# Ignoring Fields in ACE Integration Testing: A Guide

Many users are well-acquainted with the basic integration testing capabilities of the ACE framework, but its features for ignoring specific fields during testing often go unnoticed. In this post, we'll delve into how to use these capabilities to simplify your testing process and avoid failures caused by dynamic data such as UUIDs and timestamps.

If you do not feel like reading the entire test setup and just want to get to the sweet part, I got you covered as well.

A close-up of a computer screen

Description automatically generated

## Setting Up the Test Environment

Let's start with a simple setup. Our test flow will receive an HTTP call, process the data, and output the results. We have some basic code that maps, transforms, and adds fields. While this isn't groundbreaking, it is a solid foundation for our integration test example. At least it gets the job done of showing how everything works.

A computer diagram of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

## Creating and Running the Test

Let’s start by creating the test case:

1. Start the Flow Exerciser and send a test message through (a test message is available in the code project).

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Check the Response, ensure that you don’t receive an error from your flow and that your response looks like this

A screenshot of a computer screen

Description automatically generated

1. Create a Test Case from the data you just send through the flow. Right-click on the node you wish to test and select *Create Test Case*.

A screenshot of a computer

Description automatically generated

1. For this blog we will focus only on the message body, so accept the defaults for creating the test case.

A screenshot of a computer

Description automatically generated

After you click on *Finish*, you will see the newly created test case, for that is “FeedApplication\_Test”.

A screenshot of a computer

Description automatically generated

Only thing left to do is to run it and look at the results. So, right-click on the test application and select *Run Test Project*.

A screenshot of a computer

Description automatically generated

A screenshot of a computer error

Description automatically generated

And we have a failure. Who would have expected that 😉

## Handling Test Failures

Upon running the test, you might encounter a failure. In our example, the culprit is a UUID field that changes with each test run. If you look at the details of the failed test, it shows you what went wrong

A white and black line

Description automatically generated with medium confidence



As you can see from the code, the field “/Message/JSON/Data/requestSession” is indeed a uuid.





With the code generating a new UUID for every message, this leads to a mismatch between the actual and expected messages.

## Ignoring Dynamic Fields

To resolve this, we need to modify our test project to ignore such dynamic fields (sounds rather obvious, I would say).

We know from the error that the problem field is

/Message/JSON/Data/requestSession

So let’s go ahead and just ignore that one field. In your Java test code, go to the assert statement (for me that is on line 88).

A screenshot of a computer

Description automatically generated

The *equalsMessage* statement supports some additional methods that we are going to be using. One of these methods is *ignorePath.* Let’s change that line of code to

assertThat(actualMessageAssembly.equalsMessage(expectedMessageAssembly).

ignorePath("/Message/JSON/Data/requestSession", false));

A screenshot of a computer code

Description automatically generated

The ignorePath method requires two parameters:

* The path to the field you want to ignore.
* A boolean to decide if subpaths should also be ignored (true to ignore the path and its subpaths).

Save the code and rerun your test.

A screenshot of a computer

Description automatically generated

Ok, so still not a successful test, but we now have stranded on another error, perfect!

## Additional Challenges: Timestamps

After addressing the UUID issue, we are now seeing a TIMESTAMP field that is causing the failure. There are 2 options, similar to our previous approach we can simply ignore this one field, or we can go ahead and ignore all timestamps in one go.

***It is important to know, that you can chain multiple ignore\* methods together on the equalsMessage function.***

To ignore all TIMESTAMP fields, append *ignoreTimeStamps()* to *equalsMessage* (the order in which you chain ignores makes no difference).

A screenshot of a computer program

Description automatically generated

assertThat(actualMessageAssembly.equalsMessage(expectedMessageAssembly)

.ignorePath("/Message/JSON/Data/requestSession", false)

.ignoreTimeStamps());

Save and run the test again

A screenshot of a computer error

Description automatically generated

What do you know, success! Almost like I planned it.

Similar to ignoring TimeStamp fields, you can also ignore DateTime field by using the *ignoreDateTime()* method on the *equalsMessage()*,

**A close-up of a computer screen

Description automatically generated**

## Conclusion

The ACE Integration Test framework offers three key methods for ignoring fields:

* ignoreDateTime(): Ignores DATE and TIME fields.
* ignoreTimeStamps(): Ignores TIMESTAMP fields.
* ignorePath(): Ignores specific paths in the message tree (with or without sub paths).

Utilizing these methods, you can selectively (or broadly) ignore dynamically generated parts of your message tree, enhancing your testing process's accuracy and reliability.

For more integration tips and tricks, visit [Integration Designers](https://integrationdesigners.com) and check out our other blog posts.

Resources:

[https://github.com/matthiasblomme/Ace\_test\_cases/tree/features/add-unit-test-test-case/UnitTestIgnoreFields](https://github.com/matthiasblomme/Ace_test_cases/tree/features/add-unit-test-test-case/UnitTestIgnoreFields%20)

<https://www.ibm.com/docs/en/app-connect/12.0?topic=solutions-developing-integration-tests>

<https://www.ibm.com/docs/en/app-connect/12.0?topic=dit-developing-integration-tests-by-using-app-connect-enterprise-toolkit>

<https://www.ibm.com/docs/en/app-connect/12.0?topic=ditbuacet-creating-test-case-message-flow-node-using-recorded-messages>

Written by [Matthias Blomme](https://www.linkedin.com/in/matthiasblomme/)