



Intel® Endpoint Management Assistant (Intel® EMA)

JavaScript Libraries Guide

Intel® EMA Version: 1.6.0

Document update date: Tuesday, November 23, 2021

Legal Disclaimer

Copyright 2018-2021 Intel Corporation.

This software and the related documents are Intel copyrighted materials, and your use of them is governed by the express license under which they were provided to you ("License"). Unless the License provides otherwise, you may not use, modify, copy, publish, distribute, disclose or transmit this software or the related documents without Intel's prior written permission.

This software and the related documents are provided as is, with no express or implied warranties, other than those that are expressly stated in the License.

Intel technologies may require enabled hardware, software or service activation.

No product or component can be absolutely secure.

Your costs and results may vary.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Intel does not assume any liability for lost or stolen data or systems or any damages resulting from such losses. Check with your system manufacturer or retailer or learn more at <http://www.intel.com/technology/vpro>.

Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

1 Introduction	1
1.1 Overview	1
1.2 High-level flow of operation	1
1.3 File list	2
1.4 Authentication	3
1.5 Authorization	3
2 ema.js	4
2.1 Main entry point	4
2.2 Starting/stopping the connection to the Intel® EMA AJAX server	4
2.3 Endpoint group list	5
2.4 Endpoint list for the tracked endpointGroups	5
2.5 Endpoint list for the target endpoint groups	6
2.6 Endpoint information	6
2.7 Endpoint group or endpoint object from the tracked list	7
2.8 Endpoint routing type	7
2.9 WMI queries and methods	8
2.10 Processes list and operations	9
2.11 File search	9
2.12 Swarm server live statistics	10
2.13 Encrypted routing cookie	11
2.14 Power polling rate boosting	12
2.15 Event callbacks	12
2.15.1 onDebugMessage event	12
2.15.2 onStateChanged event	12
2.15.3 onEndpointsListChanged event	13
2.15.4 onEndpointGroupsListChanged event	13
2.15.5 onWmiResponse event	13
2.15.6 onSearchResponse event	13
2.15.7 onServerStats event	14
2.16 JSON object definitions	14
2.16.1 ENDPOINT_GROUP object	14
2.16.2 ENDPOINT_INTERFACE object	15
2.16.3 ENDPOINT_AMT object	15
2.16.4 ENDPOINT object	15
2.16.5 WMI_DATA object	16

2.16.6 FILE_SEARCH_DATA object	17
2.16.7 FILE_OPERATOR object	17
2.16.8 FILE_OPERATOR_RESULT object	18
2.17 Enumeration object definitions	18
2.17.1 AJAX server connection state	19
2.17.2 Swarm server tunnel connection state	19
2.17.3 Swarm server tunnel connection's error code	19
2.17.4 AJAX server connection mode	20
2.17.5 User's allowed action on endpoint group	20
2.17.6 File browsing object type	20
2.17.7 File upload state	20
2.17.8 File upload error for scheduled task	21
2.17.9 File upload error for metadata	21
2.17.10 Intel® EMA agent types	22
2.17.11 Endpoint power state	22
2.17.12 WMI result data type	22
2.17.13 Intel® AMT provision state	23
2.17.14 Intel® AMT provision mode	23
2.17.15 Intel® AMT AJAX path error type	24
2.17.16 Intel® AMT WSMAN error type	24
2.17.17 Remote desktop related enumeration	24
2.17.18 Remoter terminal related enumeration	25
3 ema_files.js	26
3.1 Access and policy checking	26
3.2 Main entry point	26
3.3 Starting the tunnel connection to endpoint via Swarm server	27
3.4 Files browsing	27
3.5 Files uploading	28
3.6 Files upload cancellation	28
3.7 File downloading	28
3.8 File downloading cancellation	29
3.9 File/folder moving/renaming	29
3.10 File/folder removal	30
3.11 Folder creation	31
3.12 Event callbacks	31

3.12.1 onDebugMessage event	31
3.12.2 onStateChanged event	32
3.12.3 onFilesResponse event	32
3.12.4 onUploadState event	32
3.13 JSON object definitions	33
3.13.1 FILE_ITEM object	33
4 ema_desktop.js	34
4.1 Limitation	34
4.2 Access and policy checking	34
4.3 Web page requirements	35
4.4 Main entry point	35
4.5 Start the remote desktop	36
4.6 Stop the remote desktop	37
4.7 Set compression and scaling	38
4.8 Rotate the rendered remote desktop	38
4.9 Refresh the rendering of remote desktop	39
4.10 Send text message to other remote desktop connections to this endpoint	39
4.11 Send Ctrl–Alt–Del to endpoint	39
4.12 Get displays from remote endpoint	39
4.13 Set the target display to render on the canvas element	39
4.14 Expand the KVM canvas to full screen mode	40
4.15 Send clipboard string as keystrokes	40
4.16 Get clipboard text from Internet Explorer	40
4.17 Releasing the Alt-key	40
4.18 Event callbacks	41
4.18.1 onStateChanged event	41
4.18.2 onDebugMessage event	41
4.18.3 onScreenResize event	41
4.18.4 onMessage event	42
4.18.5 onConnectCountChanged event	42
4.18.6 onGetDisplays event	42
5 ema_terminal.js	44
5.1 Limitation	44
5.2 Access and policy checking	44
5.3 Main entry point	45

5.4 Initialize the terminal HTML element	45
5.5 Start the terminal	45
5.6 Stop the terminal	46
5.7 Start the out-of-band Intel® AMT terminal	46
5.8 Event callbacks	47
5.8.1 onStateChanged event	47
5.8.2 onDebugMessage event	47
6 ema_amt.js	48
6.1 Access and policy checking	48
6.2 Main entry point	48
6.3 Perform simple power operations	49
6.4 Perform opt-in operations	50
6.5 Perform reset (or power) to BIOS	52
6.6 Redirect to Intel® AMT web site on the endpoint	53

1 Introduction

Intel® Endpoint Management Assistant (Intel® EMA) is a software application that provides an easy way to manage Intel vPro® platform-based devices in the cloud, both inside and outside the firewall. Intel EMA is designed to make Intel® AMT easy to configure and use so that IT can manage devices equipped with Intel vPro platform technology without disrupting workflow. This in turn simplifies client management and can help reduce management costs for IT organizations.

Intel EMA and its management console offer IT a sophisticated and flexible management solution by providing the ability to remotely and securely connect Intel AMT devices over the cloud. Benefits include:

- Intel EMA can configure and use Intel AMT on Intel vPro platforms for out-of-band, hardware-level management
- Intel EMA can manage systems using its software-based agent, while the OS is running, on non-Intel vPro® platforms or on Intel vPro® platforms where Intel AMT is not activated
- Intel EMA can be installed on premises or in the cloud
- You can use Intel EMA's built-in user interface or call Intel EMA functionality from APIs

1.1 Overview

This document describes how to use various JavaScript libraries to perform different actions for Intel EMA. This is used alongside with Intel EMA REST API. The developer should use Intel EMA REST API first whenever it is possible, and then use these JavaScript libraries for the features not supported by Intel EMA REST API.

Refer to the following documents:

- *Intel® EMA Single Server Installation and Maintenance Guide* for the description of different Intel® EMA servers and the overview of Intel® EMA architecture.
- *Intel® EMA Administration and Usage Guide* for the description of different features and user roles and naming conventions.

1.2 High-level flow of operation

The flow of performing an operation is listed in order:

1. Use the AJAX cookie to connect to Intel EMA AJAX server.
2. Depending on the target operations, the flow varies:
 - For the operation (remote desktop, remote terminal, or remote file browsing) that needs a continuous tunnel between the web browser and the target endpoint, this tunnel (web browser – Intel EMA AJAX server – Intel EMA Swarm server – endpoint) will be established. Then the web browser can send the request and get the response.
 - For Intel® AMT power operations, the web browser is sending the WSMAN command to the target endpoint. Intel EMA servers are simply forwarding the WSMAN command to the target endpoint and forwarding the response back to the web browser.
 - For the operation (remote WMI, remote file search, etc.) that can be sent to multiple endpoints at the same time,

- Intel EMA AJAX server is broadcasting this request to all Intel EMA Swarm servers. Then Intel EMA Swarm servers will send the request to the target endpoints.
- The response is also broadcasted by Intel EMA Swarm server to all Intel EMA AJAX servers, and then the source Intel EMA AJAX server will send the response back to the web browser.
- For the remaining operation (getting endpoint information, uploading files to server, etc.), Intel EMA AJAX server is processing it directly.

1.3 File list

The JavaScript libraries (ema*.js) are automatically installed in the Intel EMA website root folder (e.g., C:\inetpub\wwwroot\.) during Intel EMA server installation. These libraries are not included in the Samples folder with the sample files described below.

The sample files are in the folder [Intel EMA installation package folder] \Samples. These files are not automatically hosted on the Intel EMA website during installation. These sample files are implemented using bare-minimum code to demonstrate how to use the API and do not use secure coding practices to guard against security concerns like cross-site scripting.



IMPORTANT: These samples should *never* be hosted in a production environment.

For hosting in a test environment for development purposes, copy the Samples folder to the Intel EMA website root folder (e.g., C:\inetpub\wwwroot\).

JavaScript libraries:

Name	Comments
ema.js	This is the main library and is always required. This needs to be referenced before other library. This handles the connection to Intel EMA AJAX server and all other operations not covered by the libraries below.
ema_amt.js	This is the library that performs Intel AMT power operations.
ema_desktop.js	This is the library that performs the remote in-band desktop for a single endpoint.
ema_terminal.js	This is the library that performs the remote in-band or out-of-band terminal for a single endpoint.
ema_files.js	This the library that performs the remote file browsing for a single endpoint.

Sample files:

Name	Comments
EndpointAMTOperations.html	This is the sample file for Intel AMT power operations. This uses ema.js and ema_amt.js.
EndpointFileOps.html	This is the sample file for remote file browsing. This uses ema.js and ema_files.js.
EndpointFileSearch.html	This is the sample file for remote file search. This uses ema.js.
EndpointFileShortOps.html	This is the sample file for remote file short operations. This uses ema.js.
EndpointGroupEndpointInfo.html	This is the sample file for getting endpoint (and endpoint group) information. This uses ema.js.
EndpointProcessesOps.html	This is the sample file for getting process list and launching/terminating a

	process. This uses ema.js.
EndpointRDPIInband.html	This is the sample file for remote in-band desktop. This uses ema.js and ema_desktop.js.
EndpointTerminal.html	This the sample file for remote in-band and out-of-band terminal. This uses ema.js and ema_terminal.js.
EndpointWMIQuery.html	This is the sample file for running WMI queries and methods. This uses ema.js.
ServerStatsLive.html	This is the sample file for getting the Swarm server live statistics. This uses ema.js.

1.4 Authentication

To get authenticated and connected to Intel EMA AJAX server, use the stored AJAX cookie generated using Rest API. Intel EMA AJAX server will find the mapped user account for this cookie if this cookie is valid.

For Intel AMT power operations, the Intel AMT credential is stored with encryption in Intel EMA database. To perform an Intel AMT power operation, the flow in order is as follows:

1. The web browser is authenticated and connected to Intel EMA AJAX server.
2. The web browser gets the encrypted routing cookie for the target endpoint. The encrypted routing cookie contains the Intel AMT credential for that endpoint. This routing cookie expires after 60 minutes from the creating time.
3. The web browser sends the power operation request with the routing cookie to Intel EMA AJAX server. Intel EMA AJAX server will parse it send it to the endpoint via Intel EMA Swarm server.

1.5 Authorization

There are two different types of authorization checking.

User access:

- The access control is done by user's primary role and the endpoint group's access list (in the form of user group). Please see the *Intel® EMA Administration and Usage Guide* for details.

Endpoint group's policy set:

- An endpoint is belonging to one and only one endpoint group.
- Each endpoint group has a predefined policy set.
- The policy set defined policies that allow or disallow certain type of operation. Please see the *Intel® EMA Single Server Installation and Maintenance Guide* for details.

2 ema.js

This is the main library and is always required. This needs to be referenced before other library. This handles the connection to Intel® EMA AJAX server and many other operations.

2.1 Main entry point

EMA_SERVER() function:

Input / Output in order	Type	Comments
Output	EMA_SERVER	This is the main object to be used for connecting to Intel EMA AJAX server and for passing to the main entry points of other libraries.

Flow after getting the EMA_SERVER object:

1. (Optional) Use browser (type: string) and browserVer (type: number) members to check the current hosting web browser. Currently, these two members will show correct information for Internet Explorer, Chrome, and Firefox. The developer can use this information to limit the supported web browsers.
2. Set the applicationId (type: string) of that object. This is used by the browser to send application id string so that Intel EMA AJAX server knows what application is connecting.
3. Set the callback to the events that one wants to handle.
4. Start the connection.
5. Then call the target method for the target feature.

Sample files that use this:

- All sample files.

2.2 Starting/stopping the connection to the Intel® EMA AJAX server

Flow after getting the EMA_SERVER object, one should start connecting to Intel EMA AJAX server:

1. Get the AJAX cookie using Rest API.
2. Use connect method with the stored AJAX cookie to connect to Intel EMA AJAX server.
3. When one needs to disconnect from Intel® EMA AJAX server, use disconnect method.

connect method

Input / Output in order	Type	Comments
Input 1	string	This is the AJAX cookie used for connection.
Input 2	string	This is the path to the AJAX server. <ul style="list-style-type: none">• For websocket connection, this is "https://[ema.com]/ajax/,wss://[ema.-com]/ajax/", replacing [ema.com] with the correct server URL.

Input / Output in order	Type	Comments
		<ul style="list-style-type: none"> For normal AJAX connection, this is "https://[ema.com]/ajax/", replacing [ema.com] with the correct server URL.
Input 3	Boolean	"allowWSCrossOrigin". Values True or False (Default). By default (False), if the web socket request's target URL does not match the URL on the web browser, Intel EMA will reject the request before sending. If set to True, Intel EMA will not perform this URL check. Setting this to True can be useful for cases of third-party websites using Intel EMA JavaScript libraries to communicate with the Intel EMA server, where the third-party website and the Intel EMA web server have different URLs.
Output	none	This will trigger onStateChanged event to be fired. The connection result is returned in that event.

disconnect method

Disconnect from Intel EMA AJAX server. This will trigger onStateChanged event to be fired. The connection result is returned in that event.

Sample files that use this:

- All sample files.

2.3 Endpoint group list

Call queryEndpointGroupsList method:

Input / Output in order	Type	Comments
Output	Boolean	<p>True if it is successful.</p> <p>When the result is obtained, the endpointGroups member of EMA_SERVER object will be updated and the onEndpointGroupsListChanged event will be fired.</p> <p>The updated endpointGroups member is based on the user access. If the user does not have View access to any endpoint group, the endpointGroups is empty.</p>

Sample files that use this:

- EndpointGrouptEndpointInfo.html.

2.4 Endpoint list for the tracked endpointGroups

Call queryAllEndpoints method:

Get all the endpoints information for each endpoint group in the endpointGroups.

This uses queryEndpointsList internally. Check section “Endpoint list for the target endpoint groups” for more details.

Input / Output in order	Type	Comments
Output	Boolean	True if it is successful.

Sample files that use this:

- EndpointGroupEndpointInfo.html.

2.5 Endpoint list for the target endpoint groups

Call queryEndpointsList method:

Input / Output in order	Type	Comments
Input	string	This is the list of endpoint group IDs. The string is the combination of endpoint group HEX IDs without any separator. Each endpoint group ID string length is 64. If the string length of endpoints % 64 is not 0, this request will be dropped.
Output	Boolean	True if it is successful. When the result is obtained, the endpoints member of EMA_SERVER object will be updated and the onEndpointsListChanged event will be fired. For the endpoint group in the input that the user does not have View access, Intel® EMA AJAX server's Alert session's AlertCommands.Nodes flow will insert audit log to Intel® EMA database. If the user cannot access any endpoint group in the input, then onEndpointsListChanged event will not be fired.

Sample files that use this:

- queryAllEndpoints used in EndpointGroupEndpointInfo.html is using this method.

2.6 Endpoint information

Call queryEndpoint method:

Input / Output in order	Type	Comments
Input	string	This is the endpoint ID in HEX format without hyphen. If the endpoint ID string length is not 64, this function will fail and return false.
Output	Boolean	True if it is successful. For the endpoint that the user does not have View access (or that does not exist), Intel EMA AJAX server's Alert session's AlertCommands.Node flow will insert audit log to Intel EMA database, and onEndpointChanged will not be fired.

This queries a single endpoint.

- If the target endpoint is not tracked in "endpoints" member of EMA_SERVER object yet, then
 - onEndpointChanged event will be fired with changeType == "added" and with other parameters undefined, and the endpoints member will be added with this new target endpoint.
- If the target endpoint is already tracked in endpoints array, then
 - onEndpointChanged event will be fired with changeType == "updated" and with the endpoint object.

Sample files that use this:

- EndpointGroupEndpointInfo.html.

2.7 Endpoint group or endpoint object from the tracked list

Two tracked lists

EMA_SERVER object has an endpointGroups member, which is the array of ENDPOINT_GROUP object. It is updated when queryEndpointGroupsList is called.

EMA_SERVER object has an endpoints member, which is the array of ENDPOINT object. It is updated when queryAllEndpoints, queryEndpointsList, or queryEndpoint is called.

Call getTrackedEndpointGroupById method:

Get the target ENDPOINT_GROUP object from the endpointGroups array.

Input / Output in order	Type	Comments
Input	string	This is the endpoint ID in HEX format without hyphen.
Output	ENDPOINT_GROUP	ENDPOINT_GROUP object or null if there is no endpoint group associated with the ID in the endpointGroups.

Call getTrackedEndpointById method:

Get the target ENDPOINT object from the endpoints array.

Input / Output in order	Type	Comments
Input	string	This is the endpoint ID in HEX format without hyphen.
Output	ENDPOINT	ENDPOINT object or null if there is no endpoint associated with the ID in the endpoints.

Sample files that use this:

- EndpointGroupEndpointInfo.html.

2.8 Endpoint routing type

Call queryEndpointRouting method:

Input / Output in order	Type	Comments
Input 1	string	This is the endpoint ID in HEX format without hyphen. If the endpoint ID string length is not 64, this function will fail and return false.
Input 2	number	This is the target port to query. For port >= 16992 && port <= 16995, we also check for CIRA tunneling.
Input 3	function	This is the callback when the result is ready. <ul style="list-style-type: none"> Parameter 1 (type: string): This is the endpoint ID in HEX without hyphen. Parameter 2 (type: number): This is the target port to query. Parameter 3 (type: number): This indicates the routing type(s). <ul style="list-style-type: none"> If this is 0, then the user does not have View access or the endpoint does not exist. "Self Routing" type if 1st bit is 1. "Relay Routing" type if 2nd bit is 1. "CIRA" type if 4th bit is 1.
Output	Boolean	True if it is successful. If the user does not have View access to the endpoint (or the endpoint does not exist), Intel® EMA AJAX server's Alert session's AlertCommands.QueryTrafficRouting flow will insert audit log to Intel EMA database, and return flags = 0 on "func" callback.

Sample files that use this:

- EndpointGrouptEndpointInfo.html.

2.9 WMI queries and methods

Call sendWmiQueryToEndpoints method:

This sends a WMI invocation to one or more endpoints.

Input / Output in order	Type	Comments
Input 1	string	These are endpoints' IDs, each in HEX without hyphen. Each ID is directly appended after the other. Each endpoint ID string length is 64. If the string length of endpoints % 64 is not 0, this request will be dropped.
Input 2	number	This is the ID of this request, which can be any number. This is used at the onWmiResponse event callback to associate a WMI response to the original WMI request.
Input 3	number	This is only meaningful for "W" (query). This is the number of starting "rows" to skip reporting for the WMI query result.
Input 4	string	This is W or M appended with the WMI namespace. For example, WROOT\\CIMV2. Use "W" when this is a WMI query. Use "M" when this is a WMI method.

Input / Output in order	Type	Comments
Input 5	string	For "W" (query), this is the query string. For "M", this is the target WMI object.
Input 6	string	This is only defined for "M" (method). This is the WMI method.
Input 7	object	This is only defined for "M" (method). This are the WMI arguments in JSON format. Name is the argument name. Value is the argument value. Value can be of 3 types: string, number, and Boolean.
Output	Boolean	<p>True if it is successful.</p> <p>The result will be at the onWmiResponse event.</p> <p>For policy checking, for the endpoint where MANAGE and TCPRELAY policies are not allowed,</p> <ul style="list-style-type: none"> Intel EMA AJAX server's Alert session's AlertCommands.MultiTargetHopMessage flow will filter it out and not send the request to that endpoint. <p>For user access checking, for the endpoint that the user does not have Execute right,</p> <ul style="list-style-type: none"> Intel EMA AJAX server's Alert session's AlertCommands.MultiTargetHopMessage flow will add audit log to Intel® EMA database, filter it out, and not send the request to that endpoint.

Sample files that use this:

- EndpointProcessesOps.html
- EndpointWMIQuery.html

2.10 Processes list and operations

These are all done via WMI queries.

For enumerate, terminate, and launch, the result is all in onWmiResponse event. We need a way to differentiate them in onWmiResponse event. One can use the request ID input of sendWmiQueryToEndpoints method to associate the onWmiResponse event with the initial request. Please the sample codes for more details.

Sample files that use this:

- EndpointProcessesOps.html

2.11 File search

Current limitation:

- This feature depends on Windows Indexing. It will only find files that are in the "indexed" locations.
- "filter" input only accepts characters from a to z, from A to Z, from 0 to 9, *, and ?. All other characters will be filtered out.
- The maximum returned search result is about 20,000 characters. Any results after the limit will be truncated. In the future, we will support "pagination" type of request and result.

Call sendSearchToEndpoints method:

Send a file search query to a list of endpoints.

Input / Output in order	Type	Comments
Input 1	string	This is an array of string. Each item is the endpoint ID in HEX without hyphen. Each endpoint ID string length is 64. If some endpoints' ID string length is not 64, this request will be dropped.
Input 2	string	This is the search filter. Despite that we only search for file/folder name, the search syntax is similar to Windows search with the wildcard = * or ?. <ul style="list-style-type: none"> Search for ABC and ABC* : It finds "ABC", "ABCDE", "ABC DEF", "XYZ ABC DEF", but it does not find "XYZABC". Search for *ABC* and *ABC: It finds "ABC", "ABCDE", "ABC DEF", "XYZ ABC DEF", and "XYZABC". Search for "ABC": It finds "ABC", "ABC DEF", and "XYZ ABC DEF", but it does not find "ABCDE" and "XYZABC".
Input 3	number	This is the request ID, which can be any number. This is used at the onSearchResponse event callback to associate a response to the original request
Output	Boolean	True if it is successful. The result will be at onSearchResponse event. For policy checking, for the endpoint where FILES and TCPRELAY policies are not allowed, <ul style="list-style-type: none"> Intel EMA AJAX server's Alert session's AlertCommands.MultiTargetHopMessage flow will filter it out and not send the request to that endpoint. For user access checking, for the endpoint that the user does not have Execute right, <ul style="list-style-type: none"> Intel EMA AJAX server's Alert session's AlertCommands.MultiTargetHopMessage flow will add audit log to Intel EMA database, filter it out, and not send the request to that endpoint.

For sending to all endpoints in an endpoint group, call sendSearchToEndpointGroup method:

The result will be at onSearchResponse event. The only difference from sendSearchToEndpoints is the 1st input. For this method, the 1st string input is an endpointGroupId, which is the ID of the target endpoint group in HEX form without hyphen.

Sample files that use this:

- EndpointFileSearch.html

2.12 Swarm server live statistics

Call setSubscriptions method:

Input / Output in order	Type	Comments
Output	Boolean	True if it is successful. The result will be at the onServerStats event. Currently, only Swarm server reports data. If the user is not global administrator, AJAX server's Alert session will add audit log to

Input / Output in order	Type	Comments
		Intel EMA database and the type parameter at onServerStats event will be 3.

Sample files that use this:

- ServerStatsLive.html

2.13 Encrypted routing cookie

Call **getRoutingCookie** method:

This will return the encoded routing cookie that ema_amt.js library can use. This one should not use websocket, but should use the normal AJAX call to connect to AJAX server.

Input / Output in order	Type	Comments
Input 1	string	This is the endpoint ID in HEX format without hyphen. If the endpoint ID string length is not 64, this function will fail and return false.
Input 2	function	This is the "onload" event handler of the XMLHttpRequest object used internally. This event means that this request successfully went to the AJAX server, and AJAX server sent the response back. <ul style="list-style-type: none"> • The 1st parameter is the XMLHttpRequest's responseText. The responseText is in JSON string format. The parsed JSON object has two members: Code and Message. <ul style="list-style-type: none"> • Code is the real result status. See the definitions of EMA_GLOBAL.EMA_GETROUTINGCOOKIE_ERROR object. • Message is the error text. When the Code is EMA_GLOBAL.EMA_GETROUTINGCOOKIE_ERROR.NONE, the Message is the encoded routing cookie. • The 2nd parameter is a string with fixed 'success' value. • The 3rd parameter is the XMLHttpRequest object itself.
Input 3	function	This is the "onerror" and "ontimeout" event handler of the XMLHttpRequest object used internally. The only parameter is the XMLHttpRequest object itself.
Input 4	Boolean	This is to indicate that this is a cookie for in-band operations. Most of the time, this is for out-of-band Intel® AMT operations. In this case, do not need to pass this parameter. In very limited usage (e.g., the redirection to local Intel® AMT web site), we need a cookie for in-band operation and we should pass true.
Output	Boolean	True if it is successful. For endpoint group policy checking, if the target endpoint does not allow TCPRelay policy, <ul style="list-style-type: none"> • EMA_GLOBAL.EMA_GETROUTINGCOOKIE_ERROR.ACTIONFORBIDDEN_POLICY is returned to the "onload" event handler. For the user access checking, if the user does not have Execute right for this endpoint,

Input / Output in order	Type	Comments
		<ul style="list-style-type: none"> AJAX server adds audit log to Intel® EMA DB. EMA_GLOBAL.EMA_GETROUTINGCOOKIE_ERROR.ACTIONFORBIDDEN_USERACCESS is returned to the "onload" event handler.

Sample files that use this:

- EndpointAMTOperations.html

2.14 Power polling rate boosting

Call **boostPowerPollingRate** method:

Boost power polling rate for this endpoint. This is usually used right after the user sends a power state change request. When AJAX server receives this, it forwards to Swarm server. Swarm server will start polling the power state of this endpoint more frequently, until there is no state change for ≥ 2 minutes or the state changes. This relies on Intel® AMT CIRA.

Input / Output in order	Type	Comments
Input 1	string	This is the endpoint ID in HEX format without hyphen. If the endpoint ID string length is not 64, this function will fail and return false.
Output	Boolean	<p>True if it is successful.</p> <p>If the user does not have Execute right for this endpoint, AJAX server adds audit log to Intel EMA database and the action is aborted.</p>

Sample files that use this:

- EndpointAMTOperations.html

2.15 Event callbacks

This subsection is a summary of all the events that will be used in each feature provided by ema.js. The reader can go to each feature description directly and then refer back to this section when needed.

2.15.1 onDebugMessage event

This event will be fired when some debugging information is needed.

Parameter in order	Type	Comments
Parameter	string	This is the debug information.

2.15.2 onStateChanged event

This event will be fired when the connection (to AJAX server) state is changed. This event should always be handled in the user's codes.

Call connect or connectWithPassword method to start receiving this event.

Parameter in order	Type	Comments
Parameter	number	This is the connection state. Check the definitions of EMA_GLOBAL.EMA_CONNECTION object for more details. If one receives EMA_GLOBAL.EMA_CONNECTION.DISCONNECTED before reaching EMA_GLOBAL.EMA_CONNECTION.CONNECTED_LOGIN_COMPLETE, it means that AJAX server is not reachable, or user credential is not valid (if using connectWithPassword method) and AJAX cookie is not valid (if using connect method).

2.15.3 onEndpointsListChanged event

This will be triggered by queryAllEndpoints method or queryEndpointsList method. Furthermore, this will be also triggered by queryEndpoint method.

The callback handler should loop through the “endpoints” of EMA_SERVER object. “endpoints” is an array of ENDPOINT object.

2.15.4 onEndpointGroupsListChanged event

This will be triggered by queryEndpointGroupsList method.

The callback handler should loop through the “endpointGroups” of EMA_SERVER object. “endpointGroups” is an array of ENDPOINT_GROUP object.

2.15.5 onWmiResponse event

This will be triggered by sendWmiQueryToEndpoints method. This is the result of the WMI query or method.

Parameter in order	Type	Comments
Parameter 1	string	This is the target endpoint ID in HEX format without hyphen.
Parameter 2	number	This is the request ID number passed when we sent the WMI request, used at the onWmiResponse event callback to associate a WMI response to the original WMI request.
Parameter 3	number	This is the number of rows returned in WMI_DATA, and is meaningful only for WMI query, not for WMI method. For a failed WMI query or method, this number is the Windows error code.
Parameter 4	WMI_DATA	This is the WMI result. For a failed WMI query or method, this is null.

2.15.6 onSearchResponse event

This will be triggered by sendSearchToEndpoints method and by sendSearchToEndpointGroup method.

Parameter in order	Type	Comments
Parameter 1	string	This is the target endpoint ID in HEX format without hyphen.
Parameter 2	number	This is the request ID number passed when we sent the request, used at the onSearchResponse event callback to associate a response to the original request.
Parameter 3	number	This is the number of rows returned in FILE_SEARCH_DATA.
Parameter 4	FILE_SEARCH_DATA	This is the file search result.

2.15.7 onServerStats event

This will be triggered by setSubscriptions method. After being triggered, this will be fired periodically, until the user is disconnected from the AJAX server. Currently, only Swarm server reports the data.

Parameter in order	Type	Comments
Parameter 1	string	This is the server data in comma-separated values. In order they are: server type (SS is Swarm server), stats version, server Id, milli-seconds since last report, # of connected agents, # of newly connected agents, agent inbound bytes, agent outbound bytes, # of connected consoles, # of newly connected consoles, console inbound bytes, console outbound bytes, # of connected Amt, # of newly connected Amt, Amt inbound bytes, Amt outbound bytes, used memory. If type = 3, then do not check data.
Parameter 2	number	This the type of the data. (0 = Live, 1 = Last Passed Data, 2 = Passed Data, 3 = Error due to user without access.)

2.16 JSON object definitions

This subsection is a summary of all the JSON objects that will be used in each feature provided by ema.js. The reader can go to each feature description directly and then refer back to this section when needed.

2.16.1 ENDPOINT_GROUP object

Name	Type	Comments
meshid	string	This is the ID of the endpoint group in HEX format without hyphen.
rights	string	This indicates the allowed action that this user has. See EMA_GLOBAL.EMA_USER_ALLOWED_ACTION object definitions for more details.
serial	string	This is the serial number of the policy file for this endpoint group. Usually, this is 1.

Name	Type	Comments
name	string	This is the endpoint group name.
nodecount	string	This is the number of endpoints in this endpoint group.

2.16.2 ENDPOINT_INTERFACE object

Name	Type	Comments
IPv4	string	This is IPv4 address.
IPv6	string	This is IPv6 address.
Fqdn	string	This is FQDN.
Subnet	string	This is the subnet.
Mac	string	This is the MAC address in HEX format.
Gateway	string	This is the gateway address.
GatewayMac	string	This is the gateway MAC address in HEX format.

2.16.3 ENDPOINT_AMT object

Name	Type	Comments
version	string	This is the Intel® AMT version, e.g., 9.5.61.
mode	number	This is the provision mode. Check the definition of EMA_GLOBAL.EMA_AGENT_AMT_PROVISIONING_MODE object.
state	number	This is the provisioning state. Check the definition of EMA_GLOBAL.EMA_AGENT_AMT_PROVISIONING_STATE object.
tls	Boolean	This indicates using TLS or not.

2.16.4 ENDPOINT object

Name	Type	Comments
meshid	string	This is the ID of the endpoint group in HEX format without hyphen.
nodeid	string	This is the ID of the endpoint in HEX format without hyphen.
name	string	This is the endpoint name.
desc	string	This is the endpoint's Operating System.
serial	number	This is the index of endpoints based on the order of endpoint registration to Intel EMA.
agentid	number	This is the Intel EMA agent type. Check the definition of EMA_

Name	Type	Comments
		GLOBAL.EMA_AGENT_TYPE object.
agentversion	number	This is the Intel EMA agent's release version number.
type	number	This does not mean anything. This is only used to select an icon file for this endpoint. The user can set the type and hence change the icon file used.
powerstate	number	This is the endpoint's last-known power state. Check the definition of EMA_GLOBAL.EMA_AGENT_POWER_STATE object.
interfaces	ENDPOINT_INTERFACE[]	This is the list of network interfaces, an array of ENDPOINT_INTERFACE objects. The information may vary depending on the availability.
amt	ENDPOINT_AMT	This is the Intel EMA agent's Intel® AMT information. The information may vary depending on the availability.

2.16.5 WMI_DATA object

The overall data is in a table format with the number of columns and the number of rows. The 1st row defines all the captions/names of the table.

Name	Type	Comments
rdata	string	This is the whole data content of WMI_DATA.
totalLength	number	This is the byte size of the whole WMI response data, not just WMI_DATA.
count	number	This is the total number of entries in the WMI result. This should be column count * row count.
colCount	number	This is the number of columns for the WMI result.
rowCount	number	This is the number of rows for the WMI result.
valType	number	This is the data type of this table cell. Check the definition of EMA_GLOBAL.EMA_WMI_DBTYPE object.
valTypeStr	string	This is the data type of this table cell based on valType.
valLen	string	This is the byte size for this table cell.
valRaw	string	This is the raw data for this table cell.
val	string number boolean	This is the translated value for this table cell, based on valTypeStr and valRaw.
getNext	function	Use this method to get the next table cell in the WMI result. This needs to be called first to get the first cell. When it returns false, then it does not have any more cell.

At the onWmiResponse event, WMI_DATA is passed. Then the flow is

1. Use getNext method to move to the next table cell entry.
2. Then read the valXXX members which store the value of the cell entry. One can use colCount member to determine when the result is on a new row.
3. Go back to step 1 and repeat, until getNext returns false.

2.16.6 FILE_SEARCH_DATA object

Name	Type	Comments
data	string	This is the data content of this object.
totalLength	number	This is the length of data.
fileName	string	This is the file name of the current item in the results.
filePath	string	This is the file path of the current item in the results.
fileSize	number	This is the file size of the current item in the results. If this is 0, then this entry should be a folder.
getNext	function	Use this method to get the next item in the search result. This needs to be called first to get the first item. When it returns false, then it does not have any more item.

At the onSearchResponse event, FILE_SEARCH_DATA is passed. Then the flow is

1. Use getNext method to move to the next entry.
2. Then read the fileName, filePath, and fileSize members for the entry.
3. Go back to step 1 and repeat, until getNext returns false.

2.16.7 FILE_OPERATOR object

Name	Type	Comments
readFile	function	This is to enter operation information for reading the remote file. The only string parameter is the file path on the target endpoint. For example, C:\\codes\\settings.txt or C:\codes\settings.txt.
writeFile	function	This is to enter operation information for creating a new remote file (or replacing the old one with the new file). The first string parameter is the file path on the target endpoint. For example, C:\\codes\\settings.txt or C:\codes\settings.txt. The second string parameter is the data to write to the file. We only take the first 60KB data.
appendFile	function	This is to enter operation information for appending the remote file with the new data. It creates a new file if the file does not exist. The first string parameter is the file path on the target endpoint. For example, C:\\codes\\settings.txt or C:\codes\settings.txt. The second string parameter is the data to append to the file. We only take the first 60KB data.

Name	Type	Comments
deleteFile	function	This is to enter operation information for deleting the remote file. The only string parameter is the file path on the target endpoint. For example, C:\\codes\\settings.txt or C:\\codes\\settings.txt.
checkDir	function	This is to enter operations information for checking if the folder exists. The only string parameter is the directory to check. For example, C:\\codes or C:\\codes.
execute	function	This is to really send out the request. For example, to do a "read file", one run readFile and then run execute.
getNextResult	function	This is used at the operation result callback to go through all results. The result is a FILE_OPERATOR_RESULT object. To get all the results in onShortFileOperation result callback, use a while loop until getNextResult returns null.

2.16.8 FILE_OPERATOR_RESULT object

Name	Type	Comments
operation	string	This is the target operation in the original request.
file	string	This is the target file or directory in the original request.
result	number	This is the result status of the request. For read (readFile), if it is successful, the result is the same as the datalen. If the file to read does not exist or it is not a file, the result and datalen are 0. For delete (deleteFile), if it is successful, the result is 0. If the result is not 0, the request failed (e.g., due to file not existing). For checkdir (checkDir), if the folder exists, the result is 1. Otherwise, it is 0. For write or append (writeFile or appendFile), if it is successful, the result is 1. Otherwise, it is 0.
datalen	number	This is the data length of the result data. This is only meaningful for readFile.
data	string	This is the data of the result. This is only meaningful for readFile.

2.17 Enumeration object definitions

This subsection is a summary of all the objects used as enumeration. The user can skip this subsection and go to each feature directly. Then refer back to this subsection later.

All of them are defined in EMA_GLOBAL object.

2.17.1 AJAX server connection state

```
EMA_GLOBAL.EMA_CONNECTION: {  
    DISCONNECTED: 0,  
    CONNECTING: 1,  
    CONNECTED_LOGIN_COMPLETE: 2,  
    ERROR: 3  
}
```

2.17.2 Swarm server tunnel connection state

```
EMA_GLOBAL.EMA_TUNNEL: {  
    DISCONNECTED: 0,  
    CONNECTING: 1,  
    GOT_TUNNEL: 2,  
    CONNECTED: 3  
}
```

2.17.3 Swarm server tunnel connection's error code

```
EMA_GLOBAL.EMA_TUNNEL_ERRORCODE: {  
    NONE: -1,  
    UNKNOWN: 0,  
    NOLEADERFOUND: 1,  
    TARGETNOTFOUND: 2,  
    TLS_FAILED: 3,  
    REMOTE_DISCONNECT: 4,  
    EXCEPTION: 5,  
    LEADER_DISCONNECT: 6,  
    CONSOLE_DISCONNECT: 7,  
    BADCOMMAND: 8,  
    AUTH_FAILED: 9,  
    AUTH_REQUIRED: 10,  
    FORCE_DISCONNECT: 11,  
    DUPLICATE_AGENT: 12,  
    MAX_SOCKETS_REACHED: 13,  
    DATA_OVERFLOW: 14,  
    AUTO_CLOSE: 15,  
    ACCESS_DENIED: 16,
```

```

    HTTPREQUEST: 17,
    LICENSELIMIT: 18,
    DIRECTCONNECTFAIL: 19,
    UNKNOWNCOMMAND: 20,
    TIMEOUT: 21,
    POLICYNOTALLOWED: 22,
    UNSUPPORTED: 23
}

```

2.17.4 AJAX server connection mode

```

EMA_GLOBAL.EMA_CONNECTMODE: {
    HTTP: 0,
    WEBSOCKET: 1,
    WEBRTC: 2 // currently not supported anymore
}

```

2.17.5 User's allowed action on endpoint group

```

EMA_GLOBAL.EMA_USER_ALLOWED_ACTION: {
    NONE: 0,
    READ: 1,
    CREATE: 2,
    UPDATE: 3,
    DELETE: 4
}

```

2.17.6 File browsing object type

```

EMA_GLOBAL.EMA_FILES_CONTENT_TYPE: {
    DRIVE: 1,
    FOLDER: 2,
    FILE: 3
}

```

2.17.7 File upload state

```

EMA_GLOBAL.EMA_FILE_UPLOAD_STATE: {
    DONE: 0,
    INPROGRESS: 1,
}

```

```
        ERROR: 2
    }
```

2.17.8 File upload error for scheduled task

```
EMA_GLOBAL.EMA_SCHEDULED_FILE_UPLOAD_ERROR: {
    NO_AJAXCOOKIE: 1,
    NO_TASKID_INPUT: 2,
    CANNOT_GET_FILE_STORAGE_PATH: 3,
    CANNOT_ACCESS_STORAGE_FOLDER: 4,
    INVALID_AJAXCOOKIE: 5,
    ACTIONFORBIDDEN_USERACCESS: 6,
    INVALID_OFFSET_INPUT: 7,
    INVALID_LASTCHUNK_INPUT: 8,
    NO_FILENAME_INPUT: 9,
    CANNOT_PARSE_FILENAME_INPUT_TO_EXCLUDE_PATH: 10,
    CANNOT_UPLOAD_FILE: 11,
    MISMATCH_FINAL_SIZE: 12,
    MISMATCH_FINAL_HASH: 13,
    CANNOT_REMOVE_TASK_FOLDER: 14
}
```

2.17.9 File upload error for metadata

```
EMA_GLOBAL.EMA_METADATA_FILE_UPLOAD_ERROR: {
    NO_AJAXCOOKIE: 1,
    NO_PARAMS_INPUT: 2,
    INVALID_OR_MISSING_ENDPOINTGROUP_INPUT: 3,
    CANNOT_GET_ENDPOINTGROUP_NAME: 4,
    INVALID_AJAXCOOKIE: 5,
    ACTIONFORBIDDEN_USERACCESS: 6,
    INVALID_OFFSET_INPUT: 7,
    INVALID_LASTCHUNK_INPUT: 8,
    NO_FILENAME_INPUT: 9,
    CANNOT_PARSE_FILENAME_INPUT_TO_EXCLUDE_PATH: 10,
    INVALID_JSON_FORMAT: 11,
    CANNOT_WRITE_RESPONSE: 12,
    GENERAL_ERROR: 13
}
```

2.17.10 Intel® EMA agent types

```
EMA_GLOBAL.EMA_AGENT_TYPE: {  
  
    GENERIC: 0,  
    WIN32_CONSOL: 1,  
    WIN64_CONSOLE: 2,  
    WIN32_SERVICE: 3,  
    WIN64_SERVICE: 4,  
    LINUX_X86: 5,  
    LINUX_X86_64: 6,  
    LINUX_MIPS: 7,  
    LINUX_X86_XEN_ESXI: 8,  
    ANDROID_ARM: 9  
  
}
```

2.17.11 Endpoint power state

```
EMA_GLOBAL.EMA_AGENT_POWER_STATE: {  
  
    POWERED_0: 0,  
    POWERED_1: 1,  
    POWER_OFF_0: 2,  
    POWER_OFF_1: 3,  
    SLEEP: 4,  
    HIBERNATING: 5,  
    POWERED_2: 6,  
    PRESENT: 7  
  
}
```

2.17.12 WMI result data type

```
EMA_GLOBAL.EMA_WMI_DBTYPE: {  
  
    EMPTY: 0,  
    NULL: 1,  
    I2: 2,  
    I4: 3,  
    R4: 4,  
    R8: 5,  
    CY: 6,  
    DATE: 7,  
  
}
```

```

    BSTR: 8,
    IDISPATCH: 9,
    ERROR: 10,
    BOOL: 11,
    VARIANT: 12,
    IUNKNOWN: 13,
    DECIMAL: 14,
    I1: 16,
    UI1: 17,
    UI2: 18,
    UI4: 19,
    I8: 20,
    UI8: 21,
    GUID: 72,
    VECTOR: 0x1000,
    ARRAY: 0x2000,
    BYREF: 0x4000,
    RESERVED: 0x8000,
    BYTES: 128,
    STR: 129,
   WSTR: 130,
    NUMERIC: 131,
    UDT: 132,
    DBDATE: 133,
    DBTIME: 134,
    DBTIMESTAMP: 135
}

```

2.17.13 Intel® AMT provision state

```

EMA_GLOBAL.EMA_AGENT_AMT_PROVISIONING_STATE: {
    PRE_PROVISIONING: 0,
    IN_PROVISIONING: 1,
    PROVISIONED: 2
}

```

2.17.14 Intel® AMT provision mode

```

EMA_GLOBAL.EMA_AGENT_AMT_PROVISIONING_MODE: {

```

```

    NONE: 0,
    ENTERPRISE: 1,
    SMALL_BUSINESS: 2,
    REMOTE_ASSIST: 3
}

```

2.17.15 Intel® AMT AJAX path error type

```

EMA_GLOBAL.EMA_GETROUTINGCOOKIE_ERROR: {
    NONE: 0,
    NO_AJAXCOOKIE: 1,
    INVALID_AJAXCOOKIE: 2,
    NO_ENDPOINTID: 3,
    NO_JSONINPUT: 4,
    NO_ENDPOINTINFO: 5,
    ACTIONFORBIDDEN_POLICY: 6,
    ACTIONFORBIDDEN_USERACCESS: 7,
    NO_USERNAME: 8,
    NO_ENCRYPTEDCOOKIE: 9,
    CANNOT_WRITE_RESPONSE: 10
}

```

2.17.16 Intel® AMT WSMAN error type

```

EMA_GLOBAL.EMA_AMTWSMAN_ERROR: {
    NO_AJAXCOOKIE: 1,
    INVALID_AJAXCOOKIE: 2,
    UNKOWN_USER: 3,
    INVALID_ENDPOINT: 4,
    ACTIONFORBIDDEN_USERACCESS: 5,
    NO_ROUTE: 6,
    NO_ENDPOINT_INFO: 7,
    ACTIONFORBIDDEN_POLICY: 8,
    CANNOT_WRITE_RESPONSE: 10
}

```

2.17.17 Remote desktop related enumeration

```

EMA_GLOBAL.EMA_DESKTOP_STATE: {

```

```

    DISCONNECTED: 0,
    CONNECTING: 1,
    PAUSED: 2,
    UNPAUSED: 3
  }
  EMA_GLOBAL.EMA_DESKTOP_TOUCH: {
    DISABLED: 0,
    ENABLED: 1
  }
  EMA_GLOBAL.EMA_DESKTOP_TUNNEL: {
    DISCONNECTED: 0,
    CONNECTED: 1
  }
}

```

2.17.18 Remoter terminal related enumeration

```

EMA_GLOBAL.EMA_TERMINAL_STATE: {
  DISCONNECTED: 0,
  CONNECTING: 1,
  CONNECTED: 2
}
EMA_GLOBAL.EMA_TERMINAL_TUNNEL: {
  DISCONNECTED: 0,
  CONNECTED: 1
}

```

3 ema_files.js

This library is used together with ema.js. It provides the following features: file/folder browsing, files upload, file download, file/folder removal, file/folder renaming, and folder creation.

Since the endpoint ID is passed at the object constructor, remember to create a new EMA_SERVER_REMOTE_FILES object when you have a different endpoint.

3.1 Access and policy checking

Policy checking is done based on the following order:

1. When the front-end tries to get the best Swarm server for the target endpoint, AJAX server's Alert session checks if the target endpoint group has TCPRELAY policy.
 - If it fails, AJAX server sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.POLICYNOTALLOWED) back and disconnects the tunnel connection.
2. For remote (in-band, out-of-band) terminal, remote file, and remote RDP operations, when the initial setup message is sent to the target endpoint (via Intel® EMA servers), the endpoint agent checks Command, FILES, and KVM policies.
 - If they are all disabled, endpoint will reject this operation and AJAX server sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.REMOTE_DISCONNECT) back and disconnects the tunnel connection.
3. When the starting "file prompt" message is sent to AJAX server, AJAX server's Alert session checks if the target endpoint group has FILES policy.
 - If it fails, AJAX server sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.POLICYNOTALLOWED) back and disconnects the tunnel connection.

User access checking:

- When the front-end tries to get the best Swarm server for the target endpoint, AJAX server's Alert session checks the following
 - If the user does not have Execute right for the endpoint, AJAX server adds audit log, sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.ACCESSDENIED) back, and disconnects the tunnel connection.
 - If the endpoint ID has correct format but does not exist, AJAX server adds audit log to the Intel® EMA database, sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.ACCESSDENIED) back, and disconnects the tunnel connection.
 - If Intel® EMA server cannot get the correct information for this endpoint, AJAX server adds audit log to the Intel® EMA database, sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.TARGETNOTFOUND) back, and disconnects the tunnel connection.
 - If AJAX server cannot find the Swarm server for this endpoint, AJAX server sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.NOLEADERFOUND) back, and disconnects the tunnel connection.

3.2 Main entry point

EMA_SERVER_REMOTE_FILES function:

Input / Output in order	Type	Comments
Input 1	EMA_SERVER	This is the object returned by the main entry point of ema.js. When we call EMA_SERVER_REMOTE_FILES function, the connection to AJAX server (performed by ema.js) should be established already.
Input 2	string	The value is the target endpoint ID, in HEX format (without hyphen).
Output	EMA_SERVER_REMOTE_FILES	This is the main object to be used for files/folders operations.

Sample files that use this:

- EndpointFileOps.html

3.3 Starting the tunnel connection to endpoint via Swarm server

Call start method:

This should be called before calling other file operation methods.

Input / Output in order	Type	Comments
Output	none	If the method is successful, it returns true (type: Boolean); otherwise, it returns false (type: Boolean). This will trigger firing onStateChanged event. If the method works, start monitoring the onStateChanged event.

Sample files that use this:

- EndpointFileOps.html

3.4 Files browsing

Call the requestDirectoryPath(path) method:

Input / Output in order	Type	Comments
Input 1	string	For Windows endpoint, it needs to have *.* to get all the folder contents. For example, C:\Codes*. or C:\Codes*. This can also be a file. The file information will be returned.
Output	none	The result event onFilesResponse will be fired when it is ready. If the showHidden member is true before this method is called, then the hidden files will also be returned. If the directory does not exist, it will return the list of drives for this endpoint.

Sample files that use this:

- EndpointFileOps.html

3.5 Files uploading

Call **sendFiles** method:

Wait till the current files uploading request is done before sending a new files uploading request.

Input / Output in order	Type	Comments
Input 1	FileList	This is the standard JavaScript FileList object, returned by the files property of the HTML <input> element.
Input 2	string	This is the target path on the endpoint. For Windows endpoint, this needs to end with path separator, e.g., C:\\temp\\ or C:\\temp\\.
Output	none	The result will be at onUploadState event. If the target path already has a file with the same name, that original file will be replaced. If the target path does not exist, onUploadState does not return an error. The file upload will continue until it is all transmitted; however, the file will not be generated.

Sample files that use this:

- EndpointFileOps.html

3.6 Files upload cancellation

Call **cancelSendFiles** method:

This will trigger onUploadState event to report EMA_GLOBAL.EMA_FILE_UPLOAD_STATE.DONE state.

Sample files that use this:

- EndpointFileOps.html

3.7 File downloading

Call **fileDownloadStart** method:

This method only handles a single file download. One should wait till the current downloading is done, before sending a new downloading request.

Files of size 0 bytes are not supported.

Input / Output in order	Type	Comments
Input 1	string	This is the target file to download.

Input / Output in order	Type	Comments
Input 2	function	This is the callback when the downloading response is sent back. It is called onFileDownload.
Output	number	This is the ID of this current downloading request. This can be used to cancel this downloading.

The callback function onFileDownload:

Parameter in order	Type	Comments
Parameter 1	EMA_SERVER_REMOTE_FILES	This is the EMA_SERVER_REMOTE_FILES object.
Parameter 2	string	This is the target file, including path.
Parameter 3	number	This is the current download position in bytes. Based on position and size, one can calculate the downloading percentage.
Parameter 4	number	This is the total file size in bytes. If this is 0, then there is no file to be downloaded. When position == size, the download is done.
Parameter 5	string	This is the data in transit in the format of string. This is the aggregated data downloaded so far. When the download is done, this data can be saved into a file. Check the sample codes for example.

Sample files that use this:

- EndpointFileOps.html

3.8 File downloading cancellation

Call cancelFileDownload method:

Input / Output in order	Type	Comments
Input	number	This is the ID of this current downloading request. This is returned by fileDownloadStart method.

Sample files that use this:

- EndpointFileOps.html

3.9 File/folder moving/renaming

Call fileMove method:

Input / Output in order	Type	Comments
Input 1	string	This is the target file or folder.
Input 2	string	This is the new file or folder.
Input 3	function	This is the callback function when the moving/renaming response is sent back. It is called onFileMove.

The callback function onFileMove:

Parameter in order	Type	Comments
Parameter 1	EMA_SERVER_REMOTE_FILES	This is the EMA_SERVER_REMOTE_FILES object.
Parameter 2	string	This is the old file or folder.
Parameter 3	string	This is the new file or folder.
Parameter 4	number	This is return code. This is 1 if it succeeded, and 0 if it failed.

Sample files that use this:

- EndpointFileOps.html

3.10 File/folder removal

Call fileDelete method:

Input / Output in order	Type	Comments
Input 1	string	This is the target file or the target folder to be deleted.
Input 2	Boolean	This is meaningful if the target is a folder. This indicates if we should do recursive removal for folder content.
Input 3	function	This is the callback function when the removal response is sent back. It is called onFileDelete.

The callback function onFileDelete:

Parameter in order	Type	Comments
Parameter 1	EMA_SERVER_REMOTE_FILES	This is the EMA_SERVER_REMOTE_FILES object.
Parameter 2	string	This is the target file or the target folder to be deleted.
Parameter 3	Boolean	This is meaningful if the target is a folder. This indicates if we should do recursive removal for folder content.
Parameter 4	number	This is return code. This is 1 if it succeeded, and 0 if it failed. If the

Parameter in order	Type	Comments
		target path is a folder with content, fileDelete() will fail without doing recursive removal.

Sample files that use this:

- EndpointFileOps.html

3.11 Folder creation

Call createDirectory method:

Input / Output in order	Type	Comments
Input 1	string	This is the target folder to create.
Input 2	function	This is the callback function when the creating response is sent back. It is called onFileCreateDir.

The callback function onFileCreateDir:

Parameter in order	Type	Comments
Parameter 1	EMA_SERVER_REMOTE_FILES	This is the EMA_SERVER_REMOTE_FILES object.
Parameter 2	string	This is the target folder to create.
Parameter 3	number	This is return code. This is 1 if it succeeded, and 0 if it failed.

Sample files that use this:

- EndpointFileOps.html

3.12 Event callbacks

This subsection is a summary of all the events that will be used in each feature provided by this library. The reader can go to each feature description directly and then refer back to this section when needed

3.12.1 onDebugMessage event

This event will be fired when some debugging information is needed.

Parameter in order	Type	Comments
Parameter	string	This is the debug information.

3.12.2 onStateChanged event

This event will be fired when the connection (to endpoint via Swarm server) state changes.

Parameter in order	Type	Comments
Parameter 1	EMA_SERVER_REMOTE_FILES	This is the event source.
Parameter 2	number	This is the tunnel (web page via Swarm server to endpoint) connection state. Check EMA_GLOBAL.EMA_TUNNEL object on ema.js for details.

If you get EMA_GLOBAL.EMA_TUNNEL.DISCONNECTED before EMA_GLOBAL.EMA_TUNNEL.CONNECTED, then we cannot get the tunnel to the endpoint correctly.

When the new state is EMA_GLOBAL.EMA_TUNNEL.DISCONNECTED, you should check if tunnelErrorCode member > EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.NONE for tunnel failure. For the error value mapping, check EMA_GLOBAL.EMA_TUNNEL_ERRORCODE object definition of ema.js for details.

3.12.3 onFilesResponse event

This event will be fired when we get the result of the files browsing operation.

Parameter in order	Type	Comments
Parameter 1	EMA_SERVER_REMOTE_FILES	This is the event source.
Parameter 2	FILE_ITEM[]	This is the array FILE_ITEM objects.

3.12.4 onUploadState event

This event will be fired when we get the progress result of the files uploading operation.

Parameter in order	Type	Comments
Parameter 1	EMA_SERVER_REMOTE_FILES	This is the event source.
Parameter 2	number	This is the file upload state. It can be EMA_GLOBAL.EMA_FILE_UPLOAD_STATE.DONE, EMA_GLOBAL.EMA_FILE_UPLOAD_STATE.INPROGRESS, or EMA_GLOBAL.EMA_FILE_UPLOAD_STATE.ERROR.
Parameter 3	string	This is the file name. This is null if state == EMA_GLOBAL.EMA_FILE_UPLOAD_STATE.DONE or EMA_GLOBAL.EMA_FILE_UPLOAD_STATE.ERROR.
Parameter 4	number	This is the total file size of the current file in bytes. This is 0 if state == EMA_GLOBAL.EMA_FILE_UPLOAD_STATE.DONE

Parameter in order	Type	Comments
		or EMA_GLOBAL.EMA_FILE_UPLOAD_STATE.ERROR.
Parameter 5	number	This is the current upload position of this file in bytes. This is 0 if state == EMA_GLOBAL.EMA_FILE_UPLOAD_STATE.DONE or EMA_GLOBAL.EMA_FILE_UPLOAD_STATE.ERROR.
Parameter 6	number	This is the current in-progress file index among the files list. This is only defined if state == EMA_GLOBAL.EMA_FILE_UPLOAD_STATE.INPROGRESS.
Parameter 7	number	This is the total files count in the files list. This is only defined if state == EMA_GLOBAL.EMA_FILE_UPLOAD_STATE.INPROGRESS.

3.13 JSON object definitions

This subsection is a summary of all the JSON objects that will be used in each feature provided by this library. The reader can go to each feature description directly and then refer back to this section when needed.

3.13.1 FILE_ITEM object

Name	Type	Comments
name	string	This is the file/folder/drive name.
date	Date	This is the last modified date, JavaScript Date object.
size	number	This is the low order bits of the overall size (first 32 bits). (It is 0 for non-file type.)
sizeh	number	This is the high order bits of the overall size (second 32 bits). (It is 0 for non-file type.)
type	number	This is the type. See the definition of EMA_GLOBAL.EMA_FILES_CONTENT_TYPE object.

4 ema_desktop.js

This library is used together with ema.js. This library will perform in-band remote desktop connection to the target endpoint.

Since the endpoint ID is passed at the object constructor, remember to create a new EMA_SERVER_REMOTE_DESKTOP object when you have a different endpoint.

4.1 Limitation

- When the target endpoint display's DPI value changes, the current user on the endpoint needs to sign out and sign in again so that Intel® EMA can get current display resolution. This limitation is shared by many Windows applications. In the latest Windows 7, when you change the display resolution, Windows auto changes the DPI value. In this case, this limitation applies.
- Windows 7 does not support different DPI values for multiple displays. In order to support Windows 7, if the endpoint has multiple displays, all displays need to have the same DPI as the primary display for Intel EMA to get current resolution for other displays.

4.2 Access and policy checking

Policy checking is done based on the following order:

1. When the front-end tries to get the best Swarm server for the target endpoint, AJAX server's Alert session checks if the target endpoint group has TCPRELAY policy.
 - If it fails, AJAX server sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.POLICYNOTALLOWED) back and disconnects the tunnel connection.
2. For remote (in-band, out-of-band) terminal, remote file, and remote RDP operations, when the initial setup message is sent to the target endpoint (via Intel® EMA servers), the endpoint agent checks Command, FILES, and KVM policies.
 - If they are all disabled, endpoint will reject this operation and AJAX server sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.REMOTE_DISCONNECT) back and disconnects the tunnel connection.
3. When the starting "RDP prompt" message is sent to AJAX server, AJAX server's Alert session checks if the target endpoint group has KVM policy.
 - If it fails, AJAX server sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.POLICYNOTALLOWED) back and disconnects the tunnel connection.

User access checking:

- When the front-end tries to get the best Swarm server for the target endpoint, AJAX server's Alert session checks the following
 - If the user does not have Execute right for the endpoint, AJAX server adds audit log to the Intel® EMA database, sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.ACCESSDENIED) back, and disconnects the tunnel connection.
 - If the endpoint Id has correct format but does not exist, AJAX server adds audit log to the Intel® EMA database, sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.ACCESSDENIED) back, and disconnects the tunnel connection.

- If the Intel EMA server cannot get the correct information for this endpoint, AJAX server adds audit log to Intel EMA database, sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.TARGETNOTFOUND) back, and disconnects the tunnel connection.
- If AJAX server cannot find the Swarm server for this endpoint, AJAX server sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.NOLEADERFOUND) back, and disconnects the tunnel connection.

4.3 Web page requirements

Required element:

- A <canvas> element.

Required canvas style set-up:

Need a function to set up <canvas> element's width, max-width, and height styles.

When EMA_SERVER_REMOTE_DESKTOP is connected (by checking EMA_SERVER_REMOTE_DESKTOP object's state), these style values should be based on EMA_SERVER_REMOTE_DESKTOP object's targetScreenWidth and targetScreenHeight.

If full screen mode is supported, call EMA_SERVER_REMOTE_DESKTOP object's isCanvasInFullScreenMode function to check if the canvas is in full screen mode or not. If it is in the full screen mode, call EMA_SERVER_REMOTE_DESKTOP object's getTargetCanvasRenderingSizeInFullScreenMode function to get the width and height, instead of using EMA_SERVER_REMOTE_DESKTOP object's targetScreenWidth and targetScreenHeight.

This canvas style set-up function will be called at:

- Web page start up.
- Web browser's window.onresize event callback.
- Main EMA_SERVER object's onStateChanged event callback.
- EMA_SERVER_REMOTE_DESKTOP object's onStateChanged event callback.
- EMA_SERVER_REMOTE_DESKTOP object's onScreenResize event callback.

Required HTML event handler:

- window.onresize: At this event, run the canvas style set-up function.

Sample files that use this:

- EndpointRDPIband.html

4.4 Main entry point

EMA_SERVER_REMOTE_DESKTOP function:

Input / Output in order	Type	Comments
Input 1	EMA_SERVER	This is the object returned by the main entry point of ema.js. The connection to AJAX server (performed by ema.js) should be established already.
Input 2	string	The value is the target endpoint ID, in HEX format (without hyphen).
Input 3	string	This is the HTML id of the canvas element on the HTML page that is hosting this remote desktop.

Input / Output in order	Type	Comments
Output	EMA_SERVER_REMOTE_DESKTOP	This is the main object to be used for connecting to Intel EMA tunnels and for controlling the remote desktop.

The high-level operations flow after the object creation is:

1. Create EMA_SERVER object and connect to AJAX server.
2. Create EMA_SERVER_REMOTE_DESKTOP object.
3. Assign the event callbacks for EMA_SERVER_REMOTE_DESKTOP object.
4. Set EMA_SERVER_REMOTE_DESKTOP object's compression and scaling.
5. Grab the mouse and touch events from HTML canvas, and keyboard events from HTML document.
6. Use start method to start the tunnel connection to endpoint agent via Swarm server.
7. At the onStateChanged event callback, if the event is UNPAUSED, after some delay, use sendCompressionAndScaling function to send the settings to the endpoint.

The high-level operations flow for disconnect from terminal is:

1. Use stop method.
2. Un-grab the mouse and touch events from HTML canvas, and keyboard events from HTML document.
3. When the EMA_SERVER object is disconnected from AJAX server, this disconnect flow should be run also.

Sample files that use this:

- EndpointRDPInband.html.

4.5 Start the remote desktop

Check tunnelMode first. If it is not EMA_GLOBALEMA_DESKTOP_TUNNEL.DISCONNECTED, then do not continue.

Call setCompression method:

This sets the percentage of the screen bitmap's compression level on the target endpoint's display. This does not send the new configuration to the target endpoint yet.

Input / Output in order	Type	Comments
Input	number	This is [0, 0.75]. 0.5 or less is best for speed concern. 1 means that there is no compression.
Output	Boolean	True if it is successful.

Call setScaling method:

This sets the percentage of the render resolution reduction on the target endpoint's display. This does not send the new configuration to the target endpoint.

Input / Output in order	Type	Comments
Input	number	This is [0, 1]. 1 means that there is no reduction. 0.5 and 1.0 result in best

Input / Output in order	Type	Comments
		performance. If you use other value, the display area may be cut off a bit.
Output	Boolean	True if it is successful.

Call grabMouseInput method:

Intercept the mouse and touch inputs of the HTML canvas element. This should be called before calling start method, and also when we need to intercept the inputs again.

Call grabKeyInput method:

Intercept the keyboard inputs of the HTML document. This should be called before calling start method, and also when we need to intercept the inputs again.

Call start method:

Start the tunnel connection to endpoint agent via Swarm server, and start the flow of remote desktop.

Input / Output in order	Type	Comments
Output	Boolean	True if it is successful. If the return is false (type: Boolean), this method fails, and remember to un-grab the mouse, touch, and key inputs. If the method works, monitor the onStateChanged event.

Sample files that use this:

- EndpointRDPIInband.html

4.6 Stop the remote desktop

This flow should also be run when the main EMA_SERVER object disconnects from the AJAX server.

Call stop method:

Un-grab the input events from the HTML, and stop the tunnel connection to endpoint agent via Swarm server.

Call unGrabMouseInput method:

Stop intercepting the mouse and touch inputs of the HTML canvas element. This should be called when onStateChanged fires with EMA_GLOBAL.EMA_DESKTOP_STATE.DISCONNECTED state, and also when we need to yield the inputs back to HTML again.

Call unGrabKeyInput method:

Stop intercepting the keyboard inputs of the HTML document. This should be called when onStateChanged fires with EMA_GLOBAL.EMA_DESKTOP_STATE.DISCONNECTED state, and also when we need to yield the inputs back to HTML again.

Sample files that use this:

- EndpointRDPIInband.html

4.7 Set compression and scaling

Call sendCompressionAndScaling method:

This sets the bitmap compression level and the desktop resolution reduction level.

If this is a “scaling” change and there is a rotation, then

1. Set the rotation back to 0 (unrotated).
2. After some delay (setTimeout), then do the sendCompressionAndScaling method.

Input / Output in order	Type	Comments
Input 1	number	This is the percentage of the screen bitmap's compression level on the target endpoint's display. It is (0, 0.75]. 0.5 or less is best for speed consideration. 1 means that there is no compression. If the value is not in the valid range, this method returns false. If the value is negative, we will use the recorded previous value.
Input 2	number	This is the percentage of the render resolution reduction on the target display resolution. It is (0, 1]. 1 means no reduction. If the value is not in the valid range, this method returns false. 0.5 and 1.0 result in best performance. If the value is negative, we will use the recorded previous value.
Output	Boolean	True if it is successful. This will trigger onScreenResize event.

Sample files that use this:

- EndpointRDPInband.html

4.8 Rotate the rendered remote desktop

One can use EMA_SERVER_REMOTE_DESKTOP object's rotation variable, and then plus or minus one to determine the input value of the following method to do 90 degree CW or CCW rotation.

Call setRotation method:

Rotate the rendering of the remote desktop.

Input / Output in order	Type	Comments
Input	number	This is the rotation to set. Internally, while the value is < 0, the value += 4. Then the rotation value is the value % 4. The final internal rotation value can be 0, 1, 2, and 3. 0 is no rotation. Each is a 90 degree clock-wise rotation from the previous.
Output	none	This will trigger onScreenResize event.

Sample files that use this:

- EndpointRDPInband.html

4.9 Refresh the rendering of remote desktop

Call `sendRefresh` method:

Instruct the target endpoint agent to refresh all remote desktop data, which triggers a rendering refresh on the canvas element.

Sample files that use this:

- EndpointRDPInband.html

4.10 Send text message to other remote desktop connections to this endpoint

Call `sendMessage` method:

Send a message to all other connected remote RDPs to this endpoint, excluding self.

Input / Output in order	Type	Comments
Input	string	This is the message. < and > character will be replaced with encoded version < and >.
Output	none	This will trigger onMessage event on other connected RDPs.

Sample files that use this:

- EndpointRDPInband.html

4.11 Send Ctrl–Alt–Del to endpoint

Call `sendCtrlAltDelMsg` method:

This will run Ctrl + Alt + Del on the endpoint to bring up Windows task manager.

Sample files that use this:

- EndpointRDPInband.html

4.12 Get displays from remote endpoint

Call `getDisplayNumbers` method:

Get the displays from the remote endpoint. The result will be at onGetDisplays event.

Sample files that use this:

- EndpointRDPInband.html

4.13 Set the target display to render on the canvas element

Call `setDisplay` method:

Choose the display on the remote endpoint to display on the canvas element.

Input / Output in order	Type	Comments
Input	number	This is the target display number. The choice of display numbers should be obtained from <code>getDisplayNumbers</code> method. To choose all displays, use 65535 or 0. If the number is not in the list of numbers returned by <code>onGetDisplays</code> event and not 0, this call will be ignored.
Output	none	If the display selection changes after this call, it will trigger <code>onGetDisplays</code> event.

Sample files that use this:

- `EndpointRDPIInband.html`

4.14 Expand the KVM canvas to full screen mode

Call `requestFullScreen` method:

This will use web browser's full screen API to put the KVM canvas into full screen mode.

Call `isCanvasInFullScreenMode` method:

This will return true if the KVM canvas is in full screen mode.

Call `getTargetCanvasRenderingSizeInFullScreeMode` method:

Return the target canvas rendering size in full screen mode. If the canvas is in full screen mode, this returns an array of 2 elements for width and height. If the canvas is not in full screen mode, the width and size will be 0.

Sample files that use this:

- `EndpointRDPIInband.html`

4.15 Send clipboard string as keystrokes

Call `sendStringAsKeyStroke` method:

This will send the clipboard contents of the system running the Intel EMA web based user interface as a series of keycodes to the endpoint, simulating keystrokes typed at the endpoint's keyboard.

4.16 Get clipboard text from Internet Explorer

Call `getClipboardTextFromIE` method:

This method is for use when the browser running the Intel EMA web based user interface is Internet Explorer. It will send the clipboard contents to the target endpoint, but with special consideration for the Internet Explorer browser.

4.17 Releasing the Alt-key

Call `releaseFocusCanvas` method:

During a remote desktop session, if an Alt-key sequence is sent to the remote desktop, it is possible for the character sequence to be truncated so that only the Alt-key character code is sent. This leaves the endpoint in a state where the Alt-key is pressed, so that subsequent character codes are interpreted as part of an Alt-key sequence. This method releases the Alt-key on the remote endpoint.

4.18 Event callbacks

This subsection is a summary of all the events that will be used in each feature provided by this library. The reader can go to each feature description directly and then refer back to this section when needed.

4.18.1 onStateChanged event

This will be triggered when state is changed.

Parameter in order	Type	Comments
Parameter 1	EMA_SERVER_REMOTE_DESKTOP	This is the event source.
Parameter 2	number	This is the new state. Please see definitions of EMA_GLOBAL.EMA_DESKTOP_STATE object.
Parameter 3	object	This is the HTML canvas element on the calling web page.

When the new state is EMA_GLOBAL.EMA_DESKTOP_STATE.DISCONNECTED,

- The user should un-grab the mouse, touch, and keyboard events from the HTML.
- The user should check if tunnelErrorCode member > EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.NONE for tunnel failure. For the error value mapping, check EMA_GLOBAL.EMA_TUNNEL_ERRORCODE object definition of ema.js for details.

4.18.2 onDebugMessage event

This event will be fired when some debugging information is needed.

Parameter in order	Type	Comments
Parameter 1	string	This is the debug information.

4.18.3 onScreenResize event

This will be triggered when the rendering set up is changed (e.g., rotation change, render resolution reduction change, etc.). The callback handler should update the HTML canvas element.

Parameter in order	Type	Comments
Parameter 1	EMA_SERVER_REMOTE_DESKTOP	This is the event source.
Parameter 2	number	This is the target screen width.

Parameter in order	Type	Comments
Parameter 3	number	This is the target screen height.
Parameter 4	object	This is the HTML canvas element on the calling web page.

4.18.4 onMessage event

This will be triggered when someone uses sendMessage to send a message to all other connected remote RDPs.

Parameter in order	Type	Comments
Parameter 1	string	This is the message.
Parameter 2	EMA_SERVER_REMOTE_DESKTOP	This is the event source.

4.18.5 onConnectCountChanged event

This will be triggered when the number of remote desktop connections on the target endpoint changes.

Parameter in order	Type	Comments
Parameter 1	number	This is the number of remote desktop connections on the target endpoint.
Parameter 2	EMA_SERVER_REMOTE_DESKTOP	This is the event source.

4.18.6 onGetDisplays event

This will be triggered by getDisplayNumbers method.

Parameter in order	Type	Comments
Parameter 1	number	This is the number of displays on the target endpoint plus 1. If there is only 1 display, the value is 0. Else, the value is the number of displays plus 1. The extra 1 is with value 65535, which means "all displays".
Parameter 2	number	This is the currently-selected display. If this is 65535, then we are selecting "all displays".
Parameter 3	number[]	This is an array of numbers. Each number is the display index (starting from 1) available on the target endpoint. For example, for an endpoint with 2 displays, the array is [65535, 1, 2]. 65535 means "all displays". If there is only 1 display, then this is empty array.

5 ema_terminal.js

This library is used together with ema.js. It provides the methods to create a terminal html element that can perform remote terminal on the target endpoint.

5.1 Limitation

The terminal only displays the last 80 by 25 characters.

5.2 Access and policy checking

Policy checking is done based on the following order:

1. For both in-band and out-of-band terminal, when the front-end tries to get the best Swarm server for the target endpoint, AJAX server's Alert session checks if the target endpoint group has TCPRELAY policy.
 - If it fails, AJAX server sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.POLICYNOTALLOWED) back and disconnects the tunnel connection.
2. For remote (in-band, out-of-band) terminal, remote file, and remote RDP operations, when the initial setup message is sent to the target endpoint (via Intel® EMA servers), the endpoint agent checks Command, FILES, and KVM policies.
 - If they are all disabled, endpoint will reject this operation and AJAX server sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.REMOTE_DISCONNECT) back and disconnects the tunnel connection.
3. When sending the start message:
 - For in-band terminal, when the starting "terminal prompt" message is sent to AJAX server, AJAX server's Alert session checks if the target endpoint group has Command policy. If it fails, AJAX server sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.POLICYNOTALLOWED) back and disconnects the tunnel connection.
 - For out-of-band terminal, when the starting "terminal/desktop Intel® AMT authentication" message is sent to AJAX server, AJAX server's Alert session checks if the target endpoint group has Command or KVM policy. If it fails, AJAX server sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.POLICYNOTALLOWED) back and disconnects the tunnel connection.

User access checking:

- For both in-band and out-of-band, when the front-end tries to get the best Swarm server for the target endpoint, AJAX server's Alert sessions checks the following
 - If the user does not have Execute right for the endpoint, AJAX server adds audit log to the Intel® EMA database, sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.ACCESSDENIED) back, and disconnects the tunnel connection.
 - If the endpoint Id has correct format but does not exist, AJAX server adds audit log to the Intel® EMA database, sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.ACCESSDENIED) back, and disconnects the tunnel connection.
 - If the Intel® EMA server cannot get the correct information for this endpoint, AJAX server adds audit log to the Intel® EMA database, sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.TARGETNOTFOUND) back, and disconnects the tunnel connection.

- If AJAX server cannot find the Swarm server for this endpoint, AJAX server sends tunnel error (EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.NOLEADERFOUND) back, and disconnects the tunnel connection.

5.3 Main entry point

EMA_SERVER_REMOTE_TERMINAL function:

Input / Output in order	Type	Comments
Input	string	This is the id of the HTML element (where terminal will be displayed) of the web page. The library will use document.getElementById to find the element.
Output	EMA_SERVER_REMOTE_TERMINAL	This is the main object to be used for connecting to Intel EMA tunnels and for controlling the terminal HTML element.

The high-level operations flow after the object creation is:

1. Use termResetScreen, processVt100String, and termDraw methods to set the Terminal HTML element to initial state. For an example of such HTML element, please see the sample codes.
2. Create EMA_SERVER object and connect to AJAX server.
3. Use start method to start the tunnel connection to endpoint agent via Swarm server.
4. Use grabKeyInput method to start intercepting the user key press and sending to Intel EMA.

The high-level operations flow for disconnect from terminal is:

1. Use stop and unGrabKeyInput methods.
2. Use termResetScreen, processVt100String, and termDraw methods to set the Terminal HTML element to initial state.

Sample files that use this:

- EndpointTerminal.html

5.4 Initialize the terminal HTML element

Call termResetScreen method. This reset the internal variables.

Call processVt100String method. The only input is the string message that the user wants to display on the terminal HTML element. This puts the string message in the corresponding variables.

Call termDraw method. This renders the terminal HTML element based on the internal variables.

Sample files that use this:

- EndpointTerminal.html

5.5 Start the terminal

Check tunnelMode member first. If it is not EMA_GLOBAL.EMA_TERMINAL_TUNNEL.DISCONNECTED, then do not continue.

Call start method:

This will start the tunnel connection to endpoint agent via Swarm server, and start the flow of remote terminal.

Input / Output in order	Type	Comments
Input 1	EMA_SERVER	This is the object returned by the main entry point of ema.js. The connection to AJAX server (performed by ema.js) should be established already.
Input 2	string	The value is the target endpoint ID, in HEX format (without hyphen).
Output	Boolean	True if it is successful. If this method works, start monitoring onStateChanged event.

Call grabKeyInput method:

This intercepts the keyboards event on the HTML document.

Input / Output in order	Type	Comments
Input	object	This is the HTML document object. If this is not defined, then we use the "document" object.

Sample files that use this:

- EndpointTerminal.html

5.6 Stop the terminal

Call stop method:

This will stop the tunnel connection.

Call unGrabKeyInput method:

This releases the HTML document's key events handlers

Input / Output in order	Type	Comments
Input	object	This is the HTML document object. If this is not defined, then we use the "document" object.

Sample files that use this:

- EndpointTerminal.html

5.7 Start the out-of-band Intel® AMT terminal

This is very similar to section "Start the terminal". The only difference is to call startAMT, instead of start method.

The Intel AMT credential for this endpoint is obtained internally from the Intel EMA database.

This is a Serial-Over-LAN terminal. After you are connected, you need to boot to BIOS with Serial-over-LAN turned on and the BIOS will send a text version of the BIOS into the Intel AMT serial port. Then you can see the BIOS displayed.

startAMT method:

Input / Output in order	Type	Comments
Input 1	EMA_SERVER	This is the object returned by the main entry point of ema.js. The connection to AJAX server (performed by ema.js) should be established already.
Input 2	string	The value is the target endpoint ID, in HEX format (without hyphen).
Input 3	number	This is the port for Intel® AMT connection. This is either 16994 (if not using TLS) or 16995 (if using TLS).
Output	Boolean	True if it is successful. If this method works, start monitoring onStateChanged event.

Sample files that use this:

- EndpointTerminal.html

5.8 Event callbacks

This subsection is a summary of all the events that will be used in each feature provided by this library. The reader can go to each feature description directly and then refer back to this section when needed.

5.8.1 onStateChanged event

This will be triggered when state is changed.

Parameter in order	Type	Comments
Parameter 1	EMA_SERVER_REMOTE_TERMINAL	This is the event source.
Parameter 2	number	This is the new state. Please see definitions of EMA_GLOBAL.EMA_TERMINAL_STATE object.

When the new state is EMA_GLOBAL.EMA_TERMINAL_STATE.DISCONNECTED,

- The user should un-grab the keyboard input.
- The user should check if tunnelErrorCode member > EMA_GLOBAL.EMA_TUNNEL_ERRORCODE.NONE for tunnel failure. For the error value mapping, check EMA_GLOBAL.EMA_TUNNEL_ERRORCODE object definition of ema.js for details.

5.8.2 onDebugMessage event

This event will be fired when some debugging information is needed.

Parameter in order	Type	Comments
Parameter	string	This is the debug information.

6 ema_amt.js

This library is used together with ema.js. It provides the features to perform several Intel® AMT power-related operations.

6.1 Access and policy checking

For endpoint group policy checking during getRoutingCookie, AJAX server checks the following

- If the target endpoint does not allow TCPRelay policy, getRoutingCookie will fail.

For endpoint group policy checking for the subsequent each Intel® AMT command via WSMAN, AJAX server checks the following

- If the target endpoint does not allow the associated power policy for the target power action, Error (EMA_GLOBAL.EMA_AMTWSMAN_ERROR.ACTIONFORBIDDEN_POLICY) will be returned to onWsmanResponse callback.

For the user access checking during getRoutingCookie, AJAX server checks the following

- If the user does not have Execute right for this endpoint, AJAX server adds audit log to the Intel® EMA database and getRoutingCookie will fail.

For the user access checking for the subsequent each Intel® AMT command via WSMAN, AJAX server checks the following

- If the user does not have Execute right for this endpoint, AJAX server adds audit log to the Intel® EMA database and error (EMA_GLOBAL.EMA_AMTWSMAN_ERROR.ACTIONFORBIDDEN_USERACCESS) will be returned to onWsmanResponse callback.

6.2 Main entry point

Before we create EMA_AMTSTACK object, use getRoutingCookie method in ema.js library to get the "{cookie encrypted}". Then the final path is "https://{EMA url}/ajax/redirection/wsman?RC={encrypted routing cookie}". The Intel® AMT credential for this endpoint is included in the encrypted routing cookie.

Please use a new EMA_AMTSTACK object for each power action. Reset to BIOS or Power to BIOS flow has several steps, and the user can create a new EMA_AMTSTACK object at the start of the step and use the same object for the remaining steps until the flow fails and finishes.

EMA_AMTSTACK function:

Input / Output in order	Type	Comments
Input	string	This is the target AJAX path for the target endpoint, in "https://{EMA url}/ajax/redirection/wsman" format.
Input	string	This is the encrypted routing cookie.
Output	EMA_AMTSTACK	This is the main object to be used for performing Intel AMT operations on the target endpoint.

Sample files that use this:

- EndpointAMTOperations.html

6.3 Perform simple power operations

Use the following methods (which all have the same type of input and output):

- **powerOn:** Put the system to ACPI S0 state.
- **hardwareReset:** Perform master bus reset.
- **sleepDeep:** Put the system to ACPI S3 state.
- **softwareReset:** Perform power cycle (off- soft), which puts the system to ACPI S5 and then S0.
- **hibernate:** Put the system to ACPI S4 state.
- **powerOffSoft:** Perform power off soft, which puts the system to ACPI S5.
- **powerOffSoftGraceful:** Perform power off soft graceful, which orderly puts the system to ACPI S5.
- **hardwareResetGraceful:** Perform master bus reset graceful, which does orderly system shutdown and then does hardware reset.

Input / Output in order	Type	Comments
Input	function	This is the onWsmanResponse function defined below.
Output	Boolean	True if it is successful.

onWsmanResponse callback function:

Parameter in order	Type	Comments
Parameter 1	EMA_AMTSTACK	This is the event source.
Parameter 2	string	This is the DMTF class name of the original WSMAN call.
Parameter 3	object	<p>This is the WSMAN XML response parsed into JSON object.</p> <p>The object has Header variable, which is another JSON object. The user does not need to process this.</p> <p>The object has Body variable, which is another JSON object. For our current WSMAN method call request, Body has ReturnValue and ReturnValueStr two string members.</p> <ul style="list-style-type: none">• If ReturnValue is '0', the Intel AMT operations in the WSMAN request is successful.• ReturnValueStr has the Intel AMT status string based on the ReturnValue.
Parameter 4	object	<p>This is the XMLHttpRequest object. Check XMLHttpRequest.status first.</p> <p>If XMLHttpRequest.status is 200</p> <ul style="list-style-type: none">• It means that the WSMAN request call went through successfully. However, it does not mean that the Intel® AMT operation included in the WSMAN method call request is successful.• For WSMAN get and put request, this means that the operation is successful.

Parameter in order	Type	Comments
		<p>Else,</p> <ul style="list-style-type: none"> Try parse the XMLHttpRequest.responseText into JSON object. The JSON object include two members. <ul style="list-style-type: none"> Code: This is the error code. Check the definition of EMA_GLOBAL.EMA_AMTWSMAN_ERROR enumeration object for details. Message: This is the string text for the code. If the parsing fails, show the XMLHttpRequest.status only.

Sample files that use this:

- EndpointAMTOperations.html

6.4 Perform opt-in operations

Please check the public documentation for IPS_OptInService DMTF class for more details.

Start the opt-in process

The flow in order is:

- Use get method for IPS_OptInService.
- At the response callback function,
 - Check the XMLHttpRequest.status code for 200. Then,
 - Check response (3rd parameter)'s Body.OptInRequired and Body.OptInState. If the operations you want to do require opt-in, then check OptInState. Only when OptInState is 0 (Not Started), you need to start the opt-in.
- If we need to start the opt-in, use startOptIn method.

Cancel the opt-in process

The flow in order is:

- Use get method for IPS_OptInService.
- At the response callback function,
 - Check the XMLHttpRequest.status code for 200. Then,
 - Check response (3rd parameter)'s Body.OptInRequired and Body.OptInState. If the operations you want to do require opt-in, then check OptInState. Only when OptInState is 1 (Requested) or 2 (Displayed), you need to cancel the opt-in.
- If we need to cancel the opt-in, use cancelOptIn method.

Send the opt-in consent code

The flow in order is:

1. Use get method for IPS_OptInService.
2. At the response callback function,
 - a. Check the XMLHttpRequest.status code for 200. Then,
 - b. Check response (3rd parameter)'s Body.OptInRequired and Body.OptInState. If the operations you want to do require opt-in, then check OptInState. Only when OptInState is 2 (Displayed), you need to send the consent code.
3. If we need to send the consent code, use sendOptInCode method.

get method

This performs the WSMAN GET.

Input / Output in order	Type	Comments
Input 1	string	This is the DMTF class name, e.g., CIM_AssociatedPowerManagementService or AMT_BootSettingData.
Input 2	function	This is the onWsmanResponse function defined above. The differences are <ul style="list-style-type: none"> • This is a WSMAN GET, so one can use XMLHttpRequest.status to check the operation success. • The returned data of the target DMTF class object is at response (3rd parameter)'s Body.
Output	Boolean	True if it is successful.

startOptIn method

Perform a WSMAN method call to request an opt-in code. Intel® AMT will display the code on the target endpoint.

Input / Output in order	Type	Comments
Input	function	This is the onWsmanResponse function defined above.
Output	Boolean	True if it is successful.

cancelOptIn method

Perform a WSMAN method call to cancel a previous opt-in request.

Input / Output in order	Type	Comments
Input	function	This is the onWsmanResponse function defined above.
Output	Boolean	True if it is successful.

sendOptInCode method

Perform a WSMAN method call to send Intel AMT opt-in code to the target Intel AMT endpoint.

Input / Output in order	Type	Comments
Input 1	number	This is the consent code. This needs to be a valid number.
Input 2	function	This is the onWsmanResponse function defined above.
Output	Boolean	True if it is successful.

Sample files that use this:

- EndpointAMTOperations.html

6.5 Perform reset (or power) to BIOS

Please check the public documentation for AMT_BootSettingData and CIM_BootService DMTF classes for more details.

The operation flow in order is:

1. Use section "Perform opt-in operations" to get access to Boot settings first. It is possible that the opt-in is not required or the opt-in is already "Received" or "In Session". In this case, do not need to send in the consent code, and you can go to the next step directly.
2. Upon successfully getting access to Boot settings,
 - a. Use get method for AMT_BootSettingData class to get the settings.
 - b. After getting the settings, adjust the settings data based on your need.
 - c. Use put method for AMT_BootSettingData class to put the modified settings.
3. Upon successfully setting the Boot settings, use setBootConfig method to activate the boot setting after the next power-on or reset.
4. Upon successfully running setBootConfig, run the power action methods you want, e.g., powerOn or hardwareReset.

put method:

This performs WSMAN PUT.

Input / Output in order	Type	Comments
Input 1	string	This is the DMTF class name, e.g., CIM_AssociatedPowerManagementService or AMT_BootSettingData.
Input 2	function	<p>This is the onWsmanResponse function defined above.</p> <p>The differences are</p> <ul style="list-style-type: none"> • This is a WSMAN PUT, so one can use XMLHttpRequest.status to check the operation success. • The returned data of the target DMTF class object is at response (3rd parameter)'s Body.
Output	Boolean	True if it is successful.

setBootConfig method:

Perform a WSMAN method call to activate the boot setting after the next power on or reset command.

Input / Output in order	Type	Comments
Input	function	This is the onWsmanResponse function defined above.
Output	Boolean	True if it is successful.

Sample files that use this:

- EndpointAMTOperations.html

6.6 Redirect to Intel® AMT web site on the endpoint

We do not really need ema_amt.js. We need to use getRoutingCookie method (for in-band cookie) in ema.js to get the routing cookie first. Then the target URL is "https://{EMA url}:8084/MeshR2TX/redirection/?RC={cookie encoded}".

The user access checking and endpoint group policy checking are done at getRoutingCookie method.

Then, open that URL in a new web browser (or tab).

Sample files that use this:

- EndpointAMTOperations.html