



VSERV Integration into Mobile Marmalade Android Applications

Version 1.1

Vserv Digital Solutions Pvt Ltd

605,Eco House,Vishweshwar Nagar,
Off. Aarey Road, Goregaon(E),
Mumbai – 400063.
Tel No: +91 22 29275746/82
www.vserv.mobi

CONFIDENTIAL DOCUMENT

No part of this document may be disclosed verbally or in writing, including by reproduction, to any third party without the prior written consent of Vserv Digital Solutions Pvt Ltd. This document, its associated appendices and any attachments remain the property of Vserv Digital Solutions Pvt Ltd and shall be returned upon request.

Static Library Implementation Help Document

Please use the below written permission in target app's manifest file:-

INTERNET, ACCESS_NETWORK_STATE, WRITE_EXTERNAL_STORAGE, SEND_SMS.

Set Screen Orientation to Portrait mode.

Please include the extracted Library folder path in the target app's .mkb file as:-

```
subprojects
{
    ../ VservLibrary/WebViewMarmalade
}
```

Procedure to Call the lib from the client is written below:-

Vserv library initialization code written below should be called once.

```
std::hash_map<char*, char*> hashMap;
```

Assigning values to hashMap variable-

```
void hashInit()
{
    hashMap["ShowAt"] = "both"; // "start"; //"end";
    hashMap["ViewMandatory"] = "false"; //"true";
    hashMap["ZoneID"] = "xyz"; // ZoneId
}
```

Call hashInit method during initialization-

```
hashInit();
```

Initialize vserv library with hashMap as parameter.

```
vservManagerInit(hashMap);
```

Call this function to fetch the ad -

```
vservManagerFetchingAdData();
```

Call this function to show target app's UI

```
vservManagerSkipAd();
```

Target app's implementation example:-

```
extern void vservManagerSkipAd();
extern int vservManagerFetchingAdData();
extern void vservManagerInit(std::hash_map<char*,char*> p_hashMap);
std::hash_map<char*, char*> hashMap;
void hashInit() {
    hashMap["ShowAt"] = "both";//"start";//"end";
    hashMap["ViewMandatory"] = "false"; //"true";
    hashMap["ZoneID"] = "xyz"; // ZoneID
}
void HelloWorldShutDown()
{
    std::hash_map<char*, char*>::iterator ihashMap;
    if(hashMap.size() > 1)
    {
        ihashMap = hashMap.find("ShowAt");
        s3eSurfaceUnRegister(S3E_SURFACE_SCREENSIZE,
        HelloWorldRotationCallBack);
        if(ihashMap->second == "end" || ihashMap->second == "both")
        {
            vservManagerFetchingAdData();
        }
        else if(ihashMap->second == "start")
```

```

        {
            s3eDeviceExit();
        }
    }
}

int HelloWorldMain()
{
    HelloWorldInit();
    while (!s3eDeviceCheckQuitRequest())
    {
        s3eDeviceYield(0);
        s3eKeyboardUpdate();
        s3ePointerUpdate();
        HelloWorldRender();
        if(!HelloWorldUpdate())
            break;
    }
    HelloWorldShutDown();
    return 0;
}

void vservManagerSkipAd()
{
    HelloWorldMain();
}

```

```

int main()
{
    std::hash_map<char*, char*>::iterator ihashMap;
    hashInit();
    vservManagerInit(hashMap);
    if(hashMap.size() > 1)
    {
        ihashMap = hashMap.find("ShowAt");
        if(ihashMap->second == "start" || ihashMap->second == "both"){
            vservManagerFetchingAdData();
        }
        else
        {
            HelloWorldMain();
        }
    }
    return 0;
}

```

Note:-

In the target app please use and include all the libraries used in the example app.