

Compositor

API Reference

© 2018 All rights reserved by Metrological

This document contains information which is proprietary and confidential to Metrological. It is provided with the expressed understanding that the recipient will not divulge its content to other parties or otherwise misappropriate the information contained herein. This information is furnished for guidance; specifications and availability of goods mentioned in it are subject to change without notice. No part of this publication may be reproduced, stored in a database, retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the written prior permission of Metrological.

History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 0.1 | 10-07-2018 | C. Custers | Initial version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Scope 4

1.2 Case sensitivity 4

1.3 Acronyms, Abbreviations and Terms 4

1.4 Standards 4

1.5 References 5

1.6 Open Issues 5

1.7 Limitations 5

2. Rationale behind tracing 6

2.1 Definition, logging versus tracing 6

2.2 Tracing in practice 6

2.3 Configuration 6

2.4 Flow of trace information. 6

3. TraceControl Plugin 8

3.1 Configuration 8

3.2 Application Programming Interface (API) 8

3.2.1 General information 8

3.2.2 State changes 8

3.3 Events 8

3.4 JSON definitions 9

3.4.1 General information (tracing\_info) 9

# Introduction

## Scope

This document describes the Plugin Tracing API interface. This plugin can be configured to be loaded and executed to target trace information for categories or modules (or any given combination) to be turned on and off runtime in the WPEFramework. For details on the WPEFramework API, refer to:

## Case sensitivity

All identifiers on the interface described here are case-sensitive. E.g. an id known in the plugin as 'C0FFEE' is not the same as 'c0ffee'.

All keywords, entities, properties, relations and actions should be treated as case-sensitive.

## Acronyms, Abbreviations and Terms

The next list provides an overview of acronyms and abbreviations used in this document and their definitions.

|  |  |
| --- | --- |
| **Acronym** | **Definitions** |
| API | Application Programming Interface |
| JSON | JavaScript Object Notation |
| UTC | Coordinated Universal Time |

Below terms are listed with their definitions, as used in this document.

|  |  |
| --- | --- |
| **Term** | **Definitions** |
| Callsign | The callsign is the name given to an instance of a plugin. One plugin can be instantiated multiple times, but each instance the instance name, callsign, must be unique. |
| Proxy | An object in one process space representing the “real” object in another process space. The Proxy takes care of marshalling the parameters. |
| Stub | An object in the process space that contains the actual object. The stub takes care of un-marshalling the request from the Proxy and executes the call, on behave of the Proxy object, on the real object |

## Standards

Date time formats between the systems shall be in UTC time and W3C (ISO 8601 profile) formatting [ISO 8601], e.g.: 2004-11-05T13:15:30Z. This way time discontinuities can be avoided due to daylight savings. Note that all interfacing systems must decode/encode the date time to the correct local time.

Languages used in the WPEFramework will be conform [ISO 639-1] using two letter language codes. If WPEFramework encounters a language code it does not recognize, it will use ‘xx’ instead. For a list of available two letter ISO language codes, please visit:  
<http://www.loc.gov/standards/iso639-2/php/code_list.php>

## References

This section lists the references made in this document:

|  |  |
| --- | --- |
| [WPEF] | WPEFramework API Reference  <https://github.com/WebPlatformForEmbedded/WPEFramework> |
| [HTTP] | Hypertext Transfer Protocol  <http://www.w3.org/Protocols> |
| [ISO 8601] | Date and time format  http://www.iso.org/iso/date\_and\_time\_format |
| [ISO-3166] | Country code specification  <http://www.iso.org/iso/country_codes.htm> |
| [ISO-639-1] | Language code specification (Alpha-2 code)  <http://www.loc.gov/standards/iso639-2/php/code_list.php> |
| [JSON] | JavaScript Object Notation  [http://www.json.org](http://www.json.org/) |
| [URLENC] | URL Encoding  <http://www.w3schools.com/tags/ref_urlencode.asp> |

## Open Issues

This is a list of open issues that needs to be resolved:

* This document is still a work in progress.

## Limitations

The information described in this document is preliminary and subject to change in the future.

Legend:



**Be aware of:** implementation choice is needed or side-effect needs to be handled.



**Implementation advice:** Guide line for implementation mostly related to performance.

# Compositor Plugin

## Rationale behind the Compositor plugin.

The WPEFramework allows one to choose between different display-servers (Wayland or Nexus). The plugin which allows this is called: the Compositor. The Compositor acts as a kind of “proxy” display-server, behind which the “real” display-server is running and serving the native platform displays.

## Configuration

|  |  |
| --- | --- |
| callsign | [string] the instance name for the plugin e.g. Compositor. Default: Compositor. |
| classname | [string] Compositor. |
| locator | [string] libWPEFrameworkCompositor.so |
| autostart | [bool] should the browser plugin be instantiated at the moment the WPEFramework is starts up. |
| configuration | [JSON] JSON object specifying the exact configuration for this plugin. See the next paragraph for details. |

Specific Settings:

|  |  |
| --- | --- |
| outofprocess | [bool] execute plugin out-of-process |
| width | [uint32] width of screen |
| height | [uint32] height of screen |

## Application Programming Interface (API)

### General information

Using this method, actual trace status information can be retrieved from the WPEFramework.

|  |  |
| --- | --- |
| Request: | GET /Service/Compositor/Clients or GET /Service/Compositor |
| Success: | HTTP/1.1 200 OK  { compositor\_info } |

### Screen Resolution

Using these methods, the *screen resolution* can be reset. The numbers 1-8 correspond respectively with the screen resolutions:

|  |  |
| --- | --- |
| **<Number>** | **Resolution** |
| 1 | 480i |
| 2 | 480p |
| 3 | 720p |
| 4 | 1080p50Hz |
| 5 | 1080p24Hz |
| 6 | 1080i50Hz |
| 7 | 1080p50Hz |
| 8 | 1080p60hz |

|  |  |
| --- | --- |
| Request: | PUT /Service/Compositor/Resolution/<Number> |
| Success: | HTTP/1.1 200 OK |
| Failure: | HTTP/1.1 400 Bad request |

### State changes of Netflix

Using this method, the Netflix-client with be killed.

|  |  |
| --- | --- |
| Request: | PUT /Service/Compositor/Netflix/Kill |
| Success: | HTTP/1.1 200 OK |
| Failure: | HTTP/1.1 400 Bad request |

Using this method, the opacity of the Netflix-client can be reset. The opacity range is 0-255.

|  |  |
| --- | --- |
| Request: | PUT /Service/Compositor/Netflix/Opacity/<Number> |
| Success: | HTTP/1.1 200 OK |
| Failure: | HTTP/1.1 400 Bad request |

Using this methods the Netflix-client’s visibility can be toggled between visible and invisible.

|  |  |
| --- | --- |
| Request: | PUT /Service/Compositor/Netflix/Visible/{Show,Hide} |
| Success: | HTTP/1.1 200 OK |
| Failure: | HTTP/1.1 400 Bad request |

Using this method the geometry of the Netflix-client’s display can be reset. The origin (x,y) combined width the size (width, height) of the display have to be given.

|  |  |
| --- | --- |
| Request: | PUT /Service/Compositor/Netflix/Geometry/x/y/width/height |
| Success: | HTTP/1.1 200 OK |
| Failure: | HTTP/1.1 400 Bad request |

Using this method the Netflix-client’s display becomes the top-layer.

|  |  |
| --- | --- |
| Request: | PUT /Service/Compositor/Netflix/Top |
| Success: | HTTP/1.1 200 OK |
| Failure: | HTTP/1.1 400 Bad request |

Using the method the Netflix-client’s display becomes the focus of the input device(s).

|  |  |
| --- | --- |
| Request: | PUT /Service/Compositor/Visible/{Show,Hide} |
| Success: | HTTP/1.1 200 OK |
| Failure: | HTTP/1.1 400 Bad request |

## JSON definitions

### General information

|  |  |
| --- | --- |
| outofprocess | [bool] execute out-of-process. |
| width | [uint32] screenwidth |
| hidden | [uint32] screenheight |