# Spectra Logic RioBroker Certification Test Plan

## Introduction

The document describes the certification tests for Spectra Logic RioBroker integration. In order to receive certification from Spectra Logic, this completed document must be submitted to your Spectra Logic Certification and Testing Manager for review. Before submitting this document, you must have submitted and received acceptance from Spectra Logic for the Spectra Logic RioBroker Certification Program Questionnaire. For additional information, see the Spectra Logic Developer Program website at <https://developer.spectralogic.com>.

Testing must be done with a real RioBroker installation using the BlackPearl simulator or a real BlackPearl.

All tests are required to be performed unless stated otherwise.

The remainder of this document assumes that the user is familiar with details of the RioBroker API, the BlackPearl product, and information contained on the Spectra Logic Developer Program website.

## Test Setup, Installation of the Host Client Software, Connection to RioBroker and BlackPearl and Configuration

Once you are ready to begin testing, your Spectra Logic Certification and Testing Manager will work with you to determine the appropriate test setup, test equipment requirements, test time, installation of host software and client, and other testing program details. Review the installation of your product, RioBroker, and BlackPearl connection and configuration to verify that everything is working properly prior to commencing the testing.

Testing can be done at your site or at the Spectra Logic Testing and Certification Lab in Boulder, Colorado.

## Logging, Result Capture, and Live Demo Requirements for Certification

* **Log files:** Please pull a log set from RioBroker when directed in the test plan steps below. These log files are used to assist with confirming proper Interface behavior and debugging any failures that may occur. RioBroker logs can be found in C:\ProgramData\SpectraLogic\SpectraRioBroker\logs.
* **Screen Captures:** Screen captures will be required for the tests to enable Spectra Logic to verify test results and certify your product. Please ensure that the person performing the test understands how to capture screen images and paste them into this document.
* **Live Demo**: Once this test plan, along with test results, has been submitted and accepted by Spectra Logic, Spectra Logic will contact you to schedule a live demonstration of archive and restore operations by your RioBroker integration. This live demonstration can be done at Spectra Logic headquarters or remotely via WebEx or similar screen sharing technology. **You will not receive certification until Spectra Logic concludes that you have passed all tests and that this live demonstration is successful.**

**You must complete the following information in order to receive certification.**

|  |
| --- |
| Organization/Tester Information |
| Organization Name: IPV |
| Tester Name: Richard Schwartfeger |
| Tester Contact Information (email and phone): rschwartfeger@ipv.com +1 778 677 8566 |

|  |
| --- |
| Product and Configuration Information |
| Your Application’s Name: IPV Curator |
| Application Software Version: 3.5.1 |
| Application’s RioBroker Client Software Version (if different than above): 0.0.5 |
| Spectra Logic RioBroker Version: 3.3.0 |
| RioBroker .NET SDK Version (if used): |
| BlackPearl Version: |
|  |

## Testing the RioBroker Integration

To get started with testing, set up the RioBroker system as needed: Install RioBroker, set up a cluster, initialize a Spectra device, create a broker with default agent of the Spectra device, and make sure at least one primary storage target is installed with valid credentials on the RioBroker server.

**Test #1:** Test for error reporting and handling of invalid/missing bucket.

**Goal:** To ensure application properly reports and handles unexpected error conditions on archive, displaying the RioBroker error description.

* Attempt to archive files using RioBroker after BlackPearl bucket has been deleted.
	1. Using your application and RioBroker, create Spectra device, then create associated broker and agent.
	2. Attempt to archive some files, confirm successful transfer.
	3. Delete BlackPearl bucket. In the BlackPearl web management interface, go to Configuration -> Buckets. Single click on the bucket being used by RioBroker, then go to Action -> Delete.
	4. Restart the “Spectra Rio Broker Server” service.
	5. Attempt to archive files again. Capture a screen image that clearly shows your application’s error (this should include the RioBroker error message). Paste the screenshot below:

|  |
| --- |
| Attempt to archive files using RioBroker after BlackPearl bucket has been deleted |
|  |

**After the test is complete recreate the bucket that was deleted in this test.**

**Test #2:** Test for error reporting and handling of a restore when the file does not exist.

**Goal:** To ensure that the client properly reports and handles unexpected error conditions on restore, displaying the RioBroker error description.

* Attempt to restore a file from RioBroker that does not exist
1. Archive one or more files to RioBroker using your application.
2. Delete one of these files using a BlackPearl client such as the Eon Browser (https://developer.spectralogic.com/clients) or delete the file via RioBroker using an HTTP/REST command.
3. Attempt to restore the deleted file through your application.
4. Capture a screen image that shows your application’s error (this should include the RioBroker error message). Paste the screenshot below:

|  |
| --- |
| Attempt to restore a file from RioBroker that does not exist |
|  |

**Test #3:** Test archive of 100 files.

**Goal:** To ensure that your application can perform an archive of multiple files at once. This test verifies that the archive process keeps RioBroker data movers working efficiently, and reasonable throughputs can be achieved.

* Archive 100 files to RioBroker
	1. If testing in the Spectra lab, ingest 100 files from \\10.85.41.34\TestFiles\250x1GB (or another location provided to you by Spectra Logic). If your application does not need to ingest files, they can be archived directly from the above location.
	2. Issue a command to archive the 100 files to RioBroker simultaneously
		+ **Note**:
			- The 100 files should be either in a single command (100 files in one archive command) or multiple separate commands (one or more files per archive command).
			- If the 100 files are sent to RioBroker in separate commands, some level of concurrency must be demonstrated. Please describe how many jobs can be run in parallel and, if changeable, show where the concurrency setting is configured within your application.
	3. Wait until the file transfers have completed before continuing.
	4. Capture a screen image that shows your application indicating that the transfer of the 100 files has been completed. Paste the screenshot below:

|  |
| --- |
| Archive 100 files to RioBroker |
|  |

* 1. Save a copy of the RioBroker log file (rio-main.log), renamed as **Archive100.log**.
	Include this and all other log files with your test results.
		+ **Note**: Rio logs are located on the RioBroker server at C:\ProgramData\SpectraLogic\SpectraRioBroker\logs\

**Test #4:** Test restore of 100 files.

**Goal:** To ensure that your application can perform a restore of multiple files at once. This test verifies that the restore process keeps RioBroker data movers working efficiently, and reasonable throughputs can be achieved.

* Restore 100 files from RioBroker
1. Issue a command in your application to restore the 100 files archived in Test #3 from RioBroker.
	* **Note**:
		+ - The 100 files should be either in a single command (100 files in one restore command) or multiple separate commands (one or more files per restore command).
			- If the restore is sent in separate commands, some level of concurrency must be demonstrated. Please describe how many jobs can be run in parallel and, if changeable, show where the concurrency setting is configured within your application.
2. Wait until the file transfers have completed before continuing.
3. Capture a screen image that shows your application indicating that the restore of the 100 files has been completed. Paste the screenshot below:

|  |
| --- |
| Restore 100 files from RioBroker |
|  |

1. Save a copy of the RioBroker log file (rio-main.log), renamed as **Restore100.log**.
Include this and all other log files with your test results.
	* + **Note**: Rio logs are located on the RioBroker server at C:\ProgramData\SpectraLogic\SpectraRioBroker\logs\

**The following test is required if your application supports archiving and restoring of files into multiple brokers/buckets via RioBroker**

**Test #5:** Test support of multiple brokers/buckets.

**Goal:** To ensure that the application provides the user the ability to archive to multiple independent brokers/buckets. Note that each RioBroker broker has a default agent which represents a single BlackPearl bucket.

* The application supports multiple brokers/buckets
	1. Create two brokers/buckets (if the client supports broker/bucket creation)
		1. Capture a screen image that shows how to create the additional buckets. Paste the screen image into below:

|  |
| --- |
| Multiple buckets creation screenshot |
| N/A |

* 1. Archive data to both buckets
		1. Capture screen images that show the application completed with files in both buckets. Paste the screen images below: OK to use Eon Browser and show the two buckets/files via direct BlackPearl connection.

|  |
| --- |
| Files in both buckets screenshots |
| N/A |

* 1. GET data from both buckets
		1. Capture screen images that show the application completed with files restored from both buckets. Paste the screen images below:

|  |
| --- |
| Files restored from both buckets screenshots |
| N/A |

**The following test is required if your application supports time-based partial file recovery via RioBroker**

**Test #6:** Test Partial File Recovery archive/index and restore/PFR.

**Goal:** To ensure that your application supports the RioBroker PFR system and provides reasonable and clearly understandable messaging to the end user in the case of error conditions.

* If the application supports restore of partial files from BlackPearl/RioBroker
	1. Archive file using RioBroker
		1. Capture a screen image that shows in your application that the file archived to RioBroker and was indexed for PFR. Paste the screen image into below:

|  |
| --- |
| Archived file screenshot |
| N/A |

* + 1. Use an unsupported file (e.g. .txt) and alter the file extension to a format that the indexer will look at (.mxf ,.mov, etc.) and attempt to archive the file with PFR. Show error message below.

|  |
| --- |
| Archived file not indexable screenshot |
| N/A |

* 1. Retrieve partial file
		1. Capture a screen image that shows successful retrieval of partial file that was successfully indexed. Paste the screen image below:

|  |
| --- |
| Partial file restored screenshot |
| N/A |

* + 1. Capture a screen image that shows end user messaging if a PFR restore cannot be performed as well as options provided to the end user (e.g., recover entire file):

|  |
| --- |
| PFR restore failure screenshot |
| N/A |

**The following test is required if your application supports importing foreign LTFS tapes via RioBroker and BlackPearl.**

**Test #7:** Test support of importing external files from RioBroker into your application (such as LTFS Foreign Tape imported into BlackPearl).

**Goal:** To ensure that your application can allow import of new files via RioBroker from foreign sources such as LTFS tapes. RioBroker can index the files on these tapes, but your application must then add these files to its database (if media asset management or similar application). Since any foreign LTFS tapes will be imported as read only, the suggested process is to import them into a new bucket, add a read only agent pointing to that bucket and let RioBroker scan the tape. More can be added later but if the agent already exists, then RioBroker must be given a re-index command and your application must then scan for differences. Whatever the method the application supports, this process is intended to test that imports work, that your application issues the proper commands to add the agent or index the agent and that your database is then updated so the files can be used by the client.

* Your application must support some form of foreign file import functionality
	1. Import single tape
		1. Import a single foreign tape containing files into a bucket using the BlackPearl web management interface (Status > Tape Management, then Action > Import all Foreign LTFS Tapes.
		2. Verify the tape’s State changes from “Foreign” to “Managed” on BlackPearl web management interface
		3. Verify that the application can be commanded to re-index the agent/bucket and that the number and size of recovered files/files matches expected contents of the tape. Capture screen images that show before and after status of files in application. Paste the screen image into below:

|  |
| --- |
| Foreign files imported into your application screenshot |
| N/A |

**This test is required if your application supports the displaying/listing of files that are in RioBroker**

**Test #8:** Test application ability to list files in a bucket/agent and optionally perform search.

**Goal:** To ensure that the application can perform basic list operations with reasonable pagination and can perform directed and generic search across agents.

* If application can list files in a bucket, show that it can handle (paginate) a large list
	1. Archive 500 or more files to RioBroker (can use files from prior tests above)
	2. Capture a screen image from the application showing user interface with files displayed on the first page. Paste the screen image below:

|  |
| --- |
| First page of files screenshot |
| N/A |

* 1. Display the second page of files in the application.
	2. Capture a screen image from the application showing user interface with files displayed on a second page. Paste the screen image below:

|  |
| --- |
| Second page of files screenshot |
| N/A |

* If application supports file search:
	1. Put an assortment of files in various buckets/agents
	2. Verify that the application can direct search to a specific bucket/agent and paste the screen image below:

|  |
| --- |
| Search a bucket screenshot |
| N/A |

* 1. Verify that the application can search across buckets and display the results to the user allowing actions on the results and paste the screen image below:

|  |
| --- |
| Search multiple buckets screenshot |
| N/A |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Changes** |
| 0.1 | 1/22/19 | DF | Initial format and content.  |
| 0.2 | 5/23/19 | BS | Updated test case 1 to show the correct procedure for changing BlackPearl credentials. |
| 0.3 | 6/4/19 | BS | Updated test cases 3 – 6 to make transfer speeds recommended rather than required. |
| 0.4 | 11/6/19 | EL | Combined test cases 1 – 7 into four simplified tests, focused on demonstrating error handling and job creation performance for both archive and restore operations. Optional tests are now numbered 5 – 8. |
| 0.5 | 3/31/2020 | EL | Updated tests 3 and 4 (archive and restore of 100 files) to require some level of concurrency if files are in separate jobs. |
| 0.6 | 7/15/2020 | ZG | Updated test 1 to require deleting a bucket on BlackPearl rather than changing S3 credentials. |