

GENIVI Document HMIEG001

Component Specification for Popup Manager

Revision Date: 11.06.2103

Sponsored by: GENIVI Alliance

Copyright (c) 2012 by the GENIVI Alliance.

2400 Camino Ramon, Suite 375, San Ramon, CA 94583, USA

http://www.genivi.org

All rights reserved.

Permission is granted to members of the GENIVI Alliance to reproduce this document for their own use or the use of other

GENIVI Alliance members only, provided this notice is included. All other rights reserved. Duplication for sale, or for

commercial or for-profit use is strictly prohibited without the prior written consent of the GENIVI Alliance.

Copyright © GENIVI Alliance, Inc. (2012). All rights Reserved. This information within this

document is the property of the GENIVI Alliance and its use and disclosure are restricted.

Elements of GENIVI Alliance specifications may be subject to third party intellectual property rights,

including without limitation, patent, copyright or trademark rights (such a third party may or may not

be a member of GENIVI). GENIVI is not responsible and shall not be held responsible in any

manner for identifying or failing to identify any or all such third party intellectual property rights.

This document and the information contained herein are provided on an “AS IS” basis and GENIVI

DISCLAIMS ALL WARRANTIES EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO (A)

ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY

RIGHTS OF THIRD PARTIES (INCLUDING WITHOUT LIMITATION ANY INTELLECTUAL

PROPERTY RIGHTS INCLUDING PATENT, COPYRIGHT OR TRADEMARK RIGHTS) OR (B) ANY

IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE,

TITLE OR NON-INFRINGEMENT. IN NO EVENT WILL GENIVI BE LIABLE FOR ANY LOSS OF

PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR

FOR ANY OTHER DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTAL, PUNITIVE OR

CONSEQUENTIAL DAMAGES OF ANY KIND, IN CONTRACT OR IN TORT, IN CONNECTION

WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF

THE POSSIBILITY OF SUCH LOSS OR DAMAGE. All Company, brand and product names may be

trademarks that are the sole property of their respective owners.

The above notice and this paragraph must be included on all copies of this document that are made.

GENIVI Alliance, Inc.

2400 Camino Ramon, Suite 375

San Ramon, CA 94583, USA

[1. Change History 5](#_Toc358808221)

[2. Introduction 6](#_Toc358808222)

[3. Terminology 7](#_Toc358808223)

[4. Architetcure 8](#_Toc358808224)

[4.1 Overview 8](#_Toc358808225)

[4.2 Popup manager internals 9](#_Toc358808226)

[5. Enums, Structs and API’s definitions 10](#_Toc358808227)

[5.1 Enums declaration 11](#_Toc358808228)

[5.1.1 Modality 11](#_Toc358808229)

[5.1.2 HorizontalAlignment 12](#_Toc358808230)

[5.1.3 VerticalAlignment 12](#_Toc358808231)

[5.1.4 Scaling 12](#_Toc358808232)

[5.2 Structs declaration 14](#_Toc358808233)

[5.2.1 RequestArg 14](#_Toc358808234)

[5.2.2 DimensionArg 15](#_Toc358808235)

[5.2.3 PopupPresentationArg 15](#_Toc358808236)

[5.3 API’s 16](#_Toc358808237)

[5.3.1 RegisterSurface 16](#_Toc358808238)

[5.3.2 ShowPopup 17](#_Toc358808239)

[5.3.3 HidePopup 17](#_Toc358808240)

[5.3.4 OnError 18](#_Toc358808241)

[5.3.5 OnSurfaceStateChanged 19](#_Toc358808242)

[6 Sequence Diagrams 20](#_Toc358808243)

[6.1 Display and surface registration 20](#_Toc358808244)

[6.2 Show popups 21](#_Toc358808245)

[6.3 Hide popups 22](#_Toc358808246)

[7.0 References 23](#_Toc358808247)

# Change History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change** |
| 0.1 | 27.05.2013 | Gururaja N | Initial version,  Inputs (Franca definition) from Fujitsu has been used while preparing this document |
| 0.2 | 04.06.2013 | Gururaja N | Updated after the review comment from Fujitsu |
| 0.3 | 10.06.2013 | Gururaja N | Following sections are updated after review comment from Fujitsu.  Section 5.1 enums are modified  Section 5.2 : some interfaces are removed.  5.3.6 New argument is added to error info  6.1 sequence chart modified |
| 0.4 | 11.06.2013 | Gururaja N | Added notes to 5.1  Added OnError condition to show and hide popup sequence diagram |
| 0.5 | 11.06.2013 | Gururaja N | Formatting of the document  Added genivi logo and copyright details |
| 0.6 | 12.06.2013 | Gururaja N | Updated after review comments   * Component interface diagram is corrected sec 5.0 |

# Introduction

This document contains API specification for the popup manager. The below diagram depicts the popup manager component which is a part of HMI application FW.

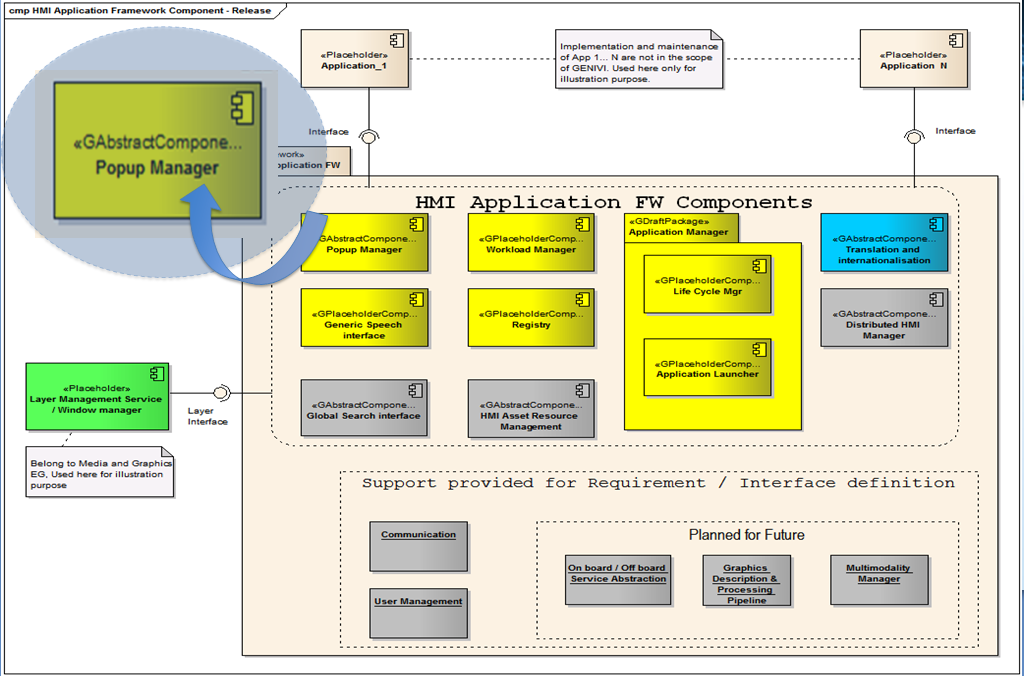


Fig (1)

For more info on the framework Refer

<https://collab.genivi.org/wiki/display/genivi/HMI+Application+Framework+Overview>

# Terminology

|  |  |
| --- | --- |
| HMI | Human Machine Interface |
| FW | Framework |
| ILM | Layer Manager |
|  |  |
|  |  |

# Architetcure

## 4.1 Overview

The following picture depicts the interaction of popup manager with other components. The applications can request popup manager to show/hide the popups. Even the system services that are running on the system can request popup manager to display the popups.

On the other hand popup manager interacts with the component that is responsible for providing the screens/surface/Layer to show the popups.

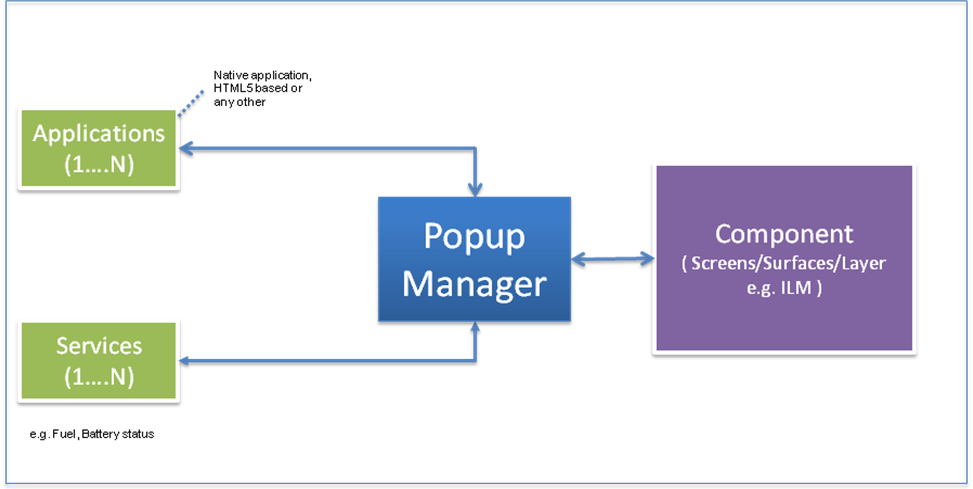
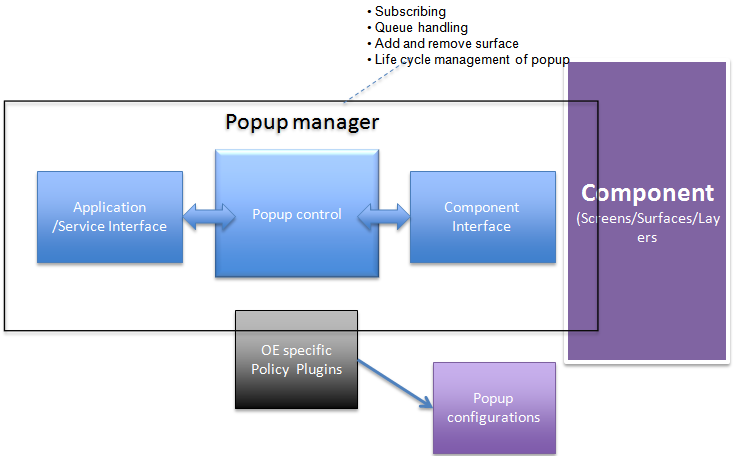


Fig (2)

## 4.2 Popup manager internals



The popup manager contains popup control internally which takes care of subscribing, queue handling, Add and remove surfaces and life cycle management of the popup. This also provides interface to Application/ Services (open point: since services are not having any UI how to process the service requests?). On the other hand it communicates with a component that is responsible for handling display, screens, surfaces and layers.

This also provides a provision for OE specific policy handling through a plugin. The popup manager forwards the application data like App ID, Request ID and the data unmodified to the OE specific policy plugin component. Based on the response from the plugin further processing is carried out.

# Enums, Structs and API’s definitions



## 5.1 Enums declaration

Note : The enumeration values shall be accessible via its own enumeration scope (e.g. class, namespace)”. Reason is that some of them have overlapping names, which may lead to definition (later compile) problems.

For details please refer franca IDL file for the popup manager.

### 5.1.1 Modality

|  |  |  |
| --- | --- | --- |
| Enum Type : Modality | The enumeration of pop-up overlay modalities |  |
| *None* | Indicates a non-modal surface |  |
| *Application* | Indicates an application-modal surface. User interaction is application-wide only possible with the regarding surface. User interaction with other applications is not affected |  |
| *System* | Indicates a system-modal surface. User interaction is system-wide only possible with the regarding surface |  |

### 5.1.2 HorizontalAlignment

|  |  |  |
| --- | --- | --- |
| ***Enum Type :*** HorizontalAlignment | The enumeration of alignment values intended to use for horizontal attributes |  |
| *Left* | Align elements to left side of the parent element |  |
| *Center* | Align elements to the center of the parent element |  |
| *Right* | Align elements to right side of the parent element |  |

### 5.1.3 VerticalAlignment

|  |  |  |
| --- | --- | --- |
| ***Enum Type :*** VerticalAlignment | The enumeration of alignment values intended to use for vertical alignment attributes |  |
| *Top* | Align elements to top of the parent element |  |
| *Center* | Align elements to the center of the parent element |  |
| *Bottom* | Align elements to the bottom of the parent element |  |

### 5.1.4 Scaling

|  |  |  |
| --- | --- | --- |
| ***Enum Type :*** Scaling | The enumeration of scaling configuration values intended to use for element scaling attributes |  |
| *None* | Indicates that no scaling shall be performed |  |
| *Cut* | Indicates that the element shall be scaled to fully cover to it's parent element by maintaining the aspect ratio of the element. The element is scaled in a way that one dimension matches the size of the corresponding parent element while the other dimension of the element may be larger than the corresponding parent element size. |  |
| *Fit* | Indicates that the element shall be scaled to fit to it's parent element by maintaining the aspect ratio of the element. The element is scaled in a way that one dimension matches the size of the corresponding parent element while the other dimension of the element may be less than the corresponding parent element size. |  |
| *Stretch* | Indicates that the element shall be scaled to the dimension of it's parent element without maintaining the aspect ratio of the element. |  |

**5.1.5 SurfaceState**

The enumeration of the surface states.

|  |  |  |
| --- | --- | --- |
| ***Enum Type :*** SurfaceState | Note: The enumeration values shall be accessible via its own enumeration scope (e.g. class, namespace). |  |
| *Unqueued* | Indicates that the surface has been removed from presentation queue |  |
| *Queued* | Indicates that the surface is queued for later presentation |  |
| *Unmapped* | Indicates that the surface is unmapped and not shown on screen. |  |
| *Mapped* | Indicates that the surface is mapped and shown on screen. |  |

## 5.2 Structs declaration

### 5.2.1 RequestArg

|  |  |  |
| --- | --- | --- |
| ***struct Type :*** RequestArg | This enum type encapsulates standard arguments for any Application requests |  |
| UInt32 requestId | The application specific ID of the request. If a corresponding response is sent back, the request ID of the response matches this argument |  |
| UInt32 appId | The application specific ID. This ID is intended to identify the application at project specific plug-ins. It is neither processed nor modified within generic popup implementation. |  |
| UInt32 userData | The application specific data intended for extra tagging or hinting for request processing by project specific plug-ins. This value is neither processed nor modified within the popup manager. |  |

### 5.2.2 DimensionArg

|  |  |  |
| --- | --- | --- |
| ***struct Type*** : DimensionArg | This class encapsulates arguments for providing dimensions related info |  |
| UInt32 width | The horizontal dimension value |  |
| UInt32 height | The vertical dimension value |  |

### 5.2.3 PopupPresentationArg

|  |  |  |
| --- | --- | --- |
| ***struct Type*** : DimensionArg | This struct encapsulates presentation arguments for pop-up overlays |  |
| UInt32 priority | The priority of the pop-up |  |
| enum.Modality modality | The modality of the pop-up |  |
| UInt32 presentationTime | The optional time period after which the pop-up is hidded automatically |  |
| UInt32 minimumPresentationTime | The optional time period for which a pop-up shall be at least be visible |  |
| UInt32 validityPeriod | The optional time period after which a not presented pop-up is discarded |  |
| Types.HorizontalAlignment horizontalAlignment | The optional horizontal alignment attribute of surface on the screen area |  |
| enum.VerticalAlignment verticalAlignment | The optional vertical alignment attribute of surface on the screen area |  |
| enum.Scaling scaling | optional scaling attribute of surface on the screen area |  |

## 5.3 API’s

### 5.3.1 RegisterSurface

Register the ILM surface addressed by "surfaceId" to the screen area addressed by "screenAreaId". Surface specific hinting information is provided in field "userData".

This API is a fire and forget type.

In Parameters

|  |  |
| --- | --- |
| String clientId | Unique client identification, identifies the sender of the request. |
| RequestArg requestArg | General application specific request information (will be returned unmodified to client in response) |
| UInt32 screenAreaId | The ID of the screen area to address |
| UInt32 surfaceId | The ID of the surface to address. |

Out parameters

|  |  |
| --- | --- |
| None |  |

### 5.3.2 ShowPopup

Request the presentation of the surface pop-up overlay addressed by "surfaceId". The surface must be registered on a screen area dedicated for pop-ups in order to be presented as pop-up. Additional arguments are passed to activator plug-ins.

In parameters

|  |  |
| --- | --- |
| String clientId | Unique client identification, identifies the sender of the request |
| RequestArg.requestArg | General application specific request information (will be returned unmodified to client in response) |
| UInt32 surfaceId | The ID of the surface to address |
| PopupPresentationArg. popupPresentationArg | The presentation arguments for pop-up overlays |

Out Parameters

|  |  |
| --- | --- |
| None |  |

### 5.3.3 HidePopup

Requests the concealing of the surface pop-up overlay addressed by "surfaceId"

In Parameters

|  |  |
| --- | --- |
| String clientId | Unique client identification, identifies the sender of the request |
| RequestArg requestArg | General application specific request information (will be returned unmodified to client in response) |
| UInt32 surfaceId | The ID of the surface to address |

Out Parameters

|  |  |
| --- | --- |
| None |  |

### 5.3.4 OnError

The error response to any failed request providing information about the error.

Note: this error response is only to be used for runtime errors (e.g. memory allocation fails, etc.), but not for programming or configuration errors (these will be seen in the DLT logs).

In parameters

|  |  |
| --- | --- |
| String clientId | Unique client identification identifies the receiver of the response. |
| RequestArg requestArg | General application specific request information (unmodified client request information) |
| Int32 systemErrno | System error codes (like linux extended error codes defined in linux/errno.h) |
| Int32 presentationErrno | Error codes specific to the visual presentation module screen broker is connected to (e.g. layer manager) |

Out parameters

|  |  |
| --- | --- |
| None |  |

### 5.3.5 OnSurfaceStateChanged

Notify the clients that the state of the indicated surface has changed.

In parameter

|  |  |
| --- | --- |
| String clientId | Unique client identification, identifies the receiver of the response. |
| UInt32 userData | The user defined data |
| UInt32 surfaceId | The ID of the affected surface |
| SurfaceState surfaceState | The new state of the surface |

Out parameters

|  |  |
| --- | --- |
| None |  |

# 6 Sequence Diagrams

## 6.1 Display and surface registration

Note: The interfaces shown here is at the moment not a part of popup manager except *Register surface* but essential for the Application to link surface with displays and the outcome is needed for the popup manager.



## 6.2 Show popups



6.3 Hide popups



# 7.0 References

[1] Following is the link to HMI Application Framework illustrating the components which has popup manager.

<https://collab.genivi.org/wiki/display/genivi/HMI+Application+Framework+Overview>

[2] Here is the link to Popup manager

<https://collab.genivi.org/wiki/display/genivi/HMI_APP_FW_Pop-up+Management>

[3] Direct link to Use cases and requirements of popup manager

<https://collab.genivi.org/wiki/display/genivi/Popup+Management+-+Usecases+and+Requirements>

[4] The UML model is available in the svn branch and the link is

<https://svn.genivi.org/uml-model/genivi/branch/eg-hmi-popmgr>

Use cases are @ GENIVI Model -> Use case View -> HMI Application Framework -> popup manager

Requirements are @ GENIVI Model -> Requirements View -> HMI Application Framework -> popup manager

[5] This document and the franca IDL

<https://collab.genivi.org/wiki/display/genivi/Poupmanager++Component+specification>