

These API's are found on components that inherits from the baseclass

Platform Core

JmlNode API

toString
setWidth
setHeight
setLeft
setTop
setZIndex
getWidth
getHeight
getLeft
getTop
getZIndex
isVisible
show
hide
loadJML
insertJML
setTabIndex
focus
blur
enable
disable
enableAlignment
enableAnchoring

JmlDom API

appendChild
insertBefore
removeNode
getElementsByTagName
cloneNode

Class API

__addJmlLoader
__addJmlDestroyer
hasFeature
dispatchEvent
addEventListener
removeEventListener
hasEventListener
destroy

Javeline IDEF's

__PACKAGE
__DEBUG
__HIGHLIGHTING
__STATUS
__WITH_INCLUDES
__WITH_DOM
__WITH_ECMAEXT
__WITH_BACKBTN
__WITH_KBSUPPORT
__WITH_HTML_PARSING
__WITH_PRINT
__SUPPORT_IE
__SUPPORT_Gecko
__SUPPORT_Safari
__SUPPORT_Opera
__INC_ALL
__J<NAME>

Html Root document

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:j="http://www.javeline.com/2005/Platform" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <head>
    <title>My Application</title>
    <link href="myapp-style.css" rel="stylesheet" type="text/css"/>
    <script src="jpf_debug.js"></script>
  </head>
  <body>
    <j:skin src="myapp-skins.xml" />
    <j:include src="myapp-interface.xml" />
    <!-- any Html or Jml tag -->
  </body>
</html>
```

Settings Syntax

```
<j:appsettings debug="true" debug-teleport="true" debug-type="memory" disable-f5="true" title="MyApp" width="500" height="600" min-width="480" min-height="300" resizable="true" icon="icoWin.gif" auto-disable-actions="true" enable-rightclick="true" allow-select="true" >
```

Jml Component Syntax

```
<j:application xmlns:j="http://www.javeline.com/2005/Platform">
  <j:componentName
    <!-- basics -->
    id="" left="" top="" width="" height="" show="" zIndex=""
    <!-- ways to extend the Jml dynamically -->
    jml=""
    <!-- special attributes -->
    disabled="" focussable="" disable-keyboard="" tabseq=""
    <!-- attributes dictated by various baseclasses -->
    render="runtime" use-delay="" caching="" rename="" skin="" multiselect=""
    autoselect="" selectable="" selected="" validation="" datatype="" min-value=""
    max-value="" max-length="" min-length="" notnull="" required="" check-equal=""
    invalidmsg="" smartbinding="" bindings="" actions="" model="" ref=""
    dragEnabled="" dropEnabled="" dragMoveEnabled="" actiontracker=""
    <!-- inline events -->
    on<event name>=""
  >
    <!-- Insert possible children here Jml or Html -->
  </j:componentName>
</j:application>
```

Special JML Elements

```
<j:application xmlns:j="http://www.javeline.com/2005/Platform">
  <j:script src="" />
  <j:script><![CDATA[ alert(1) ]]></j:script>
  <j:loader><div>Loading...</div></j:loader>
  <j:teleport />
</j:application>
```

Javascript access

Every Jml component or model with an id attribute is directly accessible by that id in JavaScript: alert(myComponent.getValue());

To get a reference to a SmartBinding call getBindClass() on the component it is assigned to or do: jpf.JMLParser.getSmartBinding("name");

To insert some dynamically created Jml do this: parentComp.insertJML(dynJmlDom);

Debugging functions

jpf.alert_r
jpf.varDump

jpf.history API

Backbutton Support
addPoint
goForward
goBack
clear

jpf.printServer API

printContent
setContent
Print

jpf.JMLParser API

getBindClass
getActionTracker
replaceNode
createElement
loadSubNode

jpf API

setStyleRule
importStylesheet
importCssString
loadStylesheet
getStyle
addEventListener
lookup
sleep
isNot
setReference
debugMsg
showDebug
formErrorString
destroy

ECMA Ext.

Global

extend
serialize
unserialize
isNull
isArray
copyObject
copyArray
htmlEntities
htmlEntitiesdecode

Array

copy
arrayAdd
isEqual
makeUnique
contains
indexOf
pushUnique
search
remove
removeIndex
insertIndex
Invert

Math

decToHex
hexToDec

String

trim
repeat
count
pad