Abstract

This document provides a high-level description of known issues, fixed issues and new features in RHMAP 4.7.0.
Table of Contents

CHAPTER 1. RELEASE ANNOUNCEMENT .......................................................... 3
  1.1. NODE.JS 8 AND 10 ................................................................. 3
  1.2. TEMPLATE REMOVAL ............................................................ 3
  1.3. NATIVE IOS (SWIFT) SDK UPDATE ........................................... 3

CHAPTER 2. DEPRECATION NOTICE .......................................................... 5
  2.1. NODE.JS 6 ............................................................................ 5
  2.2. BUILD SERVICE ...................................................................... 5

CHAPTER 3. KNOWN ISSUES ..................................................................... 6
  3.1. ISSUE WITH 4.7.0 AND 4.7.1 RPM WHEN UPGRADING FROM 4.6 .... 6
  3.2. ISSUE WHEN BUILDING IOS APPS ......................................... 6
  3.3. MISLEADING ERROR MESSAGES IN SPECIFIC SCENARIOS .... 6
  3.4. ISSUE WHEN DEPLOYING TO MULTIPLE ENVIRONMENTS ...... 6
  3.5. ISSUE WITH MAIL NOTIFICATIONS RELATING TO FORM SUBMISSIONS .................................................. 6
  3.6. ISSUE WITH FORMS USING CHECK BOXES .............................. 6
  3.7. $FH.DB ISSUE ........................................................................ 7
  3.8. ISSUES WHEN FILTERING DATA USING THE _ID COLUMN ...... 7
  3.9. UPGRADING SWIFT SDK FOR XCODE ................................... 7
  3.10. STUDIO UNRESPONSIVE ....................................................... 7
  3.11. SORTING OF ITEMS IN STUDIO ............................................ 7
  3.12. FORM-BASED EMAIL NOTIFICATIONS ................................ 7
  3.13. ERROR MESSAGES IF AN MBAAS IS UNAVAILABLE .............. 7
  3.14. EXPORT BUTTON .................................................................. 8
  3.15. JENKINSFILE IN TEMPLATES ............................................... 8
  3.16. DATA BROWSER EXPORT .................................................... 8
  3.17. DATA BROWSER IMPORT ...................................................... 8
  3.18. DELETING PROJECTS ........................................................... 8
  3.19. SYNC FRAMEWORK STORAGE ISSUES ................................ 8
  3.20. SYNC FRAMEWORK ............................................................. 8

CHAPTER 4. $FH.SYNC EXAMPLES FOR IOS SWIFT ................................. 10

CHAPTER 5. UPGRADING NODE.JS FOR CLOUD APPS AND MBAAS SERVICES USING RHMAP STUDIO .................................................. 11
CHAPTER 1. RELEASE ANNOUNCEMENT

NOTE
RHMAP 4.7.0 supports OpenShift versions 3.7, 3.9, and 3.10.

1.1. NODE.JS 8 AND 10

With this release, Node.js 8 and 10 are supported by Red Hat Mobile Application Platform 4.7. Red Hat recommends that you upgrade to either Node.js 8 or 10.

NOTE
To use Node.js 8 or 10, update fh-mbaas-api to 9.0.4 or later. For more information, see Upgrading Node.js for Cloud Apps and MBaaS services using RHMAP Studio.

1.2. TEMPLATE REMOVAL

A number of Templates were removed from RHMAP 4.7, however the following Templates are still available in RHMAP 4.7:

- iOS Hello World Project.
- Android Hello World Project.
- Cordova Hello World Project.
- Empty Project.
- Forms Project.

NOTE
Each of the Templates have an associated Cloud App apart from the Empty Project Template.

1.3. NATIVE IOS (SWIFT) SDK UPDATE

With version 6.1 of the iOS SDK, you can change the config file name using the initWithCustomConfigName method. All the same constraints that apply to init method apply to this method. It is advised that you only call a single init method.

This procedure supplements the iOS Initialization procedure:

1. Set the config as follows:

   ```swift
   config = FeedHenry.Config.init(propertiesFile: "dev/fhconfig.plist")
   ```

2. Initialize the SDK:

   ```swift
   FH.initWithCustomConfigName(config: config!, completionHandler: {(resp: Response, error: NSError?) -> Void in
   if let error = error
   ```
{ self.statusLabel.text = "FH init in error" print("Error: \(error)") return }
self.statusLabel.text = "FH init successful"

FH.cloud("hello", completionHandler: { (response: Response, error: NSError?) -> Void in
if let error = error
    print("Error: \(error)") return

    print("Response from Cloud Call: (response.parsedResponse)")
})

print("Response: (resp.parsedResponse)")
}
CHAPTER 2. DEPRECATION NOTICE

2.1. NODE.JS 6

Support for Node.js 6 is scheduled to be removed from RHMAP in a release scheduled for early 2019.

2.2. BUILD SERVICE

The existing RHMAP Build Service ('Build Farm') is deprecated. Building client binaries directly from RHMAP will no longer be supported by Red Hat in an upcoming release of RHMAP.

Build Farm documentation has been removed from RHMAP 4.7. There are links to existing Buildfarm documentation where it was necessary to retain the integrity of the documentation.
CHAPTER 3. KNOWN ISSUES

3.1. ISSUE WITH 4.7.0 AND 4.7.1 RPM WHEN UPGRADING FROM 4.6

If the RHMAP 4.7.0 or RHMAP 4.7.1 rpm is used to upgrade from RHMAP 4.6, all currently running
Node.js 6 apps are rebuilt and redeployed with the new underlying Node.js 6 image. This is due to the
Node.js 6 update to the imagestream. Use the RHMAP 4.7.2 rpm to resolve this issue.

3.2. ISSUE WHEN BUILDING IOS APPS

Building iOS Apps using the default Xcode 10 settings fails. To workaround this issue, set the project to
use the Legacy Build System as described in the Xcode 10 Release Notes. Red Hat recommends using
Xcode 9 where the default target is iOS 11.

3.3. MISLEADING ERROR MESSAGES IN SPECIFIC SCENARIOS

If your application requires a CORS preflight check and an error occurs, the text associated with that
error is not displayed. For example, if you disable a connection tag in RHMAP Studio for an iOS app, and
provide “Disabled Status Text”, that text is never displayed. Instead, the WKWebView library performs a
CORS preflight check, encounters an error and displays a generic JSONP error.

3.4. ISSUE WHEN DEPLOYING TO MULTIPLE ENVIRONMENTS

An issue exists where Cloud App deployments might fail if deployed to multiple environments that are
configured to use different MBaaS targets.

If you encounter this issue:

1. Navigate to the Deploy Screen.
2. Make sure the Clean Stage checkbox is enabled.
3. Click Deploy Cloud App.

3.5. ISSUE WITH MAIL NOTIFICATIONS RELATING TO FORM SUBMISSIONS

If your OpenShift cluster is behind an outbound HTTP proxy, mail notifications relating to form
submissions fail.

To work around this issue, configure your firewall as follows:

1. Determine the SMTP_SERVER and SMTP_PORT environment variables in the millicore
deployment configuration of the RHMAP Core project:
   `oc env dc/millicore --list -n rhmap-core | grep SMTP_SERVER`
   `oc env dc/millicore --list -n rhmap-core | grep SMTP_PORT`
2. Add a firewall rule to allow outbound traffic to allow traffic from the IP address and port
determined in Step 1.

3.6. ISSUE WITH FORMS USING CHECK BOXES
If you click submit on a draft or review form, sometimes a check box selection error is displayed, despite the correct selection of check boxes.

To work around this issue:

1. Open the submission from the Review menu.
2. Unselect the previously selected check boxes, leaving unselected check boxes unchanged.
3. Click Submit.

### 3.7. $FH.DB ISSUE

$fh.db is a Cloud API that supports CRUDL (create, read, update, delete, list). However there is an issue with the list operation if you are using a 8.x version of fh-mbaas-api earlier than version 8.0.3. Red Hat recommends that you update fh-mbaas-api to 9.0.4 or later. For more information, see [Upgrading Node.js for Cloud Apps and MBaaS services using RHMAP Studio](#).

### 3.8. ISSUES WHEN FILTERING DATA USING THE _ID COLUMN

The Data Browser allows you create filters to query data. However, if you create a filter using the _id column, incorrect results are displayed. There is no workaround for this issue.

### 3.9. UPGRADING SWIFT SDK FOR XCODE

If you want to develop apps for iOS with Swift using XCode 8.3 or later, you must upgrade the iOS Swift SDK in order to build those apps using RHMAP. To download the latest SDK, see [fh-ios-swift-sdk](#).

### 3.10. STUDIO UNRESPONSIVE

Very large files or repos may cause your browser to become unresponsive when accessing the Studio editor.

### 3.11. SORTING OF ITEMS IN STUDIO

The lists for Projects and Environments in the Studio are not sorted based on creation time. There is no workaround for this issue at the moment. Studio allows you to sort items, for example, you can view a list of Projects sorted by last update. However, only the current page is sorted, if you click the next page, the sorting is lost. There is no workaround for this issue at the moment.

### 3.12. FORM-BASED EMAIL NOTIFICATIONS

RHMAP includes a feature that allows you to create a form that triggers an email after each submission. One of the email options is to include the app name, however the emails sent from RHMAP do not include the app name. There is no known workaround for this issue at this time.

### 3.13. ERROR MESSAGES IF AN MBAAS IS UNAVAILABLE

If you perform an action that applies to all environments, but one or more environments is associated with an MBaaS that is currently unavailable, you may see messages that are unhelpful. For example, if you delete a form for a project that is associated with an environment MBaaS that is unstable might result in a 401 error. To work around this problem, ensure all MBaaS instances are available.
3.14. EXPORT BUTTON

The Data Browser Export button is incorrectly presented to user as being disabled. However, clicking the button will export data.

3.15. JENKINSFILE IN TEMPLATES

Every Client App template includes a Jenkinsfile in the root directory of the template repo. This file is intended for the self-managed Build Farm and does not provide any functionality for hosted RHMAP.

3.16. DATA BROWSER EXPORT

If you try to export BSON data as described in the Server-side Developer Guide, the export fails. To workaround this issue, export data as JSON or CSV.

If you try to export data from the Data Browser, the export may fail due to a database version issue. To workaround this issue, upgrade the database and try again.

To upgrade the database:

1. In the Data Browser section of the Cloud App page in Studio, click the Upgrade Database button in the top right corner, and confirm by clicking Upgrade Now. Wait until the upgrade process finishes.

2. Redeploy the Cloud App by clicking Deploy Cloud App in the Deploy section.

3.17. DATA BROWSER IMPORT

If you try to import data as described in Server Side Developer’s Guide, the import fails. There is no workaround for this issue at the moment.

3.18. DELETING PROJECTS

If you enter a name containing special characters when creating a project, and you later decide to delete that project, you are prompted to enter the HTML entity code instead of the special character. For example, if you create a project named Tom’s App, and you choose to delete it, you are prompted to enter the following:

Tom&amp;#x27;s App

To delete the project, enter the text, including the HTML entity code, as prompted.

3.19. SYNC FRAMEWORK STORAGE ISSUES

If you are using the Sync Framework, note that there is a known issue with iOS apps that use webkit-sqlite for storage. After the app is running for longer than a few minutes, you might encounter storage failures. Use indexDB for storage instead.

3.20. SYNC FRAMEWORK

The RHMAP data synchronization framework allows you create mobile apps that provide data sync capabilities as described in the Server-side Developer Guide. However, due to limitations imposed by
Apple and iOS, your iOS background apps might not synchronize data as expected. Although it might be possible for a developer to workaround the limitations, Red Hat does not support the framework for background apps, as it is beyond the intended scope of the framework.
The Swift sync examples documented in the Client API Guide are specific to Swift 3. If you are using Swift 4, autocorrect the examples using the Fix button in Xcode before compiling your app.
CHAPTER 5. UPGRADING NODE.JS FOR CLOUD APPS AND MBAAS SERVICES USING RHMAP STUDIO

This procedure describes how to upgrade the Node.js version in a Cloud App and MBaaS service using RHMAP Studio.

Prerequisites

- Access to RHMAP Studio.
- RHMAP Studio login credentials.

Procedure

1. Log in to RHMAP Studio.
2. Navigate to the Editor screen for your Cloud App or MBaaS service in RHMAP Studio.
3. Click package.json to display the contents.
4. Update the fh-mbaas-api value to 9.0.4.
5. Click File, then Save.
6. Test the Cloud App or MBaaS service before deploying to Production.

NOTE

When deploying a Cloud App or MBaaS service, select Node.js 8 or Node.js 10 from the Choose a Runtime dropdown in RHMAP Studio.

Additional resources

- For more information about deploying, see Deploy.