



# A Comprehensive Guide to MuleSoft Mule 4



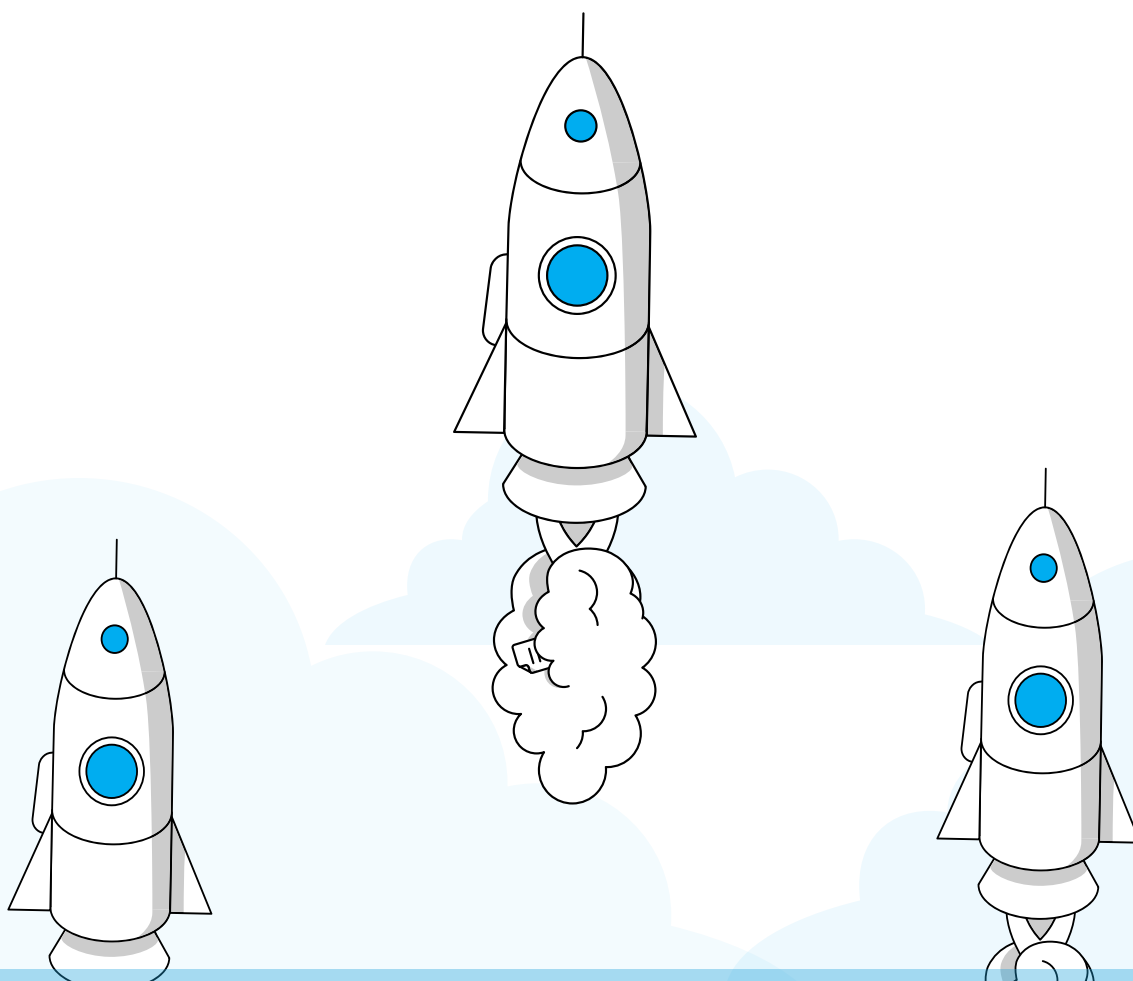
**From New  
Features to Migration**



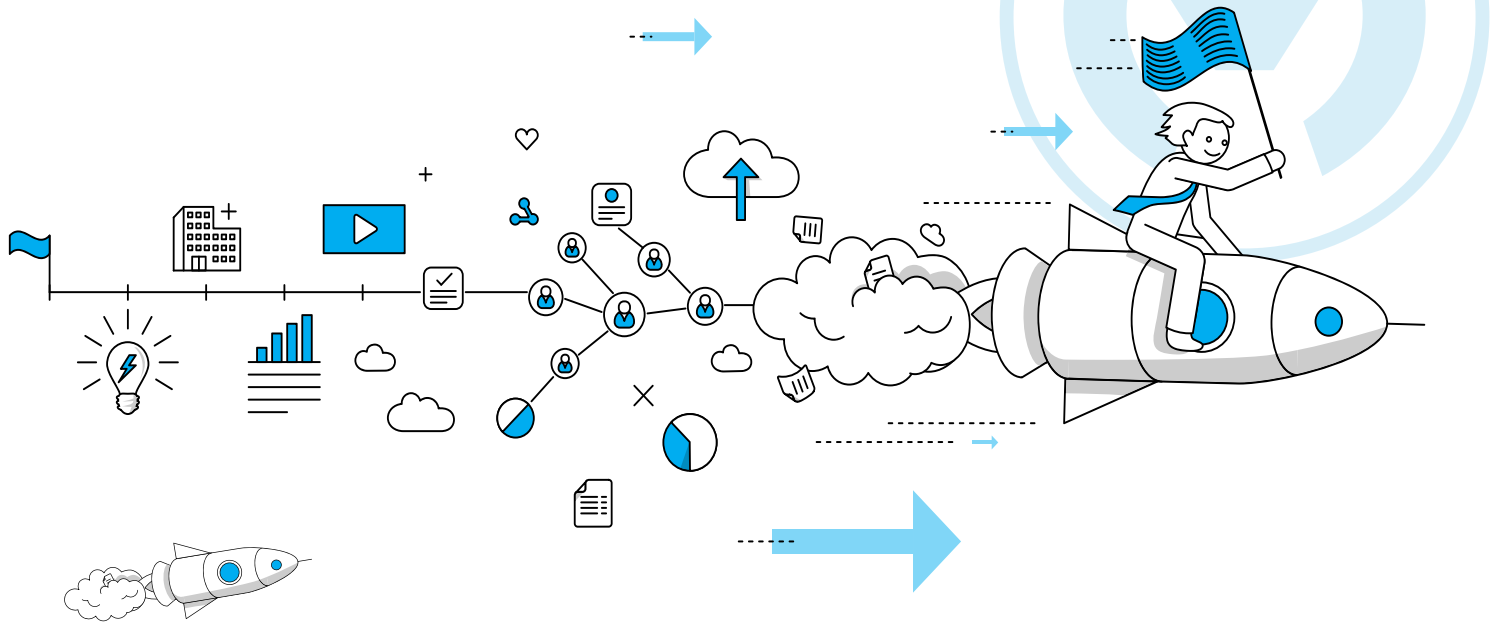
# CONTENTS



<b>Introduction</b> .....	<b>1</b>
<b>Mulesoft's Mule 4 vs Mule 3: What's New</b> .....	<b>3</b>
<b>Mule 3 to Mule 4 Migration Process</b> .....	<b>10</b>
• <b>Measures to Understand Prior to Migrating Process</b> ...	<b>10</b>
• <b>Deciding When to Migrate from Mule 3 to Mule 4</b> .....	<b>13</b>
• <b>Conducting the Mule 3 to Mule 4 Migration Process</b> ...	<b>13</b>
<b>References</b> .....	<b>16</b>



# Introduction



For businesses to stay relevant and competitive in this globalized world, digital transformation is imperative. Also, many businesses have disparate systems and applications. Based on this, the need for integration of these disparate systems is required. The key to implementing a system integration process requires a precise understanding of when to look for software integration. Here, we assist you with situation that necessitate the need for software integration solutions.

APIs bind the data, processes and business systems and enables the outside world to access the systems and data through APIs. These APIs act as a governing authority to the business data, systems, processes and user interface. To access an application or server systems, APIs are used the same as processes and apps. This way structuring APIs enables agile businesses to integrate with any number of outside applications easily. Business systems can have better, more secure connectivity with endpoint applications or devices at any given time.

Especially while dealing with legacy systems, APIs prove to be effective in utilizing the functionalities in a secure manner without having to worry about system configurability on the other end. Businesses don't have to replace legacy systems in order to gain digital transformation benefits. This makes modernizing business systems easy. APIs can also help two or more distinct platforms communicate securely. And they have hundreds of other uses. Joe McKendrick of Forbes notes that creating an API strategy adds functionality, increases agility and decreases operating costs.

MuleSoft is a powerful partner because it enables enterprises with robust API integration solutions and strategies. With MuleSoft, we can create new programs with open-source technology. MuleSoft implementation allows companies to create an IT Department with new business opportunities, rather than leaving it as a slow cost center. MuleSoft can help your company integrate, connect and build its enterprise in surprising ways.

Mulesoft's latest launch, Mule 4, is offering interesting integration solutions, more simple and less expensive than the Mule 3 release. If you are considering migrating from Mule 3 to Mule 4, here we discuss the differences between the two releases and why migrating to Mule 4 is more beneficial.





# Mulesoft's Mule 4 vs Mule 3: What's New



## Integrated and Effective Error Handling



Unlike Mule 3, exception handling in Mule 4 is not Java styled exceptions. The Mule 4 error handling framework directly handles the exception with an integrated and configurable error handling mechanism. Instead of completely avoiding the Java exceptions handling mechanism, Mule 4 abstracts Java exception objects right into the Mule 4 error objects. Java exceptions validators are used which include:

1. Description – a string
2. ErrorType – an object
3. Cause – the underlying Java throwable objects that caused the failure



The errors are located and handled based on type during the design phase. The problem with the Mule 3 approach is it's mandatory to go through the complete code to understand, either to plan the error handling or to locate the cause of any execution failure. With Mule 4, the programmers can pre-define the complete error handling block to configure better. A well-defined error handling block includes:

1. Description: Designates issue.
2. Type: Exemplifies problem. Namespace and identifier indicate the type of error. Each error type has a parent and by default, ANY is the parent type for any error.
3. Cause: Indicates the cause that triggered the execution failure.
4. Error Message: Communicates and makes the error message clear.

Whenever an error occurs the natural Mule execution flow halts and an event is raised and passed to the error handler. The error handler component can contain any number of internal handler scopes defined within it. Each internal error handler scope can have many event processors. On-error-continue and on-error-propagate are the internal error handlers. As the event is passed to each error handler, the appropriate internal error handler is identified and directed to it. The detailed flow of user-defined error handler is illustrated in the figure below:

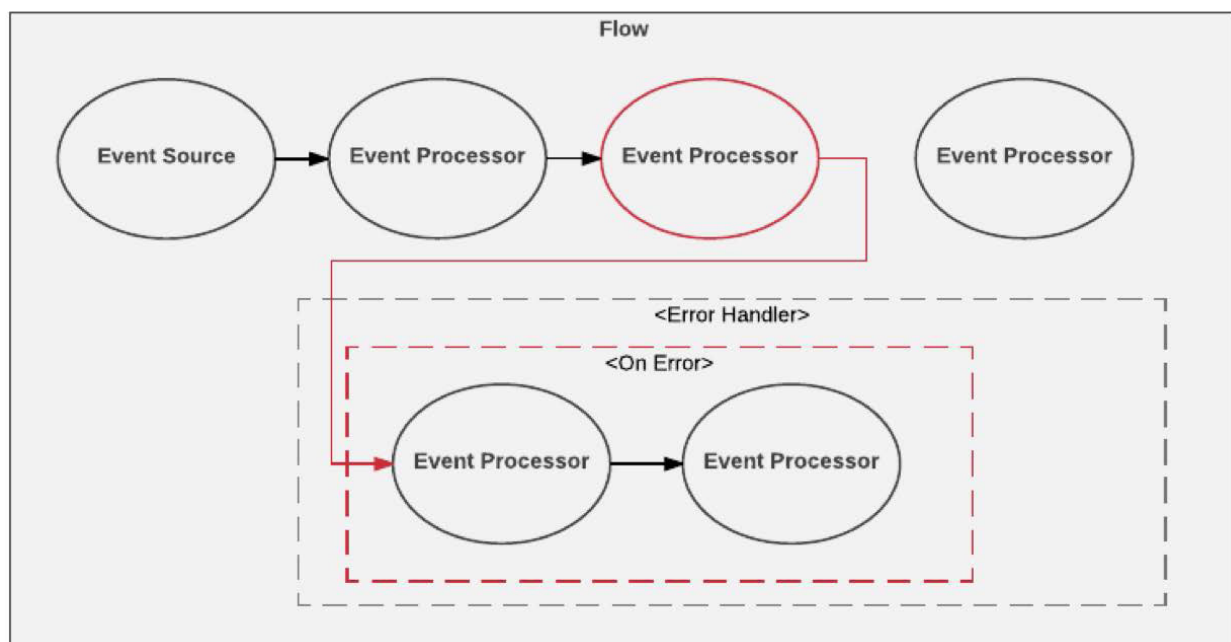


Figure: User-Defined Error Handling Flow



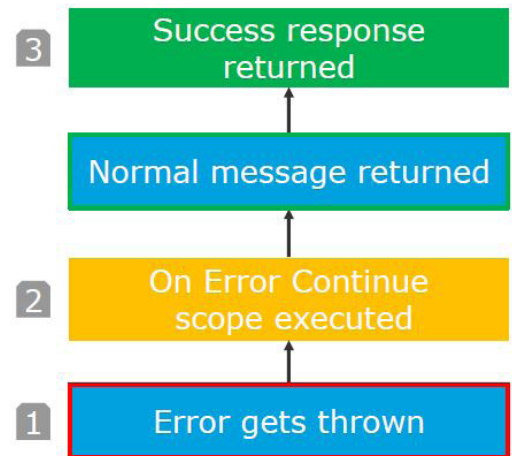
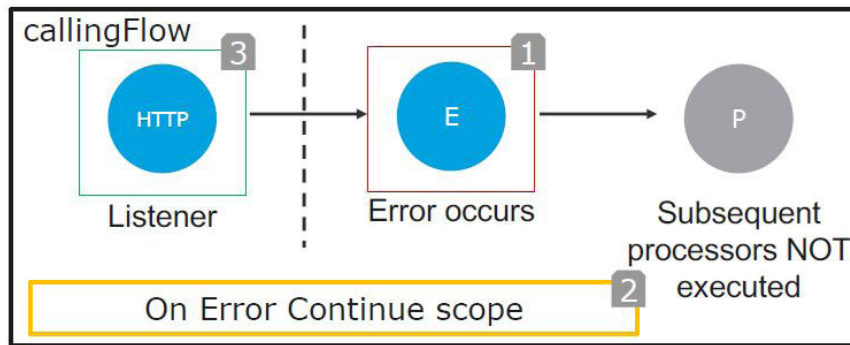


Figure: Flow of On Error Continue

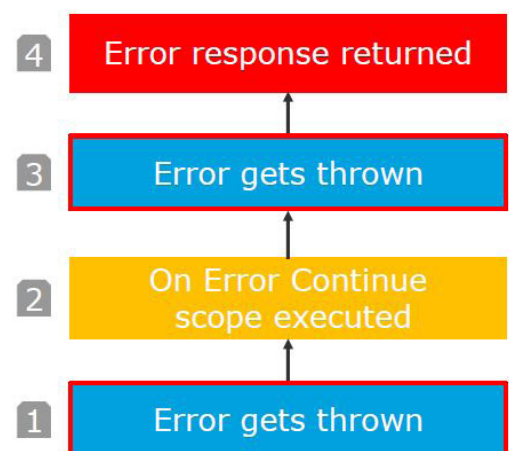
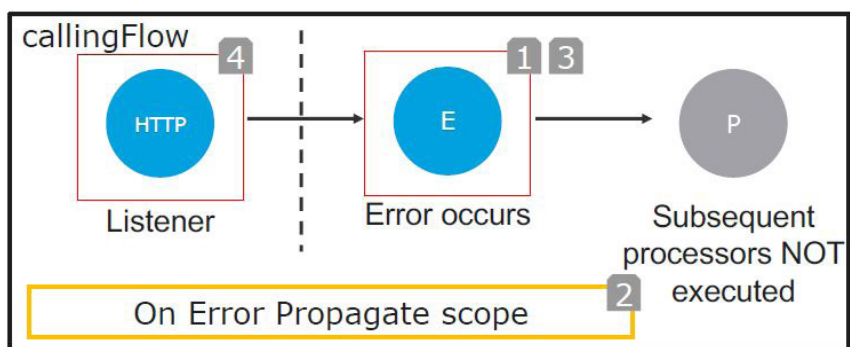


Figure: Flow of On Error Propagate Scope

## Default Error Handling

If the error handler is undefined, inherently the default Mule 4 error handler takes care of error handling, offering no configurability options. To increase the visibility of the error it is suggested to define the error handling block.



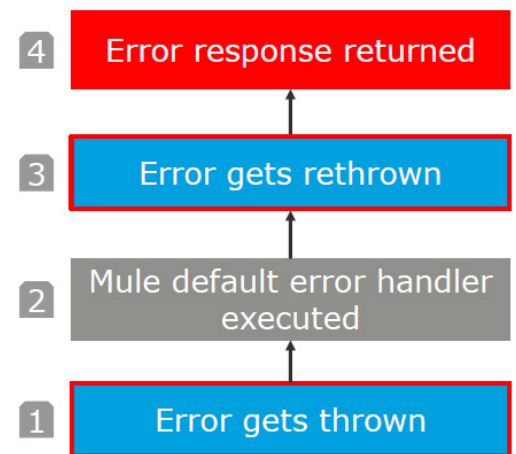
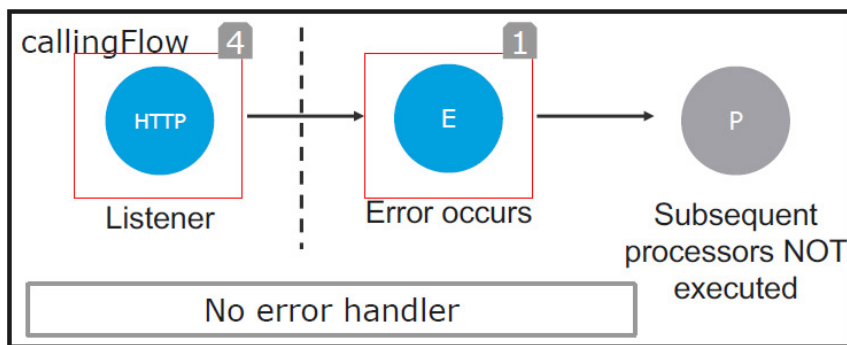


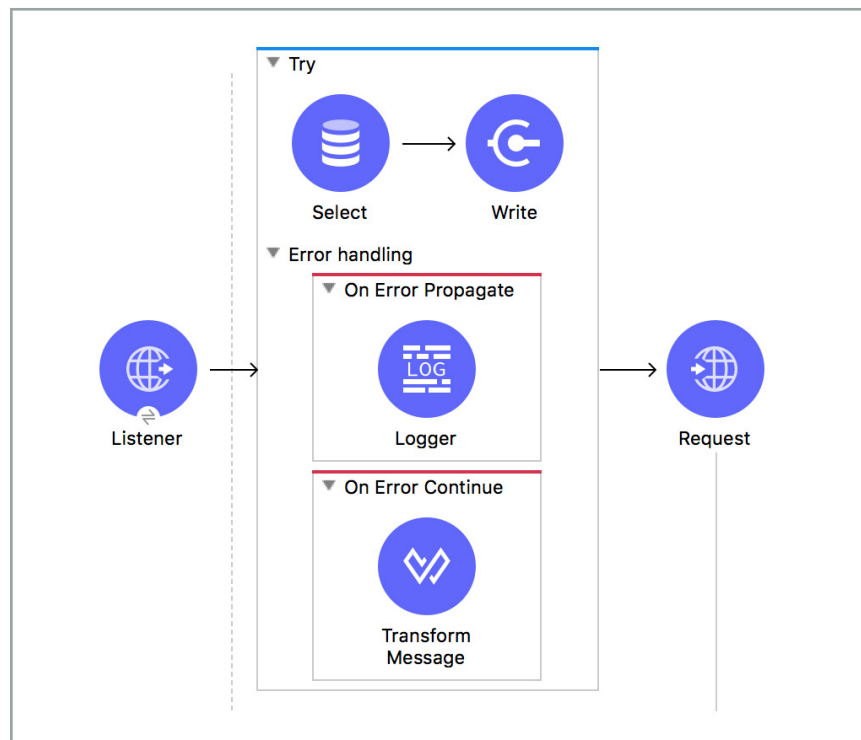
Figure: Flow of default error handler processes

## Try

Mule 4 has introduced “Try” to catch any number of event processes during the flow itself, thereby avoiding the need for creation of another flow. This way Mule 4 ensures uninterrupted and smooth processing. The beauty of “Try” scope is it envelops a number of error event processors prior to handling any exceptions. As the error handling strategy is more inline, this eliminates the need to define multiple new error flows for each error event process. Since there is no need to extract each error flow separately, the error handling flow cost and time are optimized.







## To Summarize

- Mule 4 abstracts the Java exception into Mule 4 error objects
- A hierarchy can be used to catch groups of error objects together in Mule 4
- Errors can be mapped with a custom Error object to differentiate errors in different locations of the application
- Try scope allows micro-control of message processing inside the flow

## Better Application Configurability: Completely Maven

To improve the configuring and management of the application development process, Mule 4 comes with profound integration with Maven by ensuring all Mule 4 applications are Maven applications by default (Mule 3 only offers the option to build Maven projects). Mulesoft has changed the overall application structure of Mule 4.

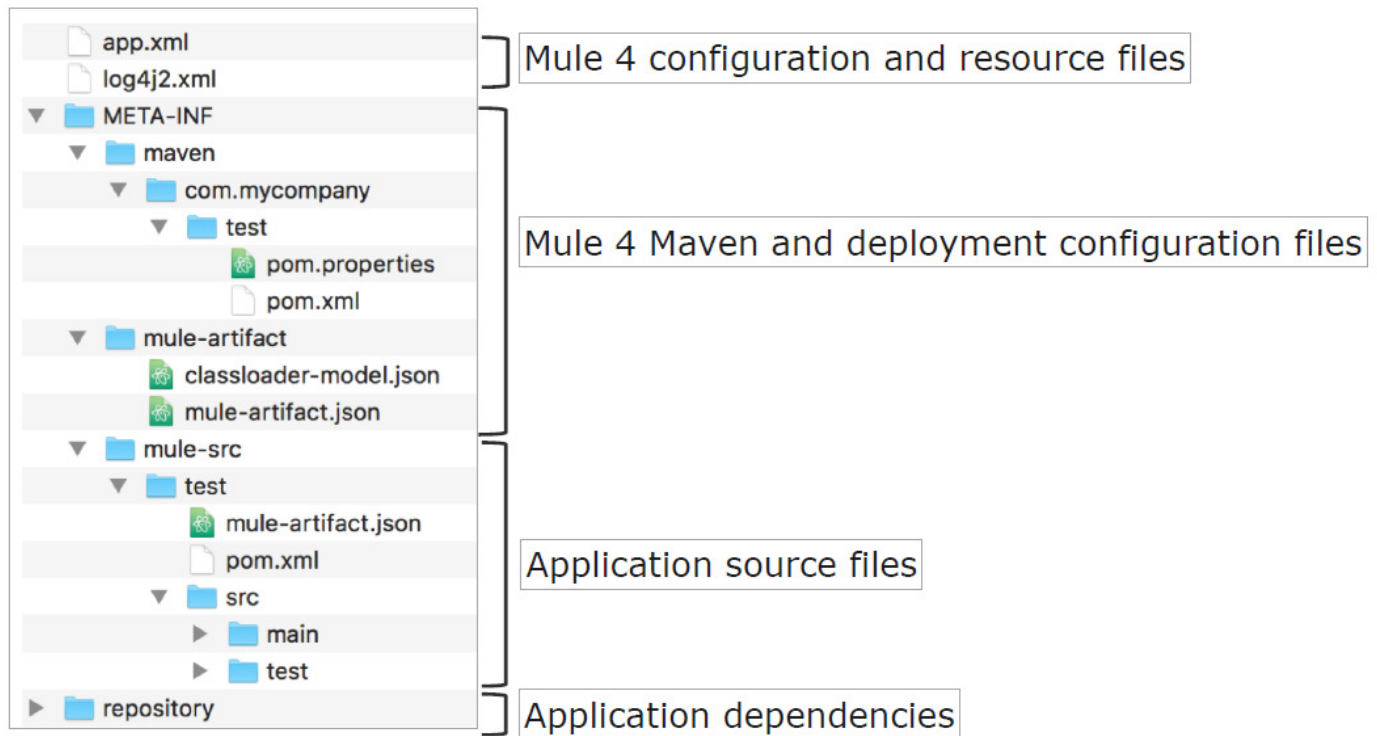


Figure: Mule 4 Application Structure. Source: MuleSoft's Mule 4 Documentation

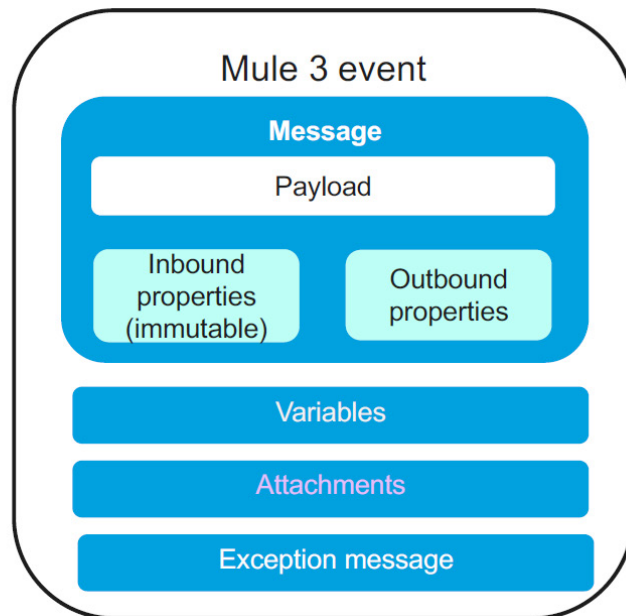
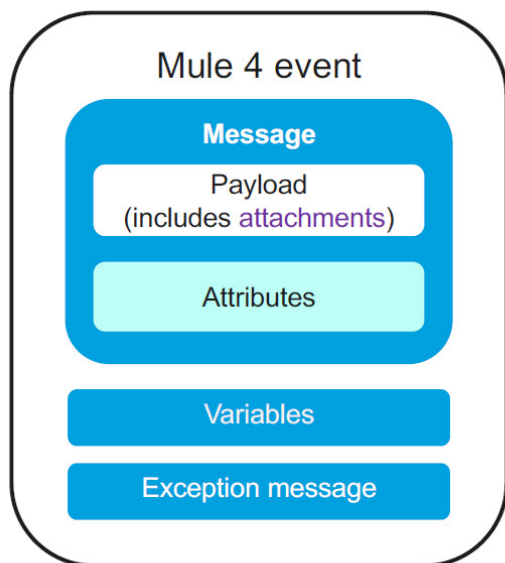
## Simplified Event Processing and Messaging



Mule 4 brings a more compact event processing model by optimizing unwanted hierarchies and workflows. In the message section of the Mule 4 event architecture, the inbound and outbound properties in Mule 3 are merged under one category — “Attributes” — which holds the payload’s metadata information, like any file content and updates on files, query parameters, flow’s message source, inbound properties, outbound properties, and message processor info.

Unlike in Mule 3, Mule 4 payload allows piggybacking attachments to optimize the flow. In Mule 4, event creation happens whenever changes are made to Mule events. This avoids data discrepancies across all the running threads or other events relying on this event.

In Mule 3, a message is embedded within the Mule message objects (contains: variable, attachment and exception payloads) and metadata holds info about the message. As a part of message passing, messages are to be transformed explicitly into java objects, whereas in Mule 4 it happens by default.



## DataWeave 2.0 Improves Data Handling

The DataWeave module, powered by the DataWeave data transformation language, is a full-featured and fully native framework for querying and transforming data on Anypoint Platform. DataWeave's language delivers up to 5x performance against comparable approaches.

*Fully integrated with the graphical interface of Anypoint Studio and DataSense, DataWeave makes even the most complex data integration simple.*

*– The MuleSoft Team*

In Mule 3 releases, developers use Mule Expression Language (MEL) as well as DataWeave for developing mule messages. This approach had some data inconsistencies and scattered approaches. To streamline data activities, DataWave 2.0 was launched. DataWeave 2.0 pushes Mule 4 messages right into a connector, rather than using MEL. To have greater data transparency, Mule 4 stores the event structures and responses (data and context).

In contrast to Mule 3, Mule 4 DataWeave avoids the stress of converting data objects to Java objects by the usage of expressions. The DataWeave avoids caching of data in memory while providing access to memory or any data repositories. This way data is streamed fast, transparently and avoiding unnecessary memory lags.

# Mule 3 to Mule 4 Migration Process



MuleSoft's latest launch Mule 4 offers some interesting integration solutions for businesses at a low cost, compared to Mule 3. Migrating from Mule 3 to Mule 4 should be carried out with the utmost care. To assist businesses in secured migration process, here we detail on how to conduct a migration from Mule 3 to Mule 4.

## Measures to Understand Prior to Migrating Process

Companies are modernizing their legacy applications (also known as a migration process). While conducting the migration process, companies are failing to consider the robust security aspects, which are very important. Here are key security measures an organization should not miss when performing a legacy migration.

### Data Sensitivity

Data is crucial in any type of legacy migration process. During the data migration, extreme care must be taken, as data blocks may be missed or deleted.

Consider an example of a banking application migration process, where customer data is critical. In this application, data is abstracted at various levels and based on the authorization data access is given. Data viewed by the bank manager and the customer may not be the same, as the bank manager has more authority to view sensitive information than a customer.



*“Data migration typically takes 30 to 40% of the effort in any new application project”  
- Upplex Technologies*

During the migration process, due to poor data migration practices, customer data may get deleted or stored in the wrong place. Data can also be damaged due to changes in data security levels. This will result in access to data by unauthorized persons, i.e. the sensitive information normally only seen by the manager is now visible to the customer. If the loss is pertaining to critical business data, the company may face huge losses.

## Encryption and Decryption Issues

To ensure secure access to data, the storage and retrieval process involves encryption and decryption. This involves using customized hybrid encryption algorithms to ensure maximum security. The migration of such encrypted databases, if not properly done, can clutter up the data and ultimately lead to data corruption and loss. This may also lead to security vulnerabilities in the migrated system.

## Usage of Primitive Tools or Processes

In the haste to migrate, companies often employ primitive tools, which lead to building a weak system that will fall prey to hackers.

For example, consider the issue of an auto update. ServiceNow's upgrade Helsinki is going to expire soon. If you are still using Helsinki, ServiceNow will first send a notification and later will auto upgrade to the latest version, as per instance types and its upgrade dates. If the ServiceNow auto-upgrade uses a primitive process, this situation would result in many problems, such as breaking the existing environment customization, functionality failures, lost instances and depreciation of existing plugins. To ensure a smooth migration to the new version, you must define well-structured processes and use robust tools that are functionally specific.





## Insufficiencies In the Systems Framework

“To err is human” is the saying, and it applies to the technology created by humans too. Every technology comes with some security flaws. Legacy systems with older technologies have a different set of problems and the migration may bring in new problems. This would result in new bugs arising, due to a lack of understanding the insufficiencies of the older framework. So, prior to performing the migration, get a detailed understanding of the systems framework and plan interfaces to conduct smooth migration.

## Less Security Testing Measures

The legacy migration process used entails testing, either after building the system or when a vulnerability arises. This is a bad practice. Instead, assess the probable bugs and use these as a guide in assessing threats. The best practice is to develop an Automated Test Framework, which acts as a guidepost in the entire testing process to check and fix vulnerabilities on a timely basis.

## Lack of Stringent Security Policies

Most organizations fail miserably in getting their security measures right due to lack of a concise security policy. This is also known as a security road map. This security policy defines the direction and actions the migration performing teams need to abide by. This guide helps with understanding and handling data and provides a clear view on current and evolving threats.



## Deciding When to Migrate from Mule 3 to Mule 4



Prior to opting into the Mule 3 to Mule 4 migration process, it is important to recheck organizational readiness. Consider these points in your decision-making process:

- Are MuleSoft experts fully trained in using Mule 4?
- Do you understand the components changing from Mule 3?  
(Learn Mule 3 Vs Mule 4)
- Is your organization willing to keep all MuleSoft supported applications up to date with Mule 4 upgrades?
- Is Mule 3 and its variants license expiring?
- Do customers/clients want to upgrade all their applications to Mule 4?
- To deploy Mule 4 manually, it is important to get deployment environment set up. To do so, getting Anypoint Studio 7 installed is must. In this case, the deployment must be planned in a local environment that has Mule 4 standalone runtime.

## Conducting the Mule 3 to Mule 4 Migration Process



Though the migration process can be done manually, it is advised to use the MuleSoft migration tool. To kick start the migration, all the modules should be added to the Anypoint Studio 7 palette. The sequence to follow when migrating is as follows:



## 1. Migrate Patterns

- 1.1. Migrate message properties
- 1.2. Migration re-connection strategies

## 2. Migrate Secure Properties Placeholders

## 3. Migrate Watermarks

## 4. Migrate Core Components

- 4.1. Migrate Batch Components
- 4.2. Migrate Choice Router
- 4.3. Migrate Exception Strategies to Error Handlers
- 4.4. Migrate Enrichers to Target Parameters
- 4.5. Migrate Filters
- 4.6. Migrate the For Each Component
- 4.7. Migrate Poll Component
- 4.8. Migrate Scatter-Gather Router
- 4.9. Migrate Transformers

## 5. Migrate Connectors

- 5.1. Migrate Anypoint Enterprise Security (AES) Module
- 5.2. Migrate to the AMQP Connector
- 5.3. Migrate Database Connector
- 5.4. Migrate Email Connector
- 5.5. Migrate File Connector
- 5.6. Migrate FTP and SFTP Connector
- 5.7. Migrate HTTP Connector
- 5.8. Migrate JMS Connector
- 5.9. Migrate Object Store Connector
- 5.10. Migrate Scripting Module
- 5.11. Migrate Spring Module
- 5.12. Migrate Validate Module
- 5.13. Migrate VM Module
- 5.14. Migrate Web Service Consumer Module
- 5.15. Migrate XML Module



## 6. MuleSoft Connector Migration from 3 to 4

MuleSoft connector architecture has completely changed in Mule 4. In Mule 3, Dev-Kit is used to develop the connector but in Mule 4, it is based on Mule 4 SDK. Each processor is displayed as a separate connector in Anypoint Studio Pallet.

**To migrate the connector to Mule 4, you can use the conversion tool to:**

- Enable the Connector as Mule 4 extension in pom file
- Modify Package element:  
`<packaging>mule-extension</packaging>`
- Modify parent element  
`<parent>`  
`<groupId>org.mule.extensions</groupId>`  
`<artifactId>mule-modules-parent</artifactId>`  
`<version>1.0.0</version>`  
`</parent>`
- Modify the folder structure  
The typical folder structure for Mule 4 is:  
`<Module>/api`  
`<Module>/internal`
- Update Annotations and Params Classes
- Delete/Add/Update Classes based on Mule 4 SDK (Please refer Java Docs)



## References

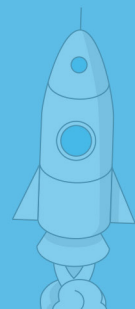
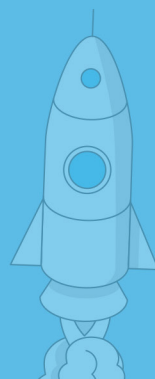
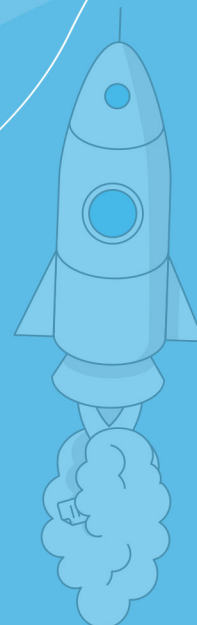
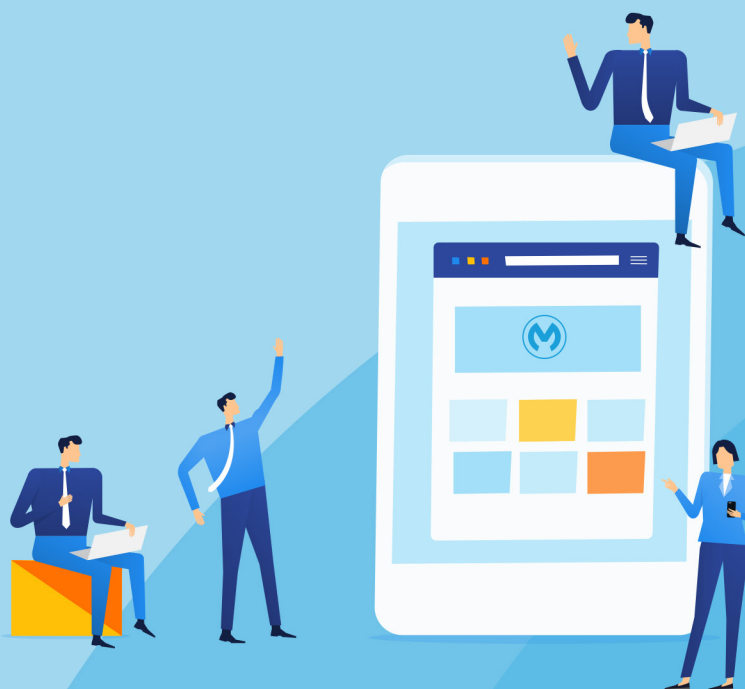
- API-led Connectivity the Next Step in the Evolution of SOA- White Paper
- <https://docs.mulesoft.com/mule-runtime/4.2/intro-overview>
- <https://blog.vsoftconsulting.com/blog/understanding-error-handling-in-mule-4>
- <https://blog.vsoftconsulting.com/blog/mule-4-vs-mule-3-release-which-is-more-beneficial>
- <https://blog.vsoftconsulting.com/blog/a-step-by-step-process-for-mule3-to-mule-4-migration>
- <https://blog.vsoftconsulting.com/blog/best-security-practices-for-migration>







To learn More Visit  
[www.vsoftconsulting.com/expertise/mulesoft/](http://www.vsoftconsulting.com/expertise/mulesoft/)



**V-SOFT**<sup>®</sup>  
LABS - INVENTING TECH