

## DpdAPI Doc Vers. 1.0

Document Author: ACosta (e-mail: armincosta@hotmail.com)

---

### *dmdl.dmdlDmdlEngine* **Class DmdlEngine**

`dmdl.dmdlDmdlEngine.DmdlEngine`

---

public class **DmdlEngine** extends [dmdlRuntime](#)

This API may be called from different programming languages, Dmdl scripts included.

---

### Field Summary

static int	<a href="#">MAX_PRIORITY</a>
static int	<a href="#">MIN_PRIORITY</a>
static int	<a href="#">NORM_PRIORITY</a>

### Fields inherited from class `dmdl.dmdlEncDec`.[mslsByteCodeDescriptor](#)

....

### Fields inherited from class `dmdl.dmdlDmdlEngine`.[mslsDmdlParams](#)

....

### Fields inherited from class `dmdl.dmdlDmdlEngine`.[mslsDmdlDebugger](#)

[LEVEL0](#), [LEVEL1](#), [LEVEL2](#), [LEVEL3](#), [LEVEL4](#)

# Constructor Summary

**DpdlEngine**(boolean targetClient, [Display](#) display\_)  
main constructor DpdlEngine targetClient = true means that this Engine is used as Decoder

# Method Summary

void	<a href="#">DPDLAPI_allocateDpdlGui</a> ( String[] gui_data, String[] const_var, String[] type)
int	<a href="#">DPDLAPI_callExtension</a> ( String name)
int	<a href="#">DPDLAPI_clearCache</a> ( String className)
int	<a href="#">DPDLAPI_clearClientRX</a> (int device)
int	<a href="#">DPDLAPI_clearClientTX</a> (int device)
boolean	<a href="#">DPDLAPI_clearDpdlChunks</a> ( String service_name, String[] chunks) this function is used to clear the CHUNK allocations in the Cached data
int	<a href="#">DPDLAPI_connectServerToClients</a> ()
int	<a href="#">DPDLAPI_createBTClient</a> (int instance, int inquiryCode, int authentication, int authorization, int encryption)
int	<a href="#">DPDLAPI_createBTImageClient</a> ()
int	<a href="#">DPDLAPI_createBTImageServer</a> ()
int	<a href="#">DPDLAPI_createBTServer</a> (int inquiryCode, int authentication, int authorization, int encryption)
int	<a href="#">DPDLAPI_createObexServer</a> (int inquiryCode)

int	<a href="#"><u>DPDLAPI_createThread</u></a> (int instance, String code_, int priority, int milliseconds)
int	<a href="#"><u>DPDLAPI_deleteBTServer</u></a> ()
void	<a href="#"><u>DPDLAPI_deleteDpdlScripting</u></a> ()
int	<a href="#"><u>DPDLAPI_deleteThread</u></a> (int instance)
int	<a href="#"><u>DPDLAPI_disconnectServerToClients</u></a> ()
int	<a href="#"><u>DPDLAPI_discoverBTDevices</u></a> ()
int	<a href="#"><u>DPDLAPI_discoverBTFinished</u></a> ()
int	<a href="#"><u>DPDLAPI_discoveryClientFinished</u></a> (int client_instance)
int	<a href="#"><u>DPDLAPI_discoveryServerFinished</u></a> ()
int	<a href="#"><u>DPDLAPI_execCode</u></a> (String file)
int	<a href="#"><u>DPDLAPI_execDpdlCode</u></a> (Vector dpdl_script)
int	<a href="#"><u>DPDLAPI_execScript</u></a> (String script_dpdl_)
int	<a href="#"><u>DPDLAPI_execStartupCode</u></a> ()
String	<a href="#"><u>DPDLAPI_getClientBTAddr</u></a> (int instance)
String	<a href="#"><u>DPDLAPI_getEngineVersion</u></a> ()
DataObj[]	<a href="#"><u>DPDLAPI_getNativeResultSet</u></a> ()

	<p>this method returns the Native DataObj classes</p>
int	<a href="#"><u>DPDLAPI_getNrResults</u></a> ()
String	<a href="#"><u>DPDLAPI_getResultSet</u></a> (int pos, String var)
String	<a href="#"><u>DPDLAPI_getServerBTAddr</u></a> ()
String	<a href="#"><u>DPDLAPI_getServerVisibleBTAddr</u></a> ()
Object	<a href="#"><u>DPDLAPI_getVar</u></a> (int instance, String name)
String	<a href="#"><u>DPDLAPI_getVersion</u></a> ()
int	<a href="#"><u>DPDLAPI_includeDpdlCode</u></a> ( String file)
int	<a href="#"><u>DPDLAPI_initDpdlScripting</u></a> ()
void	<a href="#"><u>DPDLAPI_initEngine</u></a> () this method initializes the Dpdl core engine
boolean	<a href="#"><u>DPDLAPI_isTargetClient</u></a> () this method returns the target Client status of the Dpdl core engine
int	<a href="#"><u>DPDLAPI_mallocDpdlService</u></a> ( String service_name) both functions are used to allocate the Decoder && dpdl-GuiEngine if the Service was NOT allocated previously, the DataBuilder reboots --> proper Dpdl-Byte[] is copies to RAM and processed
boolean	<a href="#"><u>DPDLAPI_mallocDpdlService</u></a> ( String service_name, <a href="#"><u>msslVisualizer</u></a> visualData)
void	<a href="#"><u>DPDLAPI_printAPI</u></a> ()
int	<a href="#"><u>DPDLAPI_printDpdlScript</u></a> ( String serviceName, int mode, String file) this function prints the Dpdl code embedded in a given paket

String	<a href="#"><u>DPDLAPI_processXPath</u></a> (byte[] file, String query)
String	<a href="#"><u>DPDLAPI_processXPath</u></a> (String file, String query)
int	<a href="#"><u>DPDLAPI_processXPathVar</u></a> (String varName, String query, String mode, String outputFile)
String	<a href="#"><u>DPDLAPI_readClientData</u></a> (int device)
String	<a href="#"><u>DPDLAPI_readServerData</u></a> ()
void	<a href="#"><u>DPDLAPI_rebootDpdl</u></a> () this method reboots the Dpdl core engine
int	<a href="#"><u>DPDLAPI_searchClientsOnServer</u></a> ()
int	<a href="#"><u>DPDLAPI_selectDpdlService</u></a> (String service_name, String chunk, String const_) this method is used to decode and process one chunk of data
boolean	<a href="#"><u>DPDLAPI_selectDpdlServices</u></a> (String service_name, String[] chunks, String[] const_, <a href="#"><u>mslsVisualizer</u></a> visual) this method is used to decode and process chunks of data
int	<a href="#"><u>DPDLAPI_sendClientData</u></a> (int device, String data)
int	<a href="#"><u>DPDLAPI_sendObex</u></a> (String data)
int	<a href="#"><u>DPDLAPI_sendObexImage</u></a> (String img_name)
int	<a href="#"><u>DPDLAPI_sendServerData</u></a> (String data)
int	<a href="#"><u>DPDLAPI_serviceDiscoveryClientFinished</u></a> (int client_instance)
int	<a href="#"><u>DPDLAPI_serviceDiscoveryServerFinished</u></a> ()

int	<a href="#"><u>DPDLAPI_setVar</u></a> (int instance, String name, Object obj)
void	<a href="#"><u>DPDLAPI_shutdownDpdl</u></a> () this method shuts-down the Dpdl core engine
void	<a href="#"><u>DPDLAPI_storeDpdlRuntimeInfo</u></a> () this method stores the Current RuntimeInfo of the Dpdl core engine
void	<a href="#"><u>DPDLAPI_swapDpdlChunk</u></a> ( String service_name, String chunk) this method swaps CHUNKS of a given Dpdl-Service, does not if the CHUNKS are already swapped DPDL_Script feature ,(no arrays there!!!)
int[]	<a href="#"><u>DPDLAPI_swapDpdlChunks</u></a> ( String service_name, String[] chunks) this method swaps CHUNKS of a given Dpdl-Service, does not if the CHUNKS are already swapped
<a href="#"><u>Display</u></a>	<a href="#"><u>getDisplay</u></a> () get the display of this device implementation
<a href="#"><u>Displayable</u></a>	<a href="#"><u>getDisplayable</u></a> ()
void	<a href="#"><u>initEngine</u></a> () this method initializes the Dpdl core engine
void	<a href="#"><u>setDisplayable</u></a> ( <a href="#"><u>Displayable</u></a> mainUI_) set the main displayable object

**Methods inherited from class dpdl.dpdlDpdlEngine.[dpdlRuntime](#)**

...

**Methods inherited from class dpdl.dpdlDpdlEngine.[ROMStorage](#)**

...

**Methods inherited from class dpdl.dpdlEncDec.[mslsByteCodeDescriptor](#)**

...

**Methods inherited from class dpdl.dpdlDpdlEngine.[mslsDpdlParams](#)**

...

### Methods inherited from class `dpdl.dpdlDpdlEngine`.[mslsDpdlDebugger](#)

[Error](#), [LogDpdlMsg](#), [p](#)

### Methods inherited from class `Object`

...

## Field Detail

### MIN\_PRIORITY

```
public static int MIN_PRIORITY
```

---

### MAX\_PRIORITY

```
public static int MAX_PRIORITY
```

---

### NORM\_PRIORITY

```
public static int NORM_PRIORITY
```

## Constructor Detail

### DpdlEngine

```
public DpdlEngine(boolean targetClient,  
                  Display display_)
```

main constructor `DpdlEngine targetClient = true` means that this Engine is used as Decoder

## Method Detail

### getDisplay

```
public Display getDisplay()
```

get the display of this device implementation

**Returns:**

Display

---

## **setDisplayable**

```
public void setDisplayable(Displayable mainUI_)
```

set the main displayable object

---

## **getDisplayable**

```
public Displayable getDisplayable()
```

---

## **initEngine**

```
public void initEngine()
```

this method initializes the Dpdl core engine

---

## **DPDLAPI\_initEngine**

```
public void DPDLAPI_initEngine()
```

this method initializes the Dpdl core engine

---

## **DPDLAPI\_printAPI**

```
public void DPDLAPI_printAPI()
```

---

## **DPDLAPI\_initDpdlScripting**

```
public int DPDLAPI_initDpdlScripting()
```

---

## **DPDLAPI\_execStartupCode**

```
public int DPDLAPI_execStartupCode()
```



---

## **DPDLAPI\_execCode**

```
public int DPDLAPI_execCode( String file)
```

---

## **DPDLAPI\_execScript**

```
public int DPDLAPI_execScript( String script_dpdl_)
```

---

## **DPDLAPI\_createThread**

```
public int DPDLAPI_createThread(int instance,  
                                String code_,  
                                int priority,  
                                int milliseconds)
```

---

## **DPDLAPI\_getVar**

```
public Object DPDLAPI_getVar(int instance,  
                              String name)
```

---

## **DPDLAPI\_setVar**

```
public int DPDLAPI_setVar(int instance,  
                           String name,  
                           Object obj)
```

---

## **DPDLAPI\_deleteThread**

```
public int DPDLAPI_deleteThread(int instance)
```

---

## **DPDLAPI\_discoverBTDevices**

```
public int DPDLAPI_discoverBTDevices()
```

---

## **DPDLAPI\_discoverBTFinished**

```
public int DPDLAPI_discoverBTFinished()
```

---

## **DPDLAPI\_createBTServer**

```
public int DPDLAPI_createBTServer(int inquiryCode,  
                                   int authentication,  
                                   int authorization,  
                                   int encryption)
```

---

## **DPDLAPI\_createBTClient**

```
public int DPDLAPI_createBTClient(int instance,  
                                   int inquiryCode,  
                                   int authentication,  
                                   int authorization,  
                                   int encryption)
```

---

## **DPDLAPI\_createObexServer**

```
public int DPDLAPI_createObexServer(int inquiryCode)
```

---

## **DPDLAPI\_sendObex**

```
public int DPDLAPI_sendObex( String data)
```

---

## **DPDLAPI\_sendObexImage**

```
public int DPDLAPI_sendObexImage( String img_name)
```

---

## **DPDLAPI\_discoveryClientFinished**

```
public int DPDLAPI_discoveryClientFinished(int client_instance)
```

---

## **DPDLAPI\_serviceDiscoveryClientFinished**

```
public int DPDLAPI_serviceDiscoveryClientFinished(int client_instance)
```

---

## **DPDLAPI\_getClientBTAddr**

```
public String DPDLAPI_getClientBTAddr(int instance)
```

---

## **DPDLAPI\_sendClientData**

```
public int DPDLAPI_sendClientData(int device,  
                                   String data)
```

---

## **DPDLAPI\_readClientData**

```
public String DPDLAPI_readClientData(int device)
```

---

## **DPDLAPI\_clearClientTX**

```
public int DPDLAPI_clearClientTX(int device)
```

---

## **DPDLAPI\_clearClientRX**

```
public int DPDLAPI_clearClientRX(int device)
```

---

## **DPDLAPI\_discoveryServerFinished**

```
public int DPDLAPI_discoveryServerFinished()
```

---

## **DPDLAPI\_serviceDiscoveryServerFinished**

```
public int DPDLAPI_serviceDiscoveryServerFinished()
```

---

## **DPDLAPI\_getServerBTAddr**

```
public String DPDLAPI_getServerBTAddr()
```

---

## **DPDLAPI\_getServerVisibleBTAddr**

```
public String DPDLAPI_getServerVisibleBTAddr()
```

---

## **DPDLAPI\_searchClientsOnServer**

```
public int DPDLAPI_searchClientsOnServer()
```

---

## **DPDLAPI\_connectServerToClients**

```
public int DPDLAPI_connectServerToClients()
```

---

## **DPDLAPI\_sendServerData**

```
public int DPDLAPI_sendServerData( String data)
```

---

## **DPDLAPI\_readServerData**

```
public String DPDLAPI_readServerData()
```

---

## **DPDLAPI\_disconnectServerToClients**

```
public int DPDLAPI_disconnectServerToClients()
```

---

## **DPDLAPI\_createBTImageServer**

```
public int DPDLAPI_createBTImageServer()
```

---

## **DPDLAPI\_createBTImageClient**

```
public int DPDLAPI_createBTImageClient()
```

---

## **DPDLAPI\_deleteBTServer**

```
public int DPDLAPI_deleteBTServer()
```

---

## **DPDLAPI\_deleteDpdlScripting**

```
public void DPDLAPI_deleteDpdlScripting()
```

---

## DPDLAPI\_execDpdIcode

```
public int DPDLAPI_execDpdIcode( Vector dpdl_script)
```

---

## DPDLAPI\_includeDpdIcode

```
public int DPDLAPI_includeDpdIcode( String file)
```

---

## DPDLAPI\_getVersion

```
public String DPDLAPI_getVersion()
```

---

## DPDLAPI\_getEngineVersion

```
public String DPDLAPI_getEngineVersion()
```

---

## DPDLAPI\_shutdownDpdI

```
public void DPDLAPI_shutdownDpdI()
```

this method shuts-down the DpdI core engine

---

## DPDLAPI\_rebootDpdI

```
public void DPDLAPI_rebootDpdI()
```

this method reboots the DpdI core engine

---

## DPDLAPI\_isTargetClient

```
public boolean DPDLAPI_isTargetClient()
```

this method returns the target Client status of the DpdI core engine

**Returns:**

targetClient

---



this function is used to clear the CHUNK allocations in the Cached data

---

## DPDLAPI\_storeDpdlRuntimeInfo

```
public void DPDLAPI_storeDpdlRuntimeInfo()
```

this method stores the Current RuntimeInfo of the Dpdl core engine

---

## DPDLAPI\_getResultSet

```
public String DPDLAPI_getResultSet(intpos, String var)
```

---

## DPDLAPI\_getNativeResultSet

```
public DataObj[] DPDLAPI_getNativeResultSet()
```

this method returns the Native DataObj classes

---

## DPDLAPI\_getNrResults

```
public int DPDLAPI_getNrResults()
```

---

## DPDLAPI\_mallocDpdlService

```
public int DPDLAPI_mallocDpdlService( String service_name)
```

both functions are used to allocate the Decoder && dpdl-GuiEngine if the Service was NOT allocated previously, the DataBuilder reboots --> proper Dpdl-Byte[] is copied to RAM and processed

---

## DPDLAPI\_mallocDpdlService

```
public boolean DPDLAPI_mallocDpdlService( String service_name, mslsVisualizer_visualData)
```

---

## DPDLAPI\_clearCache

```
public int DPDLAPI_clearCache( String className)
```

---

## DPDLAPI\_allocateDpdlGui

```
public void DPDLAPI_allocateDpdlGui( String[]gui_data,  
                                     String[]const_var,  
                                     String[]type)
```

---

## DPDLAPI\_processXPath

```
public String DPDLAPI_processXPath(byte[]file,  
                                   String query)
```

---

## DPDLAPI\_processXPath

```
public String DPDLAPI_processXPath( String file,  
                                   String query)
```

---

## DPDLAPI\_processXPathVar

```
public int DPDLAPI_processXPathVar( String varName,  
                                    String query,  
                                    String mode,  
                                    String outputFile)
```

---

## DPDLAPI\_callExtension

```
public int DPDLAPI_callExtension( String name)
```

---

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---