

RapidImageView Frameset API
Embedding RapidImageView in 3rd party frameset
(v2.2)

Contents

Introduction.....	2
How to use RIV	2
Controlling interface.....	2
Response interface	4
Result codes	5
States.....	5
States diagram.....	6
Sample	6
Command execution diagram	7

Introduction

This HTTP based API allows eProof's viewer 'RapidImageView' to be integrated and embedded into a frame structure of a 3rd party web solution. In such 3rd party RapidImageView integration, eProof's ASP/HTML pages, top bar, menu etc will not be visible.

The 3rd party web pages can use this API to control the RapidImageView applet. They can instantiate it, hide, show and select the document to view.

This document describes version 2.2 of the API and sample that demonstrates how to use the API. The main difference of this version from previous ones is that it doesn't require adding eProof server to the list of trusted sites in case eProof and sample are located on different servers. Additionally, browser compatibility check was improved.

How to use RIV

To be able to use and manage RIV, a 3rd party solution should:

1. Provide a frame for RIV in its frameset structure and a script frame for sending commands to RIV (may be hidden). The content of these frames is loaded from a remote eProof server. So the 3rd party solution and eProof can reside on different servers. Minimizing/maximizing RIV frame is controlled by JavaScript functions in the 3rd party pages. The test page *Control.htm* includes those JavaScript functions that could be reused by the 3rd party solution.
2. Execute eProof controlling ASP in the script frame to send commands to RIV. It should be done by assigning URL that points to controlling ASP as script frame source. Command parameters should be passed as parameters of issued HTTP request (GET or POST).
Note: some commands (such as CreateRIV) have to be executed not in the script frame but directly in the RIV frame. This is connected to the limitations applied by some browsers to script code.
3. Accept response from RIV that returns status of the operation, RIV state and optionally additional extended information. eProof sends response by assigning URL that points to 3rd party product response page (e.g. asp, php, Coldfusion, etc.) as script frame source. This URL should be specified in every call to RIV as an argument. If it is omitted, RIV won't be able to return response. Arguments are passed as parameters of issued HTTP request.
4. Support eProof's custom sessions. To do this, the 3rd party solution should first accept the session ID (SID) in response on the CheckEnvironment command, store it and then use it in every subsequent command to RIV.

Controlling interface

This is the HTTP based interface that 3rd party web solution should use to control RIV. eProof implements this interface in its controlling ASP (*RIV_Control.asp*). The calls are asynchronous by their nature. Parameters are passed as parameters of HTTP request (GET or POST). Result of executing commands is returned using response interface.

The following commands are supported:

1. **Check client environment** – query information about client OS, browser version and Java virtual machine version and check whether client environment meets RIV requirements. In case it doesn't meet, additional parameter ExtResult is returned in response that specifies exactly what's wrong. You could reuse eProof's browser compatibility page by opening it in appropriate frame.
Note that this command starts new eProof's custom session. This means, additional parameter SID is returned in response that identifies new session. This parameter needs to be used in every subsequent call to RIV.
 - a. Parameters
 - i. Command = 'CheckEnvironment'

- ii. ReturnURL – URL of the 3rd party page that will be used by eProof to return results of commands execution
- 2. **Logon** – authenticate user and initialize user session in eProof. If account matching provided user name doesn't exist, it is automatically created (version 2.2.). To be able to logon, client environment should be checked first.
 - a. Parameters:
 - i. Command = 'Logon'
 - ii. ReturnURL – URL of the 3rd party page that will be used by eProof to return results of commands execution
 - iii. SID – session ID
 - iv. UID – user name
 - v. Language – interface language, represented by a 2 character ISO code ("EN", "FR", "DE", "NL" others will follow later), optional.
 - vi. CID - Delano serial number in a form of a string prefixed with "7483234" (hard coded activation key), optional.
- 3. **Install RIV** – RIV installer is executed, it checks whether RIV files are on client's PC and whether their version matches the versions on the server. In case there are no files or the versions don't match, RIV files are downloaded and saved on client's PC. Note: though you need to execute this command once per session, actual RIV files downloading occurs only once. To be able to install RIV, client environment should be checked first.
 - a. Parameters:
 - i. Command = 'InstallRIV'
 - ii. ReturnURL – URL of the 3rd party page that will be used by eProof to return results of commands execution
 - iii. SID – session ID
- 4. **Create RIV** – instantiate RIV applet. To be able to instantiate RIV, it should be installed first. Note: this command needs to be executed in the frame where you want RIV to reside.
 - a. Parameters
 - i. Command = 'CreateRIV'
 - ii. ReturnURL – URL of the 3rd party page that will be used by eProof to return results of commands execution
 - iii. SID – session ID
 - iv. Timeout – specifies how long to wait until RIV is created (in seconds). If RIV is not created within this interval, an error 'RIV not created' is returned. If this parameter is not specified, default value is used (60 seconds).
 - v. AppletWidth – optional parameter, specifies desired RIV width in pixels. If the parameter is omitted, width="100%" is used.
 - vi. AppletHeight - optional parameter, specifies desired RIV height in pixels.
 On some client configuration (Mac/IE/Safari) an applet with width="100%" height="100%" doesn't take the entire area. When RIV is created we unable to prognoses what work area it should have. Both these parameters are optional and we recommend use them only for Mac platform. If you will not specify these parameters an applet is created with 100% width and height attributes and future behavior will depend on client browser.
- 5. **Open document** – open specified document in RIV. To be able to open a document, RIV should be instantiated and user should be logged on first.
 - a. Parameters:
 - i. Command = 'OpenDocumentByID'
 - ii. ReturnURL – URL of the 3rd party page that will be used by eProof to return results of commands execution
 - iii. DocID – ID of the document in eproof database to open
 - b. Parameters:
 - i. Command = 'OpenDocumentByFileName'

- ii. ReturnURL – URL of the 3rd party page that will be used by eProof to return results of commands execution
 - iii. FileName – Original file path of the document in eproof database to open
- 6. **Remove document** – remove specified job from eProof. To be able to remove a job, user should be logged on first.
 - a. Parameters:
 - i. Command = 'RemoveDocumentByID'
 - ii. ReturnURL – URL of the 3rd party page that will be used by eProof to return results of commands execution
 - iii. DocID – ID of the job in eproof database to remove
 - b. Parameters:
 - i. Command = 'RemoveDocumentByFileName'
 - ii. ReturnURL – URL of the 3rd party page that will be used by eProof to return results of commands execution
 - iii. FileName – Original file path of the job in eproof database to remove
- 7. **Suspend** – disable RIV interface and prevent any pop-up dialogs. Synchronization of corrections and annotations is also stopped. To be able to suspend RIV, it should be instantiated first.
 - a. Parameters:
 - i. Command = 'Suspend'
 - ii. ReturnURL – URL of the 3rd party page that will be used by eProof to return results of commands execution
 - iii. SID – session ID
- 8. **Reload** – reload current document after suspending RIV. RIV interface is enabled. It is possible to reload RIV even if it hasn't been suspended first.
 - a. Parameters:
 - i. Command = 'Reload'
 - ii. ReturnURL – URL of the 3rd party page that will be used by eProof to return results of commands execution
 - iii. SID – session ID
- 9. **Logoff** – abandon current user session. If no user has been logged on, error will not occur.
 - a. Parameters
 - i. Command = 'Logoff'
 - ii. ReturnURL – URL of the 3rd party page that will be used by eProof to return results of commands execution
 - iii. SID – session ID
- 10. **Ping** – keep the application alive.
 - a. Parameters
 - i. Command = 'Ping'
 - ii. ReturnURL – URL of the 3rd party page that will be used by eProof to return results of commands execution

Response interface

This is the HTTP based interface used by eProof to return status of the command execution and RIV state to 3rd party web solution. It has to be implemented by one of 3rd party pages (asp, php, Coldfusion, etc). URL to the page to use is provided to eProof as 'ReturnURL' parameter in every command.

Following parameters are returned as a result of executing the requested command:

1. Result – result code of the command execution
2. State – resulting RIV state
3. SID – this parameter is returned in response on command 'CheckEnvironment', it identifies the custom session used in eProof
4. ExtResult – this optional parameter may be returned in response on command 'CheckEnvironment' in case the environment doesn't meet RIV requirements; this

code specifies what exactly in configuration doesn't meet RIV requirement. Possible values are:

1. Java is not enabled in browser or not installed
 2. Browser not supported
 5. ActiveX controls are not enabled in browser
 7. This Java plug-in version not supported
 8. This Java plug-in version not tested, thus Java errors may occur
 9. MacOS 10.1 not supported
 12. This Java plug-in version is not supported under Safari
 14. Windows NT is not supported
 15. Internet Explorer 5.0 in MacOS9 not supported
5. Version – full server version number. (Ex: 4.2.4129.0). Returned with each response.

Result codes

Result code may have one of the following values:

0. Command succeeded
1. Unexpected error has occurred
2. Client environment is not checked (a command was sent that requires client environment to be already checked)
3. Client environment is not supported
4. User not logged on (a command was sent that requires user to be logged on)
5. Invalid user name
6. RIV not installed (a command was sent that requires RIV to be installed)
7. RIV installation error occurred
8. RIV not created (a command was sent that requires RIV to be created)
9. RIV instantiation error occurred
10. Document is not loaded in RIV (a command was sent that requires a document to be loaded in RIV)
11. Invalid document ID
12. Session has expired
13. Invalid command
14. Invalid command parameter
15. Permission denied
16. No proofs available
17. Invalid License Code
18. Rip failed
19. Document in process

50. Invalid response from RIV server
51. Connection timeout with RIV server

States

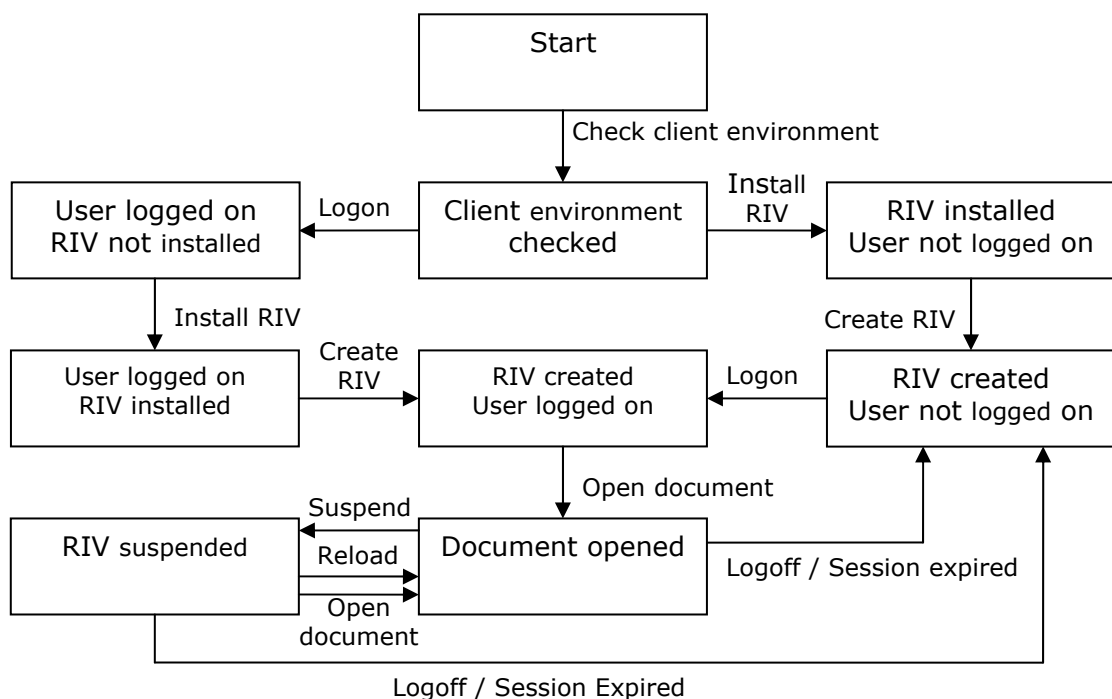
RIV state is represented as an integer that consists of a set of bits, every bit defines whether appropriate condition from the list below is met or not. Note that check mask specified for every state below is in binary format (decimal representation in brackets).

1. **Client environment checked** – if this bit is set, information about client OS, browser and Java virtual machine is detected and checked. This information is used for compatibility check.
Check mask = 00000001 (decimal 1).

2. **User logged on** – if this bit is set, user was authenticated and logged on into eProof.
Check mask = 00000010 (decimal 2).
3. **RIV installed** – if this bit is set, RIV files were delivered by RIV installer to client's PC and RIV can be instantiated.
Check mask = 00000100 (decimal 4).
4. **RIV created** – if this bit is set, RIV is instantiated and the 3rd party pages should enlarge the RIV frame (as it is done in the sample).
Check mask = 00001000 (decimal 8).
5. **Document opened** – if this bit is set, user can use RIV. RIV frame should be enlarged now.
Check mask = 00010000 (decimal 16).
6. **RIV suspended** – in this state RIV user interface is disabled. RIV frame should be minimized now.
Check mask = 00100000 (decimal 32).

States diagram

The diagram below shows possible RIV states and ways of moving between states by executing appropriate commands. Please note that not all possible commands are drawn in the diagram (only the most accurate ones). For example, it is possible to logon being in any state other than 'Start'. Actual prerequisites are specified in controlling interface description.



Sample

The development version of eProof automatically installs a sample page into the "RIVInFrame" folder. E.g. if your local eProof installation has the URL "http://localhost/eProof" the sample page is located at **http://localhost/eProof/RIVInFrame**

The proposed sample demonstrates how the integration could be achieved and how it may look like. Sample may reside on a 3rd party web server different from eProof server. All sample files are used for demonstration purposes only and should be replaced by 3rd party pages and scripts (some functions such as 'ExecuteCommand' may be reused though).

Sample consists of the following files:

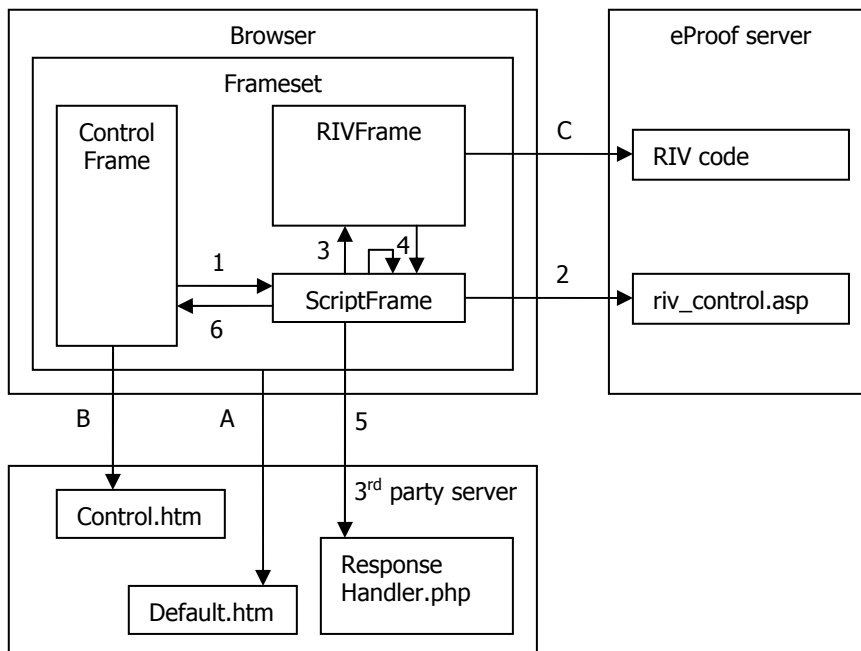
1. **Frameset** (*Default.htm*) – contains frameset of the sample.
2. **Introduction** (*Introduction.htm*) – contains text instructions how to use the sample, contains no scripts, this page is displayed when RIV is hidden.
3. **Control** (*Control.htm*) – contains user interface and Java script that allows to send commands to RIV and process response. In a real integration this page is not necessary and the controlling should be done by 3rd party code.
4. **Response handler** – a very simple ASP/PHP whose only purpose is to receive response from RIV and pass it to appropriate Java script function defined in *Control.htm* for further processing.
 - a. *ResponseHandler.asp* – ASP version of response handler.
 - b. *ResponseHandler.php* – PHP version of response handler.

Frameset of the sample consists of the following frames:

1. **ControlFrame** – control page is loaded in this frame, it is never reloaded.
2. **MainFrame** – introduction page is loaded in this frame, it is never reloaded, it is maximized when RIV is hidden and minimized when RIV is shown.
3. **RIVFrame** – RIV applet code is loaded in this frame from eProof server, it is never reloaded as soon as RIV is created, it is minimized when RIV is hidden and maximized when RIV is shown.
4. **ScriptFrame** – RIV controlling ASP is loaded in this frame from eProof server every time a new command is executed, additionally sample's response handler is loaded here (ASP or PHP) from 3rd party web server. It is always minimized.

Command execution diagram

The diagram below describes control flow of a typical RIV command execution. It assumes that 'Create RIV' command has already been executed.



Prerequisites (displayed with arrows with letters from A to D)

- A. Default.htm is loaded from 3rd party server and opened in browser. Default.htm defines a frameset that consists of frames ControlFrame and RIVFrameset.
- B. Control.htm is loaded from 3rd party server into frame ControlFrame.

- C. RIV applet is loaded from eProof server into frame RIVFrame as a result of previously executed command 'Create RIV'.

Steps below describe what's going on when user clicks one of buttons that perform some command (displayed with arrows with sequence numbers from 1 to 6)

1. Script in ControlFrame forms URL to riv_control.asp on eProof server that includes command parameters in it. This URL is then opened in frame ScriptFrame.
2. Browser loads riv_control.asp
3. Script returned by riv_control.asp interacts with RIV applet to perform the requested command.
4. Either RIV or script from riv_control.asp form URL to ResponseHandler.php on 3rd party server that includes result of executing the command and resulting RIV state. This URL is opened in ScriptFrame.
5. Browser loads ResponseHandler.php
6. Script loaded by ResponseHandler.php calls appropriate response handler in ControlFrame.

Note that control flow of 'Create RIV' command is different. The differences are:

1. riv_control.asp should be executed directly in the RIVFrame.
2. response_handler.php is executed in one of RIV's own subframes (used also for RIV notifications).