

# Zend\_Service\_Windows\_Live\_Mesh - Ivo Jansch

**Technology Preview**

Live Mesh is currently in Technology Preview state. As such, the API is subject to change. Still, I'd like to work early on this so we can have the component available when Mesh is out of technology preview state.



## Discussion points

The following points are still up for discussion:

- The Mesh uses 'devices'. Devices are someone's pc, his mobile, his media center etc. Is it feasible to think that someone's website is a 'device' as well? In other words, should we register the site using the Mesh services as a device?

## Zend Framework: Zend\_Service\_Windows\_Live\_Mesh Component Proposal

<b>Proposed Component Name</b>	Zend_Service_Windows_Live_Mesh
<b>Developer Notes</b>	<a href="http://framework.zend.com/wiki/display/ZFDEV/Zend_Service_Windows_Live_Mesh">http://framework.zend.com/wiki/display/ZFDEV/Zend_Service_Windows_Live_Mesh</a>
<b>Proposers</b>	Ivo Jansch
<b>Zend Liaison</b>	TBD
<b>Revision</b>	1.0 - 30 October 2008: Initial Draft. (wiki revision: 7)

## Table of Contents

1. Overview
2. References
3. Component Requirements, Constraints, and Acceptance Criteria
4. Dependencies on Other Framework Components
5. Theory of Operation
6. Milestones / Tasks
7. Class Index
8. Use Cases
9. Class Skeletons

### 1. Overview

Zend\_Service\_Windows\_Live\_Mesh is a component that allows applications to make use of the API's of the Windows Live Mesh platform, a platform by Microsoft that allows the sharing of data across devices and that offers integration with Windows Live services

### 2. References

- [Windows Live Mesh](#)

### 3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** allow the use of Windows Live Mesh services within PHP applications.
- This component **will** use the RESTful API that Windows Live Mesh provides.
- This component **will** use Windows Live authentication services so personalized Mesh data can be retrieved.
- This component **can** use Zend\_Cache to store data retrieved from the Mesh. By default, only Zend\_Session will be used.
- This component **will** throw Exceptions upon failure.

### 4. Dependencies on Other Framework Components

- Zend\_Service\_Rest
- Zend\_Session
- Zend\_Cache
- Zend\_Exception

### 5. Theory of Operation

The component will connect to the REST interface that Windows Live Mesh offers, and will provide convenient access to the services of the platform. Convenience classes and methods will be implemented so that Live Mesh can be accessed from a Zend Framework application transparently.

### 6. Milestones / Tasks

- Milestone 1: The exact API for the Windows Live Mesh component will be defined
- Milestone 2: Working proof of concept supporting use case #1 will be checked into the incubator
- Milestone 3: Working prototype checked into the incubator supporting the remaining use cases will be checked into the incubator.
- Milestone 4: Unit tests exist, work, and are checked into SVN.
- Milestone 5: Initial documentation exists.

### 7. Class Index

- Zend\_Service\_Windows\_Live\_Mesh
- Zend\_Service\_Windows\_Live\_Mesh\_Object
- Zend\_Service\_Windows\_Live\_Mesh\_Folder

### 8. Use Cases

UC-01

A script authenticates itself against the Live\_Mesh service using the connect method of Zend\_Service\_Windows\_Live\_Mesh.

## UC-02

A script using this component will connect to Live Mesh and display a list of folders.

## UC-03

A script using this component will upload and download files to one of the Live Mesh folders.

## 9. Class Skeletons

```
class Zend_Service_Windows_Live_Mesh_Exception extends Zend_Exception {}

class Zend_Service_Windows_Live_Mesh {
public function connect();
public function getFolders();
}

class Zend_Service_Windows_Live_Mesh_Folder {
public function getName();
public function getContents();
public function getObjectMetaData();
public function getObject();
public function uploadObject();
}

class Zend_Service_Windows_Live_Mesh_Object {
public function getName();
public function getData();
}
```

]]></ac:plain-text-body></ac:macro>  
]]></ac:plain-text-body></ac:macro>