

BOND
adapt

Specialist staffing software



User Guide: **Adapt JavaScript Services**

Adapt JavaScript Services

Date	22-Jun-2012
Document Version	3.0
Product Version	V11.7

Bond International Software (UK) Ltd

Courtlands, Parklands Avenue, Tel: +44 (0) 1903 707070
Goring, West Sussex BN12 4NG Fax: +44 (0) 1903 707080 Email: sales@bond.co.uk

www.bondadapt.com



Document History

Version	Date	Author	Summary
1.0	27-Oct-2010	Martin Armstrong	Doc created
1.1	08-Jun-2012	Helen Samoilova	Template applied. Updated based on ttp 19934.
1.2	12-Jun-2012	Olga Fedotova	Updated and reviewed
1.3	13-Jun-2012	Alexey Dorokhin	Examples inserted
2.0	14-Jun-2012	Olga Fedotova	Baselined
2.1	20-Jun-2012	Helen Samoilova	Updated with ttp20845, applied for V11.6.8.3, V11.7
2.2	21-Jun-2012	Olga Fedotova	Reviewed and Updated
3.0	22-Jun-2012	Olga Fedotova	Baselined

Bond International Software (UK) Ltd

Courtlands, Parklands Avenue, Goring, West Sussex BN12 4NG
Tel: +44 (0) 1903 707070 Fax: +44 (0) 1903 707080 Email: sales@bond.co.uk

www.bondadapt.com

Table of Contents

Overview	4
loadModule service.....	5
UI Module	6
openEntity(callData).....	7
openWorkflow(callData)	9
runBO(callData).....	14
importCV(callData).....	16
importCVList(callData)	18
Logging	20
getNewLogger(name).....	21
enableLogger(name)	22
disableLogger(name).....	23
viewLogger(name).....	24
enabled.....	25
log(text)	26
name	27
entries	28
Sample.....	29

Overview

It is possible to trigger CV import, invoke Entities, Workflows or Business Objects in Adapt from the external web applications by accessing and executing the Adapt JavaScript Services.

Access to the Adapt functions is provided through the Adapt JavaScript object, that must be loaded first within the external webpage:

```
<script
  type="text/javascript"
  src="http://<AdaptServerIP:port>/WebApp/external/Adapt.js">
</script>
```

Once a loaded, Adapt object provides JavaScript Services (Modules) to trigger Adapt functions, the **loadModule** service must be set to make available Modules and core utilities such as Logging mechanism.

Modules

Module	Description
UI	Holds functions which are used to trigger the following Adapt functions: import CV, invoke Entity, Workflow or Business Object in the Adapt Standard User Runtime.

Modules

Core Utilities

Core Utilities	Description
Logging	Provides the ability to store log entries in the named logger objects.

Core Utilities

Bond International Software (UK) Ltd

Courtlands, Parklands Avenue, Goring, West Sussex BN12 4NG
Tel: +44 (0) 1903 707070 Fax: +44 (0) 1903 707080 Email: sales@bond.co.uk

www.bondadapt.com

loadModule service

loadModule(moduleName, onloadFunction) function loads and enables the specified Module of the Adapt core object. When the module is loaded, the given function is executed.

Note Adapt core object must be loaded first to enable loading of its Module.

For more information on how to load Adapt core object, refer to the [Overview](#) section of this document.

Syntax

```
Adapt.loadModule(Adapt.ModuleNames.[ModuleName],[onloadFunctionName](){})
```

Parameters

ModuleName

Usage: Mandatory

Definition: The name of the Module to load.

Note The full list of all the available Modules can be found in the [Overview](#) section of this document

onloadFunction

Usage: Mandatory

Definition: The function to run once the module has finished loading.

Example

Load the **UI** Module and execute the specified function once the **UI** Module has been loaded. The function returns alert saying 'UI Module is loaded':

```
<script type="text/javascript">  
  Adapt.loadModule(Adapt.ModuleNames.UI,function(){alert('UI Module is loaded')});  
</script>
```

UI Module

UI Module holds functions under the Adapt object which are used to import CV, invoke Entity, Workflow or Business Object in the Adapt Standard User Runtime.

Functions

Function	Description
openEntity(callData)	Opens the specified Entity in the Adapt Standard User Runtime
openWorkflow(callData)	Opens the specified Workflow View in the Adapt Standard User Runtime
runBO(callData)	Invokes the specified Business Object in the Adapt Standard User Runtime
ImportCV(callData)	Invokes CV Import module and passes a CV file stored locally or on the external website to the CV Import module that will further download and parse information contained within that file.
ImportCVList(callData)	Invokes CV Import and passes a list of CV files stored locally or on the external website to the CV Import module that will further download and parse information contained within that files

Adapt.UI Module Functions

Note UI Module must be loaded first to make its functions accessible.

For more information on how to load UI Module, refer to the [loadModule service](#) section of this document.

Note **callData** is a set of parameters that are passed in the form of a JSONObject.

Note It is possible to import CV, invoke Entity, Workflow or Business Object in Adapt by means of http-formatted External Request strings. UI Module Functions are built on top of the http-formatted External Request strings and possess the same capabilities in triggering actions in the Adapt Standard User Runtime.

For more information about the HTTP-formatted External Requests, refer to the [Engine Documentation\Standard component documentation\External Open Request\ExternalRequests_UserGuide](#).

openEntity(callData)

openEntity(callData) function opens the specified Entity in the Standard User Runtime.

Syntax

```
Adapt.UI.openEntity({  
  "id"           : [EntityID],  
  "roleId"       : [RoleID],  
  "segmentId"    : [SegmentID],  
  "domainId"     : [DomainID],  
  "profileId"    : [ProfileID],  
})
```

callData

A set of parameters that are passed in the form of a JSONObject:

id

Usage: Mandatory

Definition: ID of the Entity to open

roleId

Usage: Optional

Definition: ID of the Role to open the Entity as.

Useful when Entity possesses several Roles.

Note

If the target Entity possesses several Roles and **roleId** parameter is not specified in the request string, the Entity View opens using the default Role.

segmentId

Usage: Optional

Definition: ID of the Segment to open

domainId

Usage: Conditional (must be specified if **profileId** parameter is used)

Definition: ID of the Domain the target Entity belongs to

Note

If **domainId** is not specified, the Domain the user is currently logged into will be considered.

If **domainId** is specified but differs from the ID of the Domain that the user is currently logged into, the request will be refused and message box asking to log off the Adapt in order to log in back to the correct Domain appears in the Standard User Runtime. In this case **openEntity** function must be triggered one more time.

In case user is not logged in at the moment, the appropriate Domain needs to be



manually selected on the Adapt Login page.

profileId

Usage: Conditional (must be specified if **domainId** parameter is used)

Definition: ID of the Domain Profile the target Entity belongs to

Note

If **profileId** is not specified, the Domain Profile the user is currently logged into will be considered.

If **profileId** is specified but differs from the ID of the Domain Profile that the user is currently logged into, the request will be refused and message box asking to log off the Adapt in order to log in back to the correct Domain Profile appears in the Standard User Runtime. In this case **openEntity** function must be triggered one more time.

In case user is not logged in at the moment, the appropriate Domain Profile needs to be manually selected on the Adapt Login page.

Example

Open the specified Entity View for the selected Role on the predefined Segment in the specified Domain Profile:

```
Adapt.UI.openEntity({  
  "id"           :12345,  
  "roleId"       :123,  
  "segmentId"    :11698,  
  "domainId"     :1002,  
  "profileId"    :6716236  
})
```

openWorkflow(callData)

openWorkflow(callData) function opens the specified Workflow View in the Standard User Runtime.

Syntax

```
Adapt.UI.openWorkflow({
  "id"           : [WorkflowID],
  "callerView"  : "E [EntityID]",
  "domainID"    : [domainID],
  "profileID"   : [profileID],
  "workflowParam" :
    {
      "[ControlName1]": "[value1]",
      ...,
      "[ControlNameN]": "[valueN]"
    }
})
```

callData

A set of parameters that are passed in the form of a JSONObject:

id

Usage: Mandatory

Definition: ID of the Workflow View to open

callerView

Usage: Optional

Definition: ID of the Entity to open first (i.e. one the Workflow will be launched from) prefixed with **E**. Useful in case Workflow picks up data from the Entity it was called from.

domainId

Usage: Conditional (must be specified if **profileId** is used)

Definition: ID of the Domain the target Workflow View belongs to

Note

If **domainId** is not specified, the Domain the user is currently logged into will be considered.

If **domainId** is specified but differs from ID of the Domain the user is currently logged into, the request will be refused, and message box asking to log off the Adapt in order to log in back to the correct Domain will appear in the Standard User Runtime. In this case **openWorkflow** function must be triggered one more time.

In case user is not logged in at the moment, the appropriate Domain will need to be manually selected on the Adapt Login page.



profileId

Usage: Conditional (must be specified if **domainId** is used)

Definition: ID of the Domain Profile the target Workflow View belongs to

Note If **profileId** is not specified, the Domain Profile the user is currently logged into will be considered.

If **profileId** is specified but differs from ID of the Domain Profile the user is currently logged into, the request will be refused, and message box asking to log off the Adapt in order to log in back to the correct Domain Profile will appear in the Standard User Runtime. In this case **openWorkflow** function must be triggered one more time.

In case user is not logged in at the moment, the appropriate Domain Profile will need to be manually selected on the Adapt Login page.

workflowParam

Usage: Optional

Definition: Names of Workflow Controls with values to pre-populate a Workflow View in the form of URLencoded JSON Object.

Note Workflow Controls are pre-populated with values in the same way as the values are set using a Business Object.

Note Only Control names need to be specified as the full Control path is obtained based on the ID of the Workflow View to open.

Syntax of passing the parameters depends on the type of Controls to be pre-populated:

Reference Text Box Control:

```
"workflowParam": {" [ReferenceControlName] ": [EntityId/UserId] }
```

Alpha Text Box Control:

```
"workflowParam": {" [AlphaControlName] ":" [AlphaValue] " }
```

Numeric Text Box Control:

```
"workflowParam": {" [NumericControlName] ":" [NumericValue] " }
```

Note Format of the "NumericValue" inserted into a Numeric Text Box Control on a Workflow View may be different from the format specified in the function. This is caused by applying the automatic formatting according to the Numeric parsing rules.

For more information on Numeric parsing rules, refer to the [Engine Documentation\Standard component documentation\ Date, Time, Numbers Parser\ DateTimeNumbers_ParsingRules](#).

Bond International Software (UK) Ltd

Courtlands, Parklands Avenue, Goring, West Sussex BN12 4NG
Tel: +44 (0) 1903 707070 Fax: +44 (0) 1903 707080 Email: sales@bond.co.uk

www.bondadapt.com





Date Text Box Control:

```
"workflowParam":{"[DateControlName]":"[DateValue]"}
```

Note Format of the "DateValue" inserted into a Date Text Box Control on a Workflow View may be different from the format specified in the request. This is caused by applying the automatic formatting according to the Date parsing rules.

For more information on Date parsing rules, refer to the [Engine Documentation\ Standard component documentation\ Date, Time, Numbers Parser\ DateTimeNumbers_ParsingRules](#).

Time Text Box Control:

```
"workflowParam":{"[TimeControlName]":"[TimeValue]"}
```

Note Format of the "TimeValue" inserted into a Time Text Box Control on a Workflow View may be different from the format specified in the request. This is caused by applying the automatic formatting according to the Time parsing rules.

For more information about Time parsing rules, refer to the [Engine Documentation\ Standard component documentation\ Date, Time, Numbers Parser\ DateTimeNumbers_ParsingRules](#).

YesNo Text Box Control:

```
"workflowParam":{"[YesNoControlName]":"[Y/N]"}
```

Code Text Box Control:

```
"workflowParam":{"[CodeControlName]":[CodeId]}
```

Email Text Box Control:

```
"workflowParam":{"[EmailControlName]":"[EmailAddress]"}
```

URL Text Box Control:

```
"workflowParam":{"[URLControlName]":"[URL]"}
```

Span Text Box Control:

```
"workflowParam":{"[SpanControlName]":"[SpanValue]"}
```

Note "SpanValue" inserted into a Span Text Box Control on a Workflow View must be specified in Decimal Hours format.

For more information about Decimal Hours formatting in Span Text Box Controls, refer to the [Engine Documentation\ Configuration Tools documentation\ Screen Tool\](#)



Controls\SpanTextBox_UserGuide.

Social Profile Text Box Control:

```
"workflowParam":{"[SocialProfileControlName]":"[SocialProfileURL]"}
```

Check Box Control:

```
"workflowParam":{"[CheckBoxControlName]":"[Y/N]"}
```

Text Area Control:

```
"workflowParam":{"[TextAreaControlName]":"[Text]"}
```

Template Control:

```
"workflowParam":{"[TemplateControlName]":"[TemplateName]"}
```

Drop Down Control:

```
"workflowParam":{"[DropDownControlName]":  
  {  
    "entry1":{"id":[id1],"value":"[dropDownValue1]","selected":"[true]"},  
    .../  
    "entryN":{"id":[idN],"value":"[dropDownValueN]"  
  }  
}
```



Example

Open the specified Workflow View with pre-populated Controls from the specified Entity in the specified Domain Profile:

```
Adapt.UI.openWorkflow({
  "id" : 9028173,
  "callerView" : "E10635",
  "domainId" : 1002,
  "profileId" : 6716236,
  "workflowParam" :
  {
    "ReferenceControlName" : 10076,
    "AlphaControlName" : "testtest test",
    "NumericControlName" : "1234567890",
    "DateControlName" : "11/06/12",
    "TimeControlName" : "9:30",
    "YesNoControlName" : "Y",
    "YesNoControlName" : "N",
    "CodeControlName" : 1303875,
    "EmailControlName" : "email@test.com",
    "URLControlName" : "www.bond-adapt.co.uk",
    "SpanControlName" : "0.5",
    "SocialProfileControlName" : "http://uk.linkedin.com/pub/Name/48/378/391",
    "CheckBoxControlName" : "Y",
    "TextAreaControlName" : "TextArea, TEXTarea, Area",
    "DropDownControlName" :
    {
      "entry1" : {"id":10635,"value":"value1"},
      "entry2" : {"id":10642,"value":"value2","selected":"true"}
    }
  }
})
```

Bond International Software (UK) Ltd

Courtlands, Parklands Avenue, Goring, West Sussex BN12 4NG
Tel: +44 (0) 1903 707070 Fax: +44 (0) 1903 707080 Email: sales@bond.co.uk

www.bondadapt.com

runBO(callData)

runBO(callData) function invokes the specified Business Object.

Syntax

```
Adapt.UI.runBO({
  "id"                : [WorkflowID],
  "callerView"       : "E [EntityID]",
  "domainId"         : [DomainID],
  "profileId"        : [ProfileID],
  "boParam"          :
    {
      "boParam[1]" : "[value1]",
      ...
      "boParam[N]" : "[valueN]"
    }
})
```

callData

A set of parameters that are passed in the form of a JSONObject:

id

Usage: Mandatory

Definition: ID of the Business Object to execute

callerView

Usage: Optional

Definition: ID of the Entity to open first (prefixed with **E**).

Useful in case Business Object picks up Entity related data.

domainId

Usage: Conditional (must be specified if **profileId** is used)

Definition: ID of the Domain the target Business Object belongs to.

Note

If **domainId** is not specified, the Domain the user is currently logged into will be considered.

If **domainId** is specified but differs from ID of the Domain the user is currently logged into, the request will be refused, and message box asking to log off the Adapt in order to log in back to the correct Domain will appear in the Standard User Runtime. In this case **runBO** function must be triggered one more time.

In case user is not logged in at the moment, the appropriate Domain will need to be manually selected on the Adapt Login page.

profileId

Bond International Software (UK) Ltd

Courtlands, Parklands Avenue, Goring, West Sussex BN12 4NG
Tel: +44 (0) 1903 707070 Fax: +44 (0) 1903 707080 Email: sales@bond.co.uk

www.bondadapt.com



Usage: Conditional (must be specified if **domainId** is used)

Definition: ID of the Domain Profile the target Business Object is run under.

Note If **profileId** is not specified, the Domain Profile the user is currently logged into will be considered.

If **profileId** is specified but differs from ID of the Domain Profile the user is currently logged into, the request will be refused, and message box asking to log off the Adapt in order to log in back to the correct Domain Profile will appear in the Standard User Runtime. In this case **runBO** function must be triggered one more time.

In case user is not logged in at the moment, the appropriate Domain Profile will need to be manually selected on the Adapt Login page.

boParam

Usage: Optional

Definition: Parameters Names with values to pass into the Business Object.

Business Objects Parameters are passed in the form of a JSONObject.

Note `_PARAM(index)` macros is used to get externally provided parameters within a Business Object:
`_PARAM(1)` corresponds to the "boParam1" and so forth.

Example

Run the specified Business Object with the set of parameters from the specified Entity in the specified Domain Profile:

```
Adapt.UI.runBO({
  "id"           :7006902,
  "callerView"  : "E10635",
  "domainId"    :1002,
  "profileId"   :6716236,
  "boParam"     :
    {
      "boParam1" : "Some Text",
      "boParam2" : "12345.25",
      "boParam3" : "12 June 2012",
      "boParam4" : "10642"
    }
})
```

importCV(callData)

importCV(callData) function invokes CV Import module and passes a CV file stored locally or on the external website to the CV Import module that will further download and parse information contained within that file.

Syntax

```
Adapt.UI.importCV({  
  "cvURL"      : "URLEncoded[CVdownloadURL] ",  
  "linkedServer" : "[LinkedServerName] ",  
  "domainId"   : [DomainID],  
  "profileId"  : [ProfileID]  
})
```

callData

A set of parameters that are passed in the form of a JSONObject:

cvURL

Usage: Mandatory

Definition: **Encoded path** to the local file in the **file:///** or **file://** format, or **encoded URL** to the file stored on the external website in the **http://** format from which a CV can be downloaded.

linkedServer

Usage: Conditional (must be used if **cvURL** contains path to the file stored on the external website in the **http://** format)

Definition: Name of the linked server.

Note

When **linkedServer** parameter is used, the file's URL will be passed to the specified linked server first in order to verify that file to be downloaded is not harmful. The ADAPT_AUTH and ADAPT_SSN parameters get automatically appended to the download URL, so that the linked server can authenticate the received request.

After the downloaded file passes verification, linked server transfers it to the application server for further processing by Adapt CV Import module.

For more information about Linked Servers, refer to the [Engine Documentation\ Linked Server\ AdaptLinkedServers_UserGuide](#).

domainId

Usage: Conditional (must be specified if **profileId** is used)

Definition: ID of the Domain the CV will be imported to.

Note

If **domainId** is not specified, the Domain the user is currently logged into will be considered.



If **domainId** is specified but differs from ID of the Domain the user is currently logged into, the request will be refused, and message box asking to log off the Adapt in order to log in back to the correct Domain will appear in the Standard User Runtime. In this case **importCV** function must be triggered one more time.

In case user is not logged in at the moment, the appropriate Domain will need to be manually selected on the Adapt **Login** page.

profileId

Usage: Conditional (must be specified if **domainId** is used)

Definition: ID of the Domain Profile the CV will be imported under.

Note

If **profileId** is not specified, the Domain Profile the user is currently logged into will be considered.

If **profileId** is specified but differs from ID of the Domain Profile the user is currently logged into, the request will be refused, and message box asking to log off the Adapt in order to log in back to the correct Domain Profile will appear in the Standard User Runtime. In this case **importCV** function must be triggered one more time.

In case user is not logged in at the moment, the appropriate Domain Profile will need to be manually selected on the Adapt **Login** page.

Example

Pass a CV download URL (<http://test.com/CV1.doc>) to the CV Import module of the predefined Domain Profile through the specified linked Server:

```
Adapt.UI.importCV({
  "cvURL"      : "http%3A%2F%2Ftest.com%2FCV1.doc",
  "linkedServer" : "ServerName",
  "domainId"   : 1002,
  "profileId"  : 7654321
})
```

Pass a locally stored CV file path (<file:///d:/CV.doc>) to the CV Import module of the predefined Domain Profile:

```
Adapt.UI.importCV({
  "cvURL"      : "file%3A%2F%2Fd%3A%2FCV.doc",
  "domainId"   : 1002,
  "profileId"  : 7654321
})
```

importCVList(callData)

importCVList(callData) function invokes CV Import and passes a list of CV files stored locally or on the external website to the CV Import module that will further download and parse information contained within that files.

Syntax

```
Adapt.UI.importCVList({  
  "cvURLs"      : ["URLEncoded[CVdownloadURL1]", ..., "URLEncoded[CVdownloadURLN]"],  
  "linkedServer" : "[LinkedServerName]",  
  "domainId"    : [DomainID],  
  "profileId"   : [ProfileID]  
})
```

callData

A set of parameters that are passed in the form of a JSONObject:

cvURLs

Usage: Mandatory

Definition: Array of **Encoded paths** to the local files in the **file:///** or **file://** format, or **encoded URLs** to the files stored on the external websites in the **http://** format from which CVs can be downloaded.

linkedServer

Usage: Conditional (must be used if **cvURLs** contains path to the files stored on the external websites in the **http://** format)

Definition: Name of the linked server.

Note

When **linkedServer** parameter is used, the files URLs will be passed to the specified linked server first in order to verify that files to be downloaded are not harmful. The ADAPT_AUTH and ADAPT_SSN parameters get automatically appended to the download URL, so that the linked server can authenticate the received request.

After the downloaded files pass verification, linked server transfers them to the application server for further processing by Adapt CV Import module.

For more information about Linked Servers, refer to the [Engine Documentation\Linked Server\AdaptLinkedServers_UserGuide](#).

domainId

Usage: Conditional (must be specified if **profileId** is used)

Definition: ID of the Domain the CV will be imported to.

Note

If **domainId** is not specified, the Domain the user is currently logged into will be considered.



If **domainId** is specified but differs from ID of the Domain the user is currently logged into, the request will be refused, and message box asking to log off the Adapt in order to log in back to the correct Domain will appear in the Standard User Runtime. In this case **importCVList** function must be triggered one more time.

In case user is not logged in at the moment, the appropriate Domain will need to be manually selected on the Adapt Login page.

profileId

Usage: Conditional (must be specified if **domainId** is used)

Definition: ID of the Domain Profile the CV will be imported under.

Note

If **profileId** is not specified, the Domain Profile the user is currently logged into will be considered.

If **profileId** is specified but differs from ID of the Domain Profile the user is currently logged into, the request will be refused, and message box asking to log off the Adapt in order to log in back to the correct Domain Profile will appear in the Standard User Runtime. In this case **importCVList** function must be triggered one more time.

In case user is not logged in at the moment, the appropriate Domain Profile will need to be manually selected on the Adapt Login page.

Example

Pass a list of CV download URLs (<http://test.com/CV1.doc>, <http://test.com/CV2.doc>, <http://test.com/CV3.doc>) to the CV Import module of the predefined Domain Profile through the specified linked Server:

```
Adapt.UI.importCVList({
  "cvURLs"      : ["http%3A%2F%2Ftest.com%2FCV1.doc",
                  "http%3A%2F%2Ftest.com%2FCV2.doc",
                  "http%3A%2F%2Ftest.com%2FCV1.doc"],
  "linkedServer" : "ServerName",
  "domainId"     : 1002,
  "profileId"    : 7654321
})
```

Pass a list of locally stored files: <file:///d:/CV1.doc>, <file:///d:/CV2.doc>, <file:///d:/CV3.doc> to the CV Import module of the predefined Domain Profile:

```
Adapt.UI.importCVList({
  "cvURLs"      : ["file%3A%2F%2F%2Fd%3A%2FCV1.doc",
                  "file%3A%2F%2F%2Fd%3A%2FCV2.doc",
                  "file%3A%2F%2F%2Fd%3A%2FCV3.doc"],
  "domainId"     : 1002,
  "profileId"    : 7654321
})
```

Logging

There is an ability to store log entries in the named logger objects. Adapt modules have their own loggers registered in here and you can add your own. Loggers can be enabled/disabled and the log entries can be viewed in a popup window.

Functions

Functions	Description
getNewLogger(name)	Creates and returns a new instance of Simple Logger object
enableLogger(name)	Enables the Logger object to accept the new log entries
disableLogger(name)	Disables the Logger object to accept the new log entries
viewLogger(name)	Opens a popup window listing the log entries of the specified Logger object

Logging Module Functions

Properties

Properties	Description
enabled	Indicates whether the Logger object accepts the new log entries
log(text)	Adds log entries to the Logger object
name	Returns the name of the Logger object
entries	Returns array of the log entries of the Logger object

Logging Module Properties

getNewLogger(name)

getNewLogger(name) function creates, registers and returns a new instance of Logger object.

Syntax

```
var [LoggerObjectName] = Adapt.Logging.getNewLogger(' [LoggerName] ');
```

Parameters

LoggerName

Usage: Mandatory

Definition: Name of the Logger to be created and returned.

Example

Create, register and return a new logger named 'myLogger':

```
var myLoggerObject = Adapt.Logging.getNewLogger('myLogger');
```

enableLogger(name)

enableLogger(name) function enables the Logger object to accept the new log entries.

Syntax

```
Adapt.Logging.enableLogger ( ' [LoggerName] ' );
```

Parameters

LoggerName

Usage: Mandatory

Definition: Name of the Logger to be enabled.

Example

Enable the logger named 'myLogger':

```
Adapt.Logging.enableLogger ( 'myLogger' );
```

disableLogger(name)

disableLogger(name) function disables the Logger object, preventing it from accepting the new log entries.

Syntax

```
Adapt.Logging.disableLogger ( ' [LoggerName] ' );
```

Parameters

LoggerName

Usage: Mandatory

Definition: Name of the Logger to be disabled.

Example

Disable the logger named 'myLogger':

```
Adapt.Logging.disableLogger ( 'myLogger' );
```

viewLogger(name)

viewLogger(name) function opens a popup window listing the log entries of the specified Logger object.

Syntax

```
Adapt.Logging.viewLogger ( ' [LoggerName] ' );
```

Parameters

LoggerName

Usage: Mandatory

Definition: Name of the Logger.

Example

Open a popup window listing the log entries of the 'myLogger' Logger:

```
Adapt.Logging.viewLogger ( 'myLogger' );
```

enabled

enabled property indicates whether the Logger object accepts the new log entries.

Syntax

```
[LoggerObjectName].enabled;
```

Example

Indicate whether Logger object is enabled:

```
if (myLoggerObject.enabled)
```

Enable the Logger object to accept new log entries:

```
myLoggerObject.enabled = true
```

log(text)

log(text) property adds log entries to the Logger object.

Syntax

```
[LoggerObjectName].log([text]);
```

Example

Add log entry: 'log entry' to the Logger object:

```
myLoggerObject.log('Log entry');
```

name

name property returns the name of the Logger object.

Syntax

```
LoggerObjectName = [LoggerObjectName].name;
```

Example

Return the name of the 'myLoggerObject' Logger object:

```
loggerObjectName = myLoggerObject.name;
```

entries

entries property returns log entries array of the Logger.

Syntax

```
var [LogEntries] = [LoggerObjectName].entries;
```

Example

Return log entries array of the 'MyLoggerObject' object:

```
var LogEntries = MyLoggerObject.entries;
```

Sample

Sample source of the external web page that allows performing the following by accessing and executing the Adapt JavaScript Services:

- 1) Trigger CV Import in the Adapt Standard User Runtime;
- 2) Invoke Entity in the Adapt Standard User Runtime;
- 3) Open Workflow in the Adapt Standard User Runtime;
- 4) Run Business Object in the Adapt Standard User Runtime;
- 5) Log Users requests to Adapt and allow to view the Log.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>

<!--load Adapt JavaScript core Object-->
<script type="text/javascript"
src="http://123.45.67.89:8080/WebApp/external/Adapt.js" >
</script>

<script type="text/javascript">

    <!--load the UI Module-->
    Adapt.loadModule(Adapt.ModuleNames.UI,function () {});

    <!--create, register and return a new logger named 'myLogger'-->
    var logger=Adapt.Logging.getNewLogger('myLogger');

    <!--enable the logger-->
    logger.enabled=true;

    function openEntity() {
        <!--open Entity-->
        Adapt.UI.openEntity({
            "id" :10649,
            "roleId" :108,
            "segmentId" :11698,
            "domainId" :1002,
            "profileId" :6716236
        });

        <!--add entry in myLogger-->
        logger.log('openEntitiy is trigerred');
    };

    function openWorkflow() {
        <!--open Workflow and populate some fields on it-->
        Adapt.UI.openWorkflow({
```



```

    "id"                :9028173,
    "callerView"       : "E10635",
    "domainId"         :1002,
    "profileId"        :6716236,
    "workflowParam"   :
    {
        "REFENTITY"    : "10635",
        "ALPHA"        : "open Workflow test",
        "NUMERIC1"     : "1234567890",
        "DATESHORT"    : "11/06/12",
        "TIMESHORT"    : "9:30",
        "YESNOCHECKBOX" : "Y",
        "YESNOCOMBOBOX" : "N",
        "CODE1"        : "1303875",
        "EMAIL"        : "email@test.com",
        "URL"          : "www.bond-adapt.co.uk",
        "SPAN"         : "0.5",
        "SOCIALP"      : "http://uk.linkedin.com/pub/linkedin/48/378/391",
        "CheckBox1"    : "Y",
        "TextAreal"    : "TextArea, TEXTarea, Areaaaaa",
        "DropDown1"    :
        {
            "entry1"    : {"id":"10635","value":"value1"},
            "entry2"    : {"id":"10642","value":"value2","selected":"true"}
        }
    }
});

<!--add entry in myLogger-->
logger.log('openWorkflow is triggerred');
};

function runBO() {
    <!--invoke the Business Object and passes some parameters to it-->
    Adapt.UI.runBO({
        "id"                :7006902,
        "callerView"       : "E10635",
        "domainId"         :1002,
        "profileId"        :6716236,
        "boParam"         :
        {
            "boParam1"     : "Some Text",
            "boParam2"     : "12345.25",
            "boParam3"     : "12 June 2012",
            "boParam4"     : "10642"
        }
    });

    <!--add entry in myLogger-->
    logger.log('runBO is triggerred');
};

function importCV() {
    <!--passe a CV download URL to the CV Import module-->
    <!--CVImport module will further download and parse the document-->
    Adapt.UI.importCV({
        "cvURL"            : "http%3A%2F%2Ftest.com%2FCV1.doc",

```

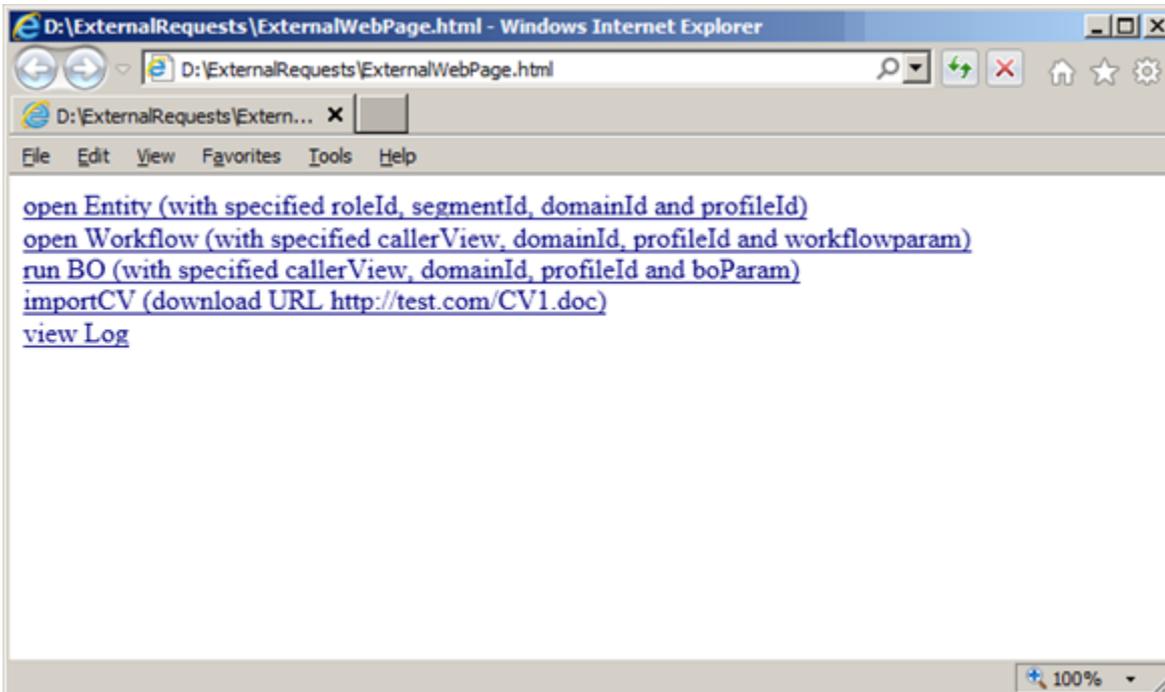


```
"linkedServer" : "ServerName",
"domainId"      : 1002,
"profileId"     : 6716236
});

<!--add entry in myLogger-->
logger.log('importCV is triggerred');
};

function viewLog() {
  <!--show a popup window listing the current log entries in the logger-->
  var logger=Adapt.Logging.viewLogger('myLogger');
};
</script>
</head>
<body>
  <a href="#" onclick='openEntity()'>open Entity (with specified roleId, segmentId,
domainId and profileId)</a><br>
  <a href="#" onclick='openWorkflow()'>open Workflow (with specified callerView,
domainId, profileId and workflowparam)</a><br>
  <a href="#" onclick='runBO()'>run BO (with specified callerView, domainId,
profileId and boParam)</a><br>
  <a href="#" onclick='importCV()'>importCV (download URL
http://test.com/CV1.doc)</a><br>
  <a href="#" onclick='viewLog()'>view Log</a><br>
</body>
</html>
```

Sample web page view:



Sample web page

Bond International Software (UK) Ltd

Courtlands, Parklands Avenue, Tel: +44 (0) 1903 707070
Goring, West Sussex BN12 4NG Fax: +44 (0) 1903 707080 Email: sales@bond.co.uk

www.bondadapt.com



Workflow that opens with pre-populated fields:

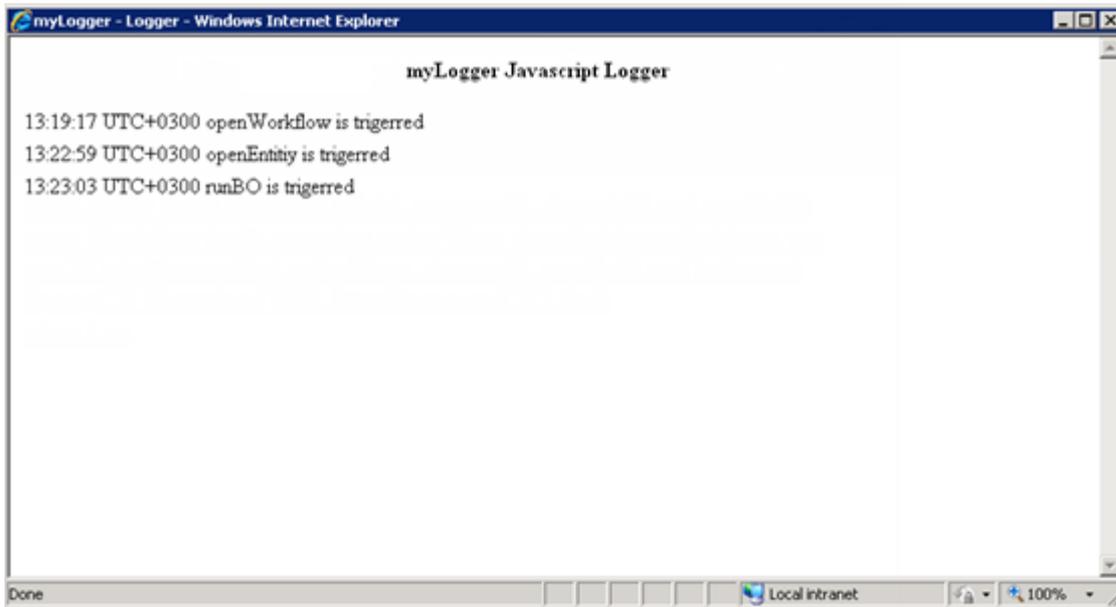
Verify TextBoxes controls

Alpha	<input type="text" value="open Workflow test"/>	YES/NO (textbox)	<input type="text"/>
Numeric (zero prec.)	<input type="text" value="1,234,567,890"/>	YES/NO (checkbox)	<input checked="" type="checkbox"/>
Numeric (prec. 2)	<input type="text"/>	YES/NO (combobox)	<input type="text" value="No"/>
Date (short)	<input type="text" value="11/06/12"/> <input type="button" value="←"/> <input type="button" value="→"/> <input type="button" value="📅"/>	YES/NO (radiogroup)	<input type="radio"/> Y <input type="radio"/> N
Date (long)	<input type="text"/> <input type="button" value="←"/> <input type="button" value="→"/> <input type="button" value="📅"/>	EMAIL	<input type="text" value="email@test.com"/> <input type="button" value="🔍"/>
Date (medium)	<input type="text"/> <input type="button" value="←"/> <input type="button" value="→"/> <input type="button" value="📅"/>	URL	<input type="text" value="www.bond-adapt.co.uk"/> <input type="button" value="🔍"/>
Date (custom)	<input type="text"/> <input type="button" value="←"/> <input type="button" value="→"/> <input type="button" value="📅"/>	SPAN	<input type="text" value="12:00"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
Date (medium, unformatted)	<input type="text"/> <input type="button" value="←"/> <input type="button" value="→"/> <input type="button" value="📅"/>	Reference (user)	<input type="text"/> <input type="button" value="🔍"/>
Code (by name)	<input type="text" value="F"/> <input type="button" value="🔍"/>	Reference (entity)	<input type="text" value="Head Office AD"/> <input type="button" value="🔍"/>
Code (by description)	<input type="text"/> <input type="button" value="🔍"/>	Social Profile	<input type="text" value="http://uk.linkedin.com/pub/lii"/> <input type="button" value="🔍"/>
Time (short)	<input type="text" value="09:30"/> <input type="button" value="📅"/>	CheckBox	<input checked="" type="checkbox"/>
Time (long)	<input type="text"/> <input type="button" value="📅"/>	DropDown	<input type="text" value="value2"/> <input type="button" value="📅"/>
Time (medium)	<input type="text"/> <input type="button" value="📅"/>	MailBox	<input type="text"/> <input type="button" value="📅"/>
Time (medium, unformatted)	<input type="text"/> <input type="button" value="📅"/>	Template	<input type="text"/>

<input type="radio"/> RadioGroup1 Value1	<div style="border: 1px solid gray; padding: 5px;"> TextArea TextArea, TEXTarea, Areaaaaa </div>
--	---

Opened Workflow with populated fields

Sample Logger resultant web page:



Logger resultant web page