# Research Document Dovetail

## Research question

Is it better for ThingsDB to use Dovetail or to create our own project?

## introduction

In this document I will do research on Dovetail and see what the advantages and disadvantages of this program are. So this document contains generic information about what Dovetail claims to be and a research I did based on an interview with a software developer who uses Dovetail on a weekly basis. Why this research? ThingsDB wants to make its system easier for users in the same way and I want to investigate whether this can be done better with Dovetail or by creating our own project for this.

## What is Dovetail?

Dovetail is an Enterprise Service Bus (ESB) software package, also known as iPaaS, that takes care of connecting internal and external applications. This is done by transforming, enriching, controlling and transporting data to any format you want. It is used with proven technology and it is very user-friendly. It can be used in all organizations, regardless of how big or small it is.
Dovetail uses a Flow Designer which ensures that you can schematically see the message flow and know how this is going. In this model, two existing systems are the start and end point. The Flow can be set entirely to your own liking, and the possibilities are endless. Setting the message flow can be done by anyone, programming knowledge is not necessary. By means of a dashboard you can easily check which flows are running and how they are progressing.
What makes Dovetail so special is that it simplifies the process flow. A process flow helps you connecting your applications to other applications. Where other process flows are normally written in technical language that requires programming knowledge, Dovetail uses a process flow with drag & drop functions with built-in components. It offers a user-friendly and graphical user interface so that it is much easier for everyone to read. Visual configuration becomes much easier, since you use the drag & drop functions to build the components in a process flow step by step from start to finish. It is very broad and you can easily link your ERP with the most famous webshop platforms (for example Bol.com, Amazon or Magento), this can be useful as nowadays connection with the digital outside world is becoming increasingly important. There must be flawless communication with your suppliers and your customers. Dovetail also simplifies the connection of exchanging information by means of, for example, an API or web services based on the pre-chewed components.
Dovetail also offers a management tool that gives a clear overview of all the flows you have created. You can easily view the statuses at a single glance. A malfunction in a flow can easily be found here.

## Interview with an expert:

To find out how Dovetail really works, I wanted to find out what the system really looks like, and how it really works when it is used for a company. That's why I needed someone who was familiar with

Dovetail and could also express his honest opinion about it. I talked to Timo Janssen who works with me in the company I have an internship at, Timo also works at a company called Wuunder, Wuunder is a transport management service that automates the shipping process. Wuunder also uses Dovetail so this was very convenient for me. Timo was able to show me Wuunder's system where a dozen flows. I got some explanation about how Dovetail itself works and how it works, but we mainly talked about the pros and cons of the program.

**Advantages Dovetail**

First of all, we talked about the benefits of Dovetail. First of all it is very nice that so many components are offered because this ensures that there are many possibilities for your flowchart, so it offers many options.

It is also nice that notes can be added to the flows, to the flow itself or to a specific component. This can be useful if you look back at your flowchart later to see what actually happens, and it is also very useful for others if they take over or have to work on the flow.

Warnings are also displayed very well on Dovetail, it clearly indicates what can be improved to your flow so that it might be better executed or that errors can occur in the future if you keep the flow as it is. It is also displayed well because the components get a different color, so you know exactly which components are involved.

You also have a very clear overview of all flows that are in your system, namely in your management tool. Here you can clearly see information such as your version, the status of the flowchart, how long it has been running, and when it was last used.

Furthermore, you can use the flows you have created on multiple systems, so you can create a test environment where you create flows and eventually use them with the system that you make public. This can easily be done by filling in a key/id that belongs to the system.

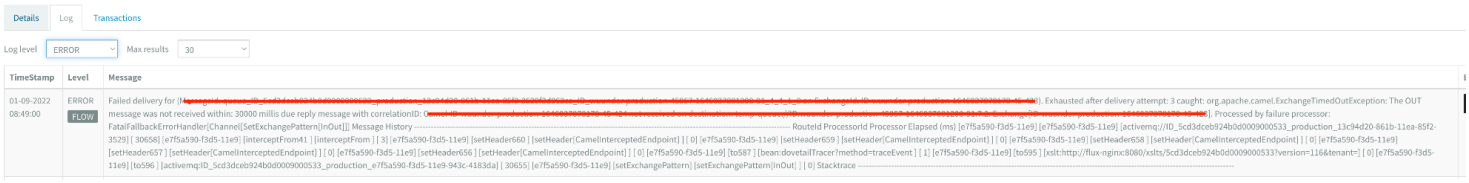
Finally, the version control of the flows is very nice to use, you have a clear overview of all saved versions and you can just click on them to see, for example, how your flow looked at that moment and what you may have changed.

**Disadvantages Dovetail**

First of all, it is a very heavy system, because it is Java based and this brings a number of disadvantages; because it is a heavy system, it works very slowly, for example if you want to expand a flow it can take a fairly long time before this action is finally performed. It is advertised that it works faster because there is a drag & drop function in it, but due to the heavy system you also waste time waiting for your program to take an action.

Another negative point about the heavy system is the costs, Dovetail does not choose to make the profit margin higher and make the prices so high to make more profit but they actually have to make the prices so high because the system is so big and heavy to be able to make money from its system. There are many other programs that work with other systems that can work much cheaper.

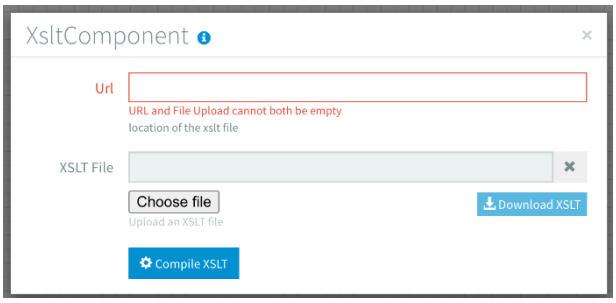
Also, the error logs are very unclear and you actually have to have knowledge of programming if you want to understand what is meant by these errors, it is advertised that it should all be possible without programming knowledge, but if you want to understand these errors, even as an intermediate programmer you really need the time if you want to understand what’s happening.




If an error is found, the flow also goes through the components a number of times because it may be that something goes wrong outside the flow (on the site for example), this causes that if there is really something wrong in a component that you get the same error message over and over again until it decides to stop looping.

| | |
|---|---|
| 01-09-2022 08:47:35 | FlowLinkComponent |
| 01-09-2022 08:47:35 | ScriptComponent |
| 01-09-2022 08:47:35 | SetHeaderComponent |
| 01-09-2022 08:47:35 | HttpComponent |
| 01-09-2022 08:47:36 | VelocityComponent |
| 01-09-2022 08:47:36 | FlowLinkComponent |
| 01-09-2022 08:47:36 | JsonToXmlCurrentComponent |
| 01-09-2022 08:47:36 | FlowLinkComponent |
| 01-09-2022 08:47:36 | XsltComponent |
| 01-09-2022 08:47:36 | XmlToJsonComponent |
| 01-09-2022 08:48:00 | FlowLinkComponent |
| 01-09-2022 08:48:00 | FlowLinkComponent |
| 01-09-2022 08:48:00 | FlowLinkComponent |
| 01-09-2022 08:48:00 | ScriptComponent |
| 01-09-2022 08:48:00 | ScriptComponent |
| 01-09-2022 08:48:01 | SetHeaderComponent |
| 01-09-2022 08:48:01 | ScriptComponent |
| 01-09-2022 08:48:01 | HttpComponent |
| 01-09-2022 08:48:01 | SetHeaderComponent |
| 01-09-2022 08:48:01 | HttpComponent |
| 01-09-2022 08:48:01 | SetHeaderComponent |
| 01-09-2022 08:48:01 | HttpComponent |
| 01-09-2022 08:48:03 | VelocityComponent |
| 01-09-2022 08:48:03 | VelocityComponent |
| 01-09-2022 08:48:03 | FlowLinkComponent |
| 01-09-2022 08:48:03 | FlowLinkComponent |
| 01-09-2022 08:48:03 | JsonToXmlCurrentComponent |
| 01-09-2022 08:48:03 | FlowLinkComponent |

What also can be complicated for people without programming knowledge is the explanation that is given for each component, when you are not sure what a component means you get the option to click on an information button and you are redirected to a page where information is given about this component. This code is very code-biased and little is explained in just general text form. It would be much easier if it were explained more in straight forwarded jargon-free language, so that the novice programmers/non-programmers can understand what is meant by the component.




# XSLT component

The **XLST component** is able to apply transformations on exchanges that carry XML content. It supports XSLT 2.0 and XPath 2.0.

It takes an an (HTTP) URL to a XLST file, which is used to transform the XML content. The XSLT file may include *absolute* links to other files.

## Special care with the XSLT

Any files referenced **inside** the XSLT must have an **absolute http url** that can be reached by Dovetail ESB, otherwise it will not be able to find these resources.

For example:

```
<!-- DON'T DO THIS -->
<xsl:with-param name="file" select="'./codeconversion.xml'" as="item()"/>

<!-- DO THIS INSTEAD! -->
<xsl:with-param name="file" select="'http://dovetail.example.local/xslt/codeconversion.xml'"
```

Apart from the fact that the components are unclearly explained, you also do not have the opportunity to make your own components. You do have many components, but when you need a flow repeatedly in your system, there is no possibility to reform this flow to a component and reuse it in another flow. So it would be much better if you could create your own component of a flow with the specific components you use.

**Summary**

To summarize the advantages and disadvantages, here in brief;

Advantages Dovetail:

- It is very nice that there are so many components offered that you can use
- You have notes that you can add to components so that you and / or others know what it is used for, you can also make a note for the entire flow
- It displays warnings well
- You have a very clear schedule of all your flows that have been created. With enough information, for example about the runtime and how often the flow has been used, which are very useful
- You can use your flows on multiple systems and use them differently based on a system ID/key
- You have version control per flow, so you can see what you made a few versions ago.

Disadvantages Dovetail:

- It is a very heavy system, as it is java based.
- Because it is such a heavy system, it is very slow, even if you only have to do small versions
- It is very expensive because it is so heavy, which makes the price to purchase Dovetail very expensive
- The error logs are very unclear and incomprehensible for someone who has little to no knowledge of programming, even for people with a lot of knowledge it takes a lot of time and that is exactly what Dovetail advertises to save with the flows. It is also not clear where in the flows things go wrong and this should be indicated much more clearly.
- If you want information about a component it is very code-based, so it is difficult to understand if you have little programming knowledge, again it is advertised on the site that it does not need any programming knowledge. This should be more 'Jip and Janneke' language.
- There are many components you can use, but you can't create them yourself. This means if you want to use a flow more often that you have to constantly remake it. So it would be much better if you could create your own component with the specific components you use.

## Conclusion

If we have to summarize this research, we can say that Dovetail has come up with a whole new way of creating an application integration, which makes it easier and more error-proof. On the other hand, there are still a few major disadvantages that make the use of Dovetail more disadvantageous than it is advantageous. To answer the research question, we could conclude that we can learn and adapt a lot from Dovetail but that we see many more possibilities when we make a Flow Control Tool ourselves, since ThingsDB works much faster than Dovetail and is much smaller in scale. Because we have listed the disadvantages of Dovetail, we can ensure with our own Flow Control Tool that we can put together a better program that is much more user-friendly and easier to use for people who have just started programming or even have no experience in this branch.

**Sources**


- M. (2022, 31 maart). *Dovetail - Integration made easy*. Dovetail. Geraadpleegd op 1 september 2022, van https://www.dovetail.world/
- *CMD Methods Pack - find a combination of research methods that suit your needs*. (z.d.). Cmdmethods. Geraadpleegd op 1 september 2022, van https://www.cmdmethods.nl/
- *Research Pattern Navigator - ICT research methods*. (z.d.). Ictresearchmethods. Geraadpleegd op 1 september 2022, van https://ictresearchmethods.nl/Research_Pattern_Navigator
- Timo Janssen, Cesbit