBrowser events to provide features such as:
   a) 'New window' handling – when a new window is requested by the WebBrowser, this is handled via an internal specialisation of CDHtmlDialog, rather than being passed out to the operating system to be handled by the user's default standard browser.
   b) 'Download' clean-up – ensuring that new windows that are created purely for downloading a file are then closed down once the 'download' dialog is displayed.
   c) 'Client site' operations – filtering out unnecessary JavaScript error dialogs and handling requests from the browser and the plug-in for information about the host.
   d) Optional functionality for some customers to further adjust the host behaviour, for example: blocking some URLs from opening; or adding information from the application and browser contents to the title bar or status bar.

Electron version



From HARMAN's perspective, the Electron version of the solution is more straightforward. Electron.js is used to handle all user interface elements and to load in the Chromium engine, with a control script being provided by HARMAN in order to set up the initial window and to provide the following functionality:

1) Load the customized Flash Player by providing the appropriate path into the parameters used by Electron to initialise the Chromium engine
2) Optionally checks for command-line parameters or configuration file settings, as requested by customers (for example to pass in a URL to be loaded at start-up).
3) Hooks into the Electron events to provide features such as:
   a) 'Download' clean-up – ensuring that new windows that are created purely for downloading a file are then closed down once the 'download' dialog is displayed.
   b) Optional functionality for some customers to further adjust the host behaviour, for example: blocking some URLs from opening; or ensuring confirmation dialogs are presented prior to closing windows.

Flash Player

HARMAN use Adobe's Flash Player source code to create the plug-in libraries, with a number of changes being applied to this as required by the agreements between Adobe and HARMAN:

1) HARMAN do not have a license to incorporate the software encoders/decoders for H.264 (AVC) or AAC formats. Any audio/video streams that include these formats will fail to play.
2) The Flash Access DRM and Encrypted RTMP functionality relies on libraries that HARMAN do not have the source code to and hence these capabilities are not supported.
3) Adobe require that HARMAN's deployments of the Flash Player include a limit on the SWF content that can be displayed – they cannot be used for open web browsing. As such, HARMAN has integrated a mandatory list of permitted URLs using a variety of pattern matching techniques. This 'allow-list' is incorporated into the customer-specific Flash Player (note that some other mechanisms could also be used to limit the permitted SWF content).

HARMAN will also continue to apply bug fixes and in particular any security patches that are required, during the term of a customer's license agreement. Operating system compatibility updates will also be provided where needed – note that each distribution of the Flash Player for different operating systems/CPU architectures and each different plug-in mechanism would be counted as separate "platforms" according to the license agreements.

Security

The two options – ActiveX using IE's WebBrowser engine, or PPAPI using Electron's Chromium engine – have different considerations regarding the overall security of the solution. Below is a comparison of these two options:

	ActiveX + IE/WebBrowser	**PPAPI + Electron/Chromium**
Browser engine	Uses the WebBrowser control as provided by Microsoft. This may be updated by Microsoft in order to provide security patches.	Uses the Chromium engine as provided by Electron.js. After Chromium remove support for PPAPI, the version of Electron and Chromium used in this solution will need to be frozen.
Flash Player	Uses the ActiveX version with an embedded content filtering list to prevent unauthorised content. Will be updated by HARMAN in order to provide security patches. The ActiveX Flash Player runs in-process within the WebBrowser control with tight coupling between the browser engine and the plug-in.	Uses the PPAPI version with an embedded content filtering list to prevent unauthorised content. Will be updated by HARMAN in order to provide security patches. The PPAPI Flash Player runs in a separate process from the Chromium renderer, and uses a sandbox with the PPAPI mechanism to separate the browser engine from the plug-in.
Host application	Uses a custom-built Windows application using C++. Cannot easily be modified; can only be run by someone with access to the computer.	Uses the open source Electron.js framework with a JavaScript control script. Could be edited by someone with access to the computer and administrator privileges.

If HARMAN are just supplying the Flash Player as a separate binary or as a browser plug-in, then HARMAN will be following the same approach as above i.e. maintaining and supporting the Flash Player to patch any security issues reported in this. This applies equally to the third plug-in mechanism, NPAPI (used by Firefox and Webkit).

There are three aspects to security that need to be considered:

Security of the host application

Both solutions run as local executable files on the end user's computer. In order for any modifications or malicious usage to be made of these executables, an attacker would already need to have access to the user's computer. From this perspective, the solution is therefore as safe as any other locally installed application.

In terms of a user trying to crash the application or expose a vulnerability in it – for example by passing in a special string as a URL that could cause issues within the application logic – there are limited opportunities for this. Generally, the only inputs provided in this way are the URL from the command-line, or configuration data provided in a file. The code has been peer reviewed and subjected to static analysis to minimise the possibility of such actions, but given the requirement of the attacked to already have access to the machine in order to use this application, the benefits of such attacks are unclear.

Security of the browser engine – including JavaScript runtime and Flash Player

This is the main area of concern. HARMAN do not create the browser engines used in this solution, they are provided by Microsoft or by Electron.js, and there are likely to be known vulnerabilities in these. The expectation is that Microsoft would patch the WebBrowser ActiveX control if a critical security flaw was exposed, and would provide this to the end user's computer as part of a Windows Update, but this is out of HARMAN's hands. In terms of the Electron.js version of Chromium, this lags the version used in the Google Chrome browser slightly, and once Chromium stop supporting PPAPI it will then no longer be possible to keep this up to date. Security vulnerabilities could therefore be found in the Chromium engine that cannot be patched within the packaged browser solution.

To help reduce the risks, please see the 'mitigation' section below.

In terms of the Flash Player, HARMAN are expecting that a number of vulnerabilities may be published at the start of 2021 once the official end of life of the Flash Player has been reached. The number of security issues has dropped considerably over the past three years however, and the expectation is that very few new vulnerabilities would be found after this since it is not worth a hacker trying to find a flaw in the Flash Player given it will not be available to home users, or the majority of business users, after 2020.

However, HARMAN will still continue to support the Flash Player and to respond with updates should there be any new vulnerabilities found within the Flash Player. New versions of the runtime will be created and provided to licensees, who can then decide whether this new version should be rolled out to their users.

Potential access of the host application from the browser engine

Another aspect to consider is whether the content loaded within the browser engine or Flash Player would be able to access the host application.

For the Electron.js version of the packaged browser, there are a number of security considerations and recommendations that the Electron team have provided. HARMAN have reviewed these and will ensure that standard deliveries will follow the given recommendations, unless customers request specific exceptions to this.

See https://www.electronjs.org/docs/tutorial/security

For the WebBrowser ActiveX control, the interfaces between the browser engine and the host application are well defined and for the most part HARMAN just uses the default behaviour as contained within the CDHtmlDialog class and its supporting framework. There is a custom client site class that is set up, but again with only a very limited implementation. Where some functionality has been added, this has been peer reviewed to ensure it cannot result in security vulnerabilities.

Mitigation

One of the requirements from Adobe, in order to reduce the security risks inherent in using the Flash Player, is that only a closed, controlled set of SWF content should be loaded. HARMAN manage this by checking the initial SWF file passed to the plug-in (and any SWF subsequently loaded in at level 0 of the Flash Player) against an 'allow list' or similar mechanism. This is based on the implementation that Adobe provide for the "enterprise enablement" feature, although is hard-coded in an obfuscated form into the Flash Player binary itself.

The goal then is to ensure that only content that is authored or approved by the licensee is able to run within the Flash Player. If the user somehow navigated to a web page containing hidden, malicious SWF files, or adverts that may contain malicious payloads, then these would not be displayed by the Flash Player – the requests for such content would be blocked, and the Flash Player would instead display an explanatory icon.

The same approach may be taken to the HTML/JavaScript content that the packaged browser solution loads. Whilst this is not implemented by default, it is an optional feature whereby the host application can filter out the URLs that the end user can navigate to (note that it may not be possible to then filter all of the content that a web page loads – but the navigation events can be filtered to prevent the user from leaving the licensee's content).

With this option, it would therefore be possible to provide an allowed URL list or pattern that the host application uses to block the web engine from navigating to certain URLs. The implementation of such a feature will be customer-specific, and is likely to require some discussions to ensure that the permitted URLs work for all deployments of a web-based application across the cloud and when deployed on customer premises.

Note too that if a customer is not satisfied with the security aspects of the host application, HARMAN would be able to offer the licensing of just the Flash Player with the appropriate plug-in mechanism, and a customer is free to develop their own host application or custom browser in order to display their application using this Flash Player.

Remote access and sockets

The Flash Player has some existing functionality within it that could result in it opening up sockets in order to provide local debugging or telemetry information. A release build would attempt to open a telemetry socket only if a configuration file is found that provides the host name and port to use for a connection to the Adobe Scout tool- such configuration file is written by Scout when it starts up, and cleared when it shuts down.

HARMAN have removed all online status/update checks and "call-home" mechanisms that were found in the Flash Player so no additional calls should be being made by the Flash Player other than as a direct result of the initial request to load a SWF file and any HTTP/socket requests that the SWF file makes, plus its standard security checks for cross-domain policy files etc.

Neither the Packaged Browser, nor the Flash Player to the best of HARMAN's knowledge, open up any ports for incoming socket communication (other than as directed by the loaded SWF content for example as part of video streaming connections).

Time-bombs and kill switches

It has been reported that the Flash Player contains a "kill switch" to prevent it from working in 2021; this is a simplistic view and is intended for general home consumers. For enterprise organisations, there is the option to set up the 'enterprise enablement' feature in advance of 2021, and Adobe provide guidance on this within the Flash Player Administration Guide. The so-called "kill switch" is just the fact that this feature will turn on by default at the start of 2021.

Whilst HARMAN add in an additional protection for evaluation copies of the Flash Player, the above kill switch operation is removed from HARMAN's builds. Commercially licensed versions of the Flash Player from HARMAN may contain an expiry date, if agreed with a customer and as defined in the license agreement, but generally the software has no kill switches or expiry dates, and will continue to run without reference to the current date and without any remote-access or call-home functionality.

Flash Player vulnerabilities

Whilst the number of security vulnerabilities discovered in the Flash Player has dropped massively over the past three years, it is still likely that there are vulnerabilities within the Flash Player and possible that some of these may be uncovered and reported after 2020. From 1st January 2021 Adobe will no longer support consumers with updates to the Flash Player, so if there are any security updates required, these will come from HARMAN for companies who have a valid license agreement in place.

Generally, standard security vulnerabilities have a 90 day disclosure window, meaning that a report is made and the software vendor the has 90 days to provide a fix out to their install base. HARMAN are looking to provide updates as needed on a quarterly basis, to fit in with this requirement. If a report is made around the beginning of a calendar quarter though, where it is then close to 90 days until the next potential release, HARMAN would look at a mid-quarter release so that in general the time to release should only be up to 60 days from the receipt of a vulnerability report.

If a zero-day issue is found – i.e. there is a known exploit for a security vulnerability – then HARMAN will continue Adobe's goal of ensuring customers have a fix within 7 days of the vulnerability being discovered. Should this occur, the top priority for the team will be to notify customers of the issue and the imminent update, and then to roll out the fixed packages to all customers.

Note that even if an exploit is discovered, it should not actually be possible for a HARMAN build of the Flash Player to be compromised by this, as any unknown/uncontrolled SWF content should be blocked by the URL filtering mechanism built into the binary.

Summary

HARMAN offer the packaged browser solution as a simple mechanism to ensure that existing web-based applications that use Flash content can still be run after the Flash Player end of life date. The solution options are both relatively simple applications that run on an end user's computer and use third party components from Microsoft and from Electron.js/Chromium in order to provide the complexities of the web browser engine. The Flash Player is supported by HARMAN and will continue to receive security updates during the supported period (currently scheduled until the end of 2023).

As the application architectures show, there is limited scope for a security issue to be exposed from the host application itself, with the majority of the risk coming from the loading of uncontrolled and malicious content within the browser engine. Whilst this risk is still considered to be load, the fact that the Chromium engine cannot be updated means that some mitigation is recommended. HARMAN's suggested option is to limit the content that the solution can load, so that the end user can only navigate to web pages that are within the customer's control. Such a mechanism is already enforced within the Flash Player, as required by Adobe.

HARMAN continue to maintain and support the Flash Player and have developed processes to ensure that customers are kept as safe and protected as is practical from security problems related to the Flash Player. These have been developed in conjunction with Adobe and aim to follow some of the practices that the Adobe Flash Platform team themselves had been using. For customers with a commercial license from HARMAN, they can be reassured that any security vulnerabilities discovered within the Flash Player will be addressed for the duration of their license.