



Log Uploader tool

The HERE Core Log Uploader is a tool that enables platform owners and application providers to retrieve logs from end-users' machines for troubleshooting purposes. This article walks you through how to configure, install, and use the Log Uploader tool.

When run, the Log Uploader shows the user a window with information about the operation that it performs, which can include a link for the user to learn more. The window includes a list of logs to be uploaded. The user can confirm or disallow the operation. The Log Uploader then sends the logs to an endpoint that you have created and configured it to use.

Requirements

- The Log Uploader is supported only on Windows.
- It requires HERE Core Runtime version 40 or later.

Set up an upload endpoint

You need an endpoint on your network to receive the uploaded files. The Log Uploader makes a POST request to a defined endpoint using `multipart/form-data` encoding. The request contains a header named `file` that points to a zip file named `logs.zip` that contains the log files. Make sure that the endpoint defined in the uploader manifest supports this behavior.

Configure the Log Uploader

To use the Log Uploader, create a manifest for it.

Here's an example of the configuration structure:

```

{
  "logUpload": {
    "endpoint": "UPLOAD-ENDPOINT-URL",
    "manifests": \["*\:\/\/\*.EXAMPLE.COM/PATH-TO-MANIFESTS/
\*.json"\],
    "logs": \["debug", "app", "rvm"\],
    "ui": {
      "title": "WINDOW-TITLE",
      "icon": "URL-OF-ICON-FOR-UPLOADER",
      "header": "COMPANY-NAME",
      "description": "EXPLANATION-OF-WHAT-THE-TOOL-IS-DOING",
      "email": "SUPPORT-EMAIL-ADDRESS",
      "website": "WEBSITE-WITH-MORE-INFO",
      "websiteDescription": "WEBSITE-DESCRIPTION"
    }
  },
  "runtime": {
    "version": "V40-OR-LATER"
  }
}

```

Configuration options

- `logUpload.endpoint`: (Required) The URL where log files are to be uploaded.
- `logUpload.manifests`: (Optional) An array of URL match patterns to define the manifests of the programs that the Log Uploader should upload logs for. Note this affects only app logs, as debug logs are tied to the RVM and not to specific programs. Default is an empty array.
- `logUpload.logs`: (Optional) An array of strings that specifies which logs should be uploaded. Accepts "debug", "app", and "rvm".
- `logUpload.ui`: (Optional) Customization options for the user interface. All of these properties are string values.
 - `title`: Title for the Log Uploader window.
 - `icon`: URL of an icon to use for the Log Uploader tool. The default is the

OpenFin logo.

- `header`: Name of the organization displayed in the Uploader window. The default is "OpenFin".
 - `description`: An explanation for the user of the purpose of the tool. Default: "The following logs are being requested."
 - `email`: An email address that the user can contact with questions. Messages about failed uploads are also sent to this address.
 - `website`: URL of a webpage where the user can find more information about log uploading.
 - `websiteDescription`: Description of the webpage where the user can find more information.
- `runtime.version`: (Optional) The version of the HERE Core Runtime to use with the Log Uploader. The minimum version that supports the Log Uploader is v40; if there might be multiple versions in the user's environment, defining this option is a best practice. Refer to [Runtime properties](#) for details.

Install

You can install the Log Uploader locally like any other HERE Core program. Set the appropriate [Shortcut properties](#) in the manifest for installation. You can distribute it to users as you would any HERE Core program, and instruct them to run it when you need to access logs. Or you can send a user a `fins: link` to the uploader manifest URL, so that they can run it without having previously installed it.

Run the uploader as a HERE app

The uploader manifest can be launched like any other HERE Core manifest. Refer to [Application launch](#) for details of launching a manifest. You can suppress the window by specifying the `--no-ui` option when launching the manifest with `OpenFinRVM.exe`.

Run the uploader programmatically

Starting in version 42, you can run the log uploader by calling a method,

`fin.System.launchLogUploader`. This method enables you to upload logs from within application code, optionally suppressing the dialog box.

! INFO

This method is experimental and subject to change.

You can pass all the same options that you can pass in the manifest (see [Configuration options](#)), via a single `options` argument:

```
type LogUploaderOptions = {
  endpoint: string; // specify the log upload endpoint
  manifests?: string[]; // Array of match patterns to specify apps
to upload logs for
  logs?: LogType[]; // Log types to upload: 'debug' | 'rvm' |
'app'
  ui?: {
    show?: boolean; // whether to show the Log Uploader dialog
    title?: string; // Log Uploader window title
    icon?: string; // Log Uploader icon
    header?: string; // Name of the organization displayed in
the Uploader window
    description?: string; // An explanation for the user of the
purpose of the tool.
    email?: string; // An email address that the user can
contact with questions.
    website?: string; // URL of a webpage where the user can
find more information about log uploading.
    websiteDescription?: string; // Description of the webpage
where the user can find more information.
  }
};
```

The method returns a Promise; if an error occurs during uploading, the Promise rejects,

and the error is returned.

This method is a [Secured API](#), and therefore must be declared in the application manifest and allowed by the desktop owner.

Examples

Launch the Log Uploader but do not show the dialog and only collect logs for specific app manifests:

```
await fin.System.launchLogUploader({
  endpoint: 'https://UPLOAD/ENDPOINT/url?user=USER'
  manifests: [
    'https://*app.com/manifest.json',
    'https://*.secondapp.com/manifest.json'
  ]
  ui: {
    show: false
  }
});
```

Launch the Log Uploader, customize the text, and collect only app logs:

```
await fin.System.launchLogUploader({
  endpoint: 'https://UPLOAD/ENDPOINT/url?user=USER',
  logs: ['app']
  ui: {
    title: 'LOG UPLOADER TITLE',
    icon: 'https://EXAMPLE/icon.png',
    header: 'COMPANY NAME',
    description: 'The following logs are being requested:',
    email: 'SUPPORT@EXAMPLE.COM';
    website: 'https://EXAMPLE.COM/LOG-INFO';
    websiteDescription: 'USER FRIENDLY EXPLANATION';
  }
});
```

Security considerations

The Log Uploader ensures that logs are uploaded only by the same organization that provides the programs whose logs it accesses. Some logs are encrypted to prevent leakage of sensitive data.

URL restrictions

To prevent unauthorized log collection, the main domain (top-level and second-level) must be the same for all of the following:

- The URL of the Log Uploader manifest
- The `manifests` match pattern
- The URL of the manifest for a program whose logs are to be uploaded

For example, a Log Uploader manifest at `https://dev.example.com/log-uploader/manifest.json` must have a `manifests` property value similar to `["*://*.example.com/*/*.json"]` and therefore can upload logs only from manifests hosted on `example.com` or its subdomains.

Log encryption

To protect sensitive information, the *RVM* logs are always encrypted.

Debug logs are encrypted unless the following criteria are met:

- The manifest has a security realm set.
- Only permitted manifests have used this security realm.

If these criteria are met, the debug logs under that security realm are sent unencrypted.

You can send encrypted logs to [HERE](#) to help with troubleshooting support requests.

Program logs are not encrypted.

Troubleshoot errors

If errors occur during log collection or upload, the Log Uploader displays an error message to the user, even in "no UI" mode. The error dialog has minimal dependencies to ensure that it functions even under degraded conditions.

If a public internet connection is available but the upload fails, the Log Uploader sends a failure message to the defined `email` address.