HOM*e*MEETING

Web JoinNet Technical Guide

April 2016

A HomeMeeting Inc. Official Document

Disclaimer; No Warranty

THIS INFORMATION AND ALL OTHER DOCUMENTATION (IN PRINTED OR ELECTRONIC FORM) ARE PROVIDED FOR REFERENCE PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THIS INFORMATION, THIS INFORMATION AND ALL OTHER DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT ANY WARRANTY WHATSOEVER AND TO THE MAXIMUM EXTENT PERMITTED, HOMEMEETING INC. DISCLAIMS ALL WARRANTIES, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE SAME. HOMEMEETING INC. SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION, DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES, ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS INFORMATION OR ANY OTHER DOCUMENTATION. NOTWITHSTANDING ANYTHING TO THE CONTRARY, NOTHING CONTAINED IN THIS INFORMATION OR ANY OTHER DOCUMENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM HOMEMEETING INC. (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF THIS SOFTWARE.

Copyright

Under the copyright laws, neither this documentation nor the software may be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of HomeMeeting Inc., except in the manner described in the documentation or the applicable licensing agreement governing the use of the software.

© Copyright 2016   HomeMeeting Inc
                     15357 NE 90th Street
                     Redmond, WA 98052

All Rights Reserved. Printed in the United States

CONTENT

1. Background
2. Customization
3. File Structure
4. Module Structure

1. Background

This document complements the Web JoinNet User Guide:

http://www.homemeeting.com/web/Web JoinNet User Guide.pdf

2. Customization

Similar to the native JoinNet, you may want to do some customization on the Web JoinNet. We can release the full source codes of the Web JoinNet to our agents. With the source codes, you can make necessary changes, or even write your own Web JoinNet from the ground up.

There are two levels of customization.

2.1 Built-in customization

We support some built-in customization items in the customization/hmtg.js file. By modifying the values of the entries in the file, a different Web JoinNet is created.

2.2 Advanced customization

If the items in the customization/hmtg.js are not enough for your requirement, you can make changes to the provided source codes, which implement all the logics in the Web JoinNet.

The Web JoinNet heavily uses Angular and Bootstrap UI. You may want to be familiar with them to do any advanced customization. For your reference, here are some resources:

https://docs.angularjs.org/api

http://angular-ui.github.io/bootstrap/

For each minified javascript file, we also provide its corresponding source file. You can modify the source files during your customization process but you MUST minify the modified source files for the production deployment. The un-minified source files SHOULD NEVER be copied to the customer's site.

We use Google Closure Complier to do the minification:

```
java -jar compiler.jar --compilation_level SIMPLE_OPTIMIZATIONS --
js_output_file hmtgs.min.js --create_source_map hmtgs.min.js.map --js
hmtgs.js
```

You can download the Google Closure Complier from
https://developers.google.com/closure/compiler/

For testing, you need to set up a local HTTP server. We use python for this:

```
python -m SimpleHTTPServer 8080
```

If you deploy the Web JoinNet to a different site, the site must be configured to support all the file types used in the Web JoinNet, such as .woff, .svg, .ogg, etc.

3. File Structure

The Web JoinNet contains several sub-folders:

css

This folder contain CSS files. Most are from public libraries such as bootstrap. We create file style.css in this folder as one part of the Web JoinNet.

fonts

This folder contains the font files.

img

This folder contains image files.

lang

This folder contains language files.

lazy_htm

This folder contains lazy-loading html files.

lazy_js_min

This folder contains lazy-loading javascript files.

media

This folder contains audio/video files.

template

This folder contains template files.

worker

This folder contains web workers, mostly used for codec.

Under the main folder, there are the following files:

index.htm

This is the main HTML file

dep.min.js

This file contains all the libraries that the Web JoinNet depends on.

hmtgs.min.js

This file is the main Javascript file (minified from hmtgs.js).

customization.js

This file is the customization file.

JoinNet Wep App User Guide.pdf

This file is the User Guide

loading.htm

This file is used for joint browsing. Some content providers don't allow their pages to be displayed in embedded browsers. Before trying to show a joint browsing page, the Web JoinNet first show this loading.htm. If the joint browsing page cannot be shown and the browser(such as Chrome) fails to show any error, the user will see this loading.htm, not the previous page.

4. Module Structure

hmtgs.js(minified to hmtgs.min.js) is the main Javascript file. It contains the following modules:

The main module is "hmtgs".

The module "hmtgs" depends on two public modules "pascalprecht.translate" and "ui.bootstrap". "hmtgs" also depends on two other modules "joinnet" and "msgr".

The module "hmtgs" contains the following components:

Service "appVersion": provide the version string for the Web JoinNet.

Config "$compileProvider": allow the Web JoinNet to show blob URL and show image using blob URL.

Config "$translateProvider": configure module "pascalprecht.translate" for multi-language and language detection.

run: entry point of the Web JoinNet

Service "appSetting": maintain settings of the Web JoinNet; provide method to switch to download tab

Service "jnjSource": provide source jnj

Service "jnjContent": parse the jnj content

Controller "JnjSourceCtrl": in charge of the "Open Jnj..." section

Controller "LogCtrl": in charge of the "Log" section

Controller "WebAppSettingCtrl": in char of the Web App Tab in "Options section

Controller "MsgrSettingCtrl": in charge of the Messenger Tab in "Options" section

Controller "JoinNetSettingCtrl": in charge of the JoinNet Tab in "Options" section

Controller "AboutCtrl": in charge of the About in the main page

Controller "LangCtrl": in charge of the Language option box on the navigation bar

Controller "ResetAllCtrl": in charge of the "Reset All" Tab in "Options" section

Controller "BrowserDetectCtrl": in charge of Browser detection part in the main page

Service "hmtgHelper": Web JoinNet level helper functions; define all message names.

Service "hmtgAlert": the model of popup alert

Controller "AlertCtrl": in charge of popup alert

Service "hmtgSound": media-related functions

Controller "TestCtrl": internal test, not visible to the user

Config "$translateProvider": resource files for multi-language support

The module "joinnet" depends on public module "pascalprecht.translate".

The module "joinnet" contains the following components:

Service "JoinNet": provide all callback function from jnkernel. After all callback function assignments, this service must call "jnkernel.jn_command_CallbackReady();". provide the function to start a JoinNet session from a jnj.

Controller "JoinnetCtrl": in charge of the "JoinNet" section

Controller "DurationCtrl": in charge of the duration string

Service "statistics": the model of Statistics

Controller "StatCtrl": in charge of Statistics

Service "userlist": the model of Control Panel

Service "main_video_canvas": the model of the Main Video Canvas

Service "main_video": the model of the Main Video Components

Controller "UserListCtrl": in charge of Control Panel

Controller "MeetingControlModalCtrl": in charge of Set Controller dialog box

Controller "SnapshotModalCtrl": in charge of Snapshot dialog box

Controller "PollRequestModalCtrl": in charge of Poll Request dialog box

Controller "PollResultModalCtrl": in charge of Poll Result dialog box

Controller " PromptPasswordModalCtrl": in charge of Prompt Password dialog box

Service "chat": the model of text chat

Controller "ChatCtrl": in charge of text chat

Service "board": the model of white board

Controller "BoardCtrl": in charge of the white board

Controller "UploadSlideModalCtrl": in charge of the upload slide dialog box

Controller "DeleteSlideModalCtrl": in charge of the delete slide dialog box

Service "browser": the model of joint browsing

Controller "BrowserCtrl": in charge of the joint browsing

Service "playback": the model of JoinNet playback session

Controller "JNRCtrl": in charge of the jnr component in JoinNet section

Controller "PlaybackCtrl": in charge of the JoinNet playback

Controller "PlaybackJumpToModalCtrl": in charge of the Jump To dialog box

Service "joinnetHelper": provide JoinNet level helper functions

Controller "WebOfficeVisitorModalCtrl": in charge of the Vistor/Owner dialog box

Service "audio_playback": the model of audio playback

Service "audio_capture": the model of audio capture

Controller "JoinNetAudioCtrl": in charge of the audio-related component of JoinNet section

Controller "JoinNetVideoCtrl": in charge of the video-related component of JoinNet section

Service "audio_codec": the model on generic audio codec

Controller "JoinNetAdvancedCtrl": in charge of advanced functions of JoinNet

Service "video_recving": the model of video receiving status

Service "video_playback": the model of video playback

Service "video_capture": the model of video capture

Service "video_bitrate": the model of video bitrate

Service "video_codec": the model on generic video codec

Service "reconnectName": the model of reconnectName

Controller "ReconnectNameCtrl": in charge of ReconnectName tab

Service "mypicture": the model of My Picture

Controller "MyPictureCtrl": in charge of My Picture setting tab

The module "msgr" depends on public modules "pascalprecht.translate" and "ui.bootstrap".

The module "msgr" contains the following components:

Service "Msgr": provide all callback function from jmkernel. After all callback function assignments, this service must call "jmkernel.jm_command_CallbackReady();".

Controller "MsgrCtrl": in charge of the "Messenger" section

Controller "MsgrActionHeaderCtrl": in charge of the Messenger header area

Controller "MsgrSigninModalCtrl": in charge of the sign-in dialog box

Controller "AppMessageBoxModalCtrl": in charge of the message box

Service "jnagentDlg: the model of the messenger main window

Service "msgrIcon": the icons of the messenger

Service "msgrHelper": provide messenger-level helper functions.