



TFS4JIRA vs. Exalate: The Complete Comparison



Table of contents

What to Consider when Looking for a Software Integration Solution

- How easy is it to use?
- How flexible is the tool?
- Can the tool serve as an Intra-Company and Cross-Company Integration?
- Are you looking for a Migration Tool or an Integration Tool?

What is TFS4Jira?

What is Exalate?

General Comparisons: TFS4JIRA vs Exalate

- Exalate vs TFS4JIRA: How easy is it to use?
- TFS4JIRA vs Exalate: Supported Platforms
- Exalate vs TFS4JIRA: How flexible is the tool?
- TFS4JIRA vs Exalate: Does the tool have built-in autonomy?
- Exalate vs TFS4JIRA: Intra-company vs. Cross-company Integration
- TFS4JIRA vs Exalate: Are you looking for a migration tool?
- Exalate vs TFS4JIRA: Which tool is better priced?
- TFS4JIRA vs Exalate: How's the support?

Bottom Line



Tracking and managing projects using commercial software solutions like Jira, Azure DevOps, or ServiceNow is nothing new. Such work management tools make the life of project managers easier and hassle-free.

As an organization, you already have your suppliers, customers, and partners that probably work on their own work management systems. So in order to collaborate with them, you will need to manually copy and paste data between your systems, which can be error-prone and frankly a lot of work!

You could plan to transition from one work management tool to another to help you communicate and collaborate with each other, but it can be too costly, time-consuming, and inconvenient.

TFS4JIRA vs. EXALATE

Thankfully, software integration tools like [TFS4JIRA](#) and [Exalate](#) can save time, money, and above all sanity. They integrate different work management systems by synchronizing data between them. This is especially useful when you want to share data outside your organization.

This is your complete comparison between TFS4JIRA and Exalate where we compare features, supported scenarios, and their pricing in detail. You can then determine the solution best suited for your business scenario.

Here's what we're going to cover in this blog post:

- [What to Consider when Looking for a Software Integration Solution](#)
- [What is TFS4JIRA?](#)
- [What is Exalate?](#)
- [General Comparisons: TFS4JIRA vs Exalate](#)
 - [How easy it is to use?](#)

- [Supported Platforms](#)
- [How flexible is the tool?](#)
- [Does the tool have built-in autonomy?](#)
- [Intra-company vs. Cross-company Integration](#)
- [Are you looking for a migration tool?](#)
- [Which tool is better priced?](#)
- [How's the support?](#)

What to Consider when Looking for a Software Integration Solution

With too many options and integrations out there, you can easily get confused about what tool might be suitable for your scenario. Here's a list of questions you should ask yourself when choosing a solution:

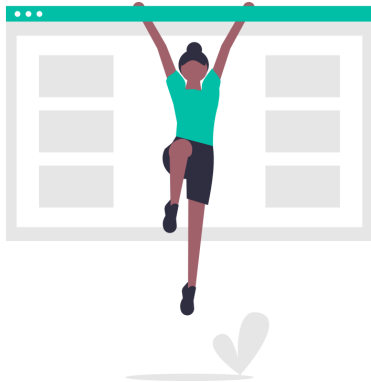
How easy is it to use?



Look out for integration tools that are easy to configure and start with.

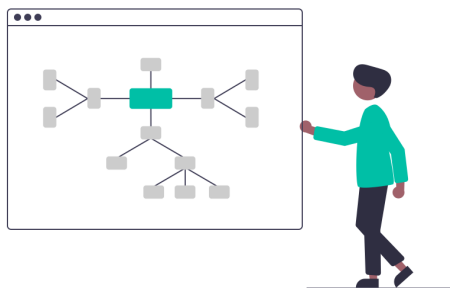
- How easy is it to use?
- Does the software integration solution have a ready-to-use predefined template?
- Is the tool easy to use by the business users or the technical users?
- Will you get confined by the mapping UI?
- Can the technical users/ developers customize the solution based on their use case?

How flexible is the tool?



Most integration tools are confined to a UI which allows for mapping predefined fields that need to be synchronized. But the most effective tool will strike a balance between an intuitive UI and not getting confined by a limited amount of fields to synchronize. Adding an option to customize the synchronization using scripting lets the tool adapt to your specific requirements. Such customization ensures that the tool is flexible to accommodate every possible business scenario.

Can the tool serve as an Intra-Company and Cross-Company Integration?



While choosing a tool for integration, it's important to consider whether the tool serves both internal teams (within different departments of the same company) or external teams (outside the company borders).

An integration solution should give both sides of the connection (be it intra-company or cross-company) the ability to control what's being passed over to and received from the other system without any intervention. Such control is necessary especially for [cross-company integration](#) scenarios as you want the freedom to choose how or whether critical data actually leaves your system or not. Also to choose how incoming data is received and interpreted.

Furthermore, consideration must also be given to how data is retrieved and synchronized in case there is a downtime, how the information exchanged across company borders is secured, and how easily the tool adapts when it comes to complex or advanced B2B use cases. So tools that inherently take into account these features must be given preference.

Are you looking for a Migration Tool or an Integration Tool?



Though the way migration and integration tools work may seem similar, there is a difference between them. Both synchronize information between different systems, but generally, migration tools require enterprise-wide and large numbers of synchronization requests to be handled concurrently.

So even if they both seem to work in a similar fashion, they are innately suitable for different scenarios - one where integration is required and the other where there's a necessity to migrate to a new tool. So the tool you choose must depend on the use case you wish to fulfill with it.

What is TFS4Jira?

TFS4JIRA is an integration and migration plugin to connect two solutions: Jira (Cloud and on-premise) and Azure DevOps/ TFS. It is used to link relevant tasks and projects based on the given context. It helps to integrate and synchronize changes made to the issues and work items between Jira and TFS/ Azure DevOps platforms. Click [here](#) to learn more about TFS4JIRA.

What is Exalate?

Exalate is an integration solution that synchronizes data (issues, work items, tickets, etc) bi-directionally between multiple work management systems. It supports Jira (Cloud and on-premise), Salesforce, Azure DevOps, ServiceNow, Zendesk, GitHub, and HP ALM/QC. The integration can be done internally between departments/ teams or externally across different organizations. Click [here](#) to learn more about Exalate.

General Comparisons: TFS4JIRA vs Exalate

Exalate vs TFS4JIRA: How easy is it to use?





As a company looking for a software integration tool, it is important you consider tools that are easy to set up and get started with.

Predefined templates for both Exalate (called Visual Mode) and TFS4JIRA (called synchronization profiles) provide an easy-to-use, low-code interface that is very intuitive. It provides predefined mappings of certain fields like status, issue type, description, etc depending on the work management tool used. So once you set up all these default mappings and confirm the synchronization connection, you can start sending data back and forth almost instantly

Feature Comparison		
Predefined interface for basic mappings		
Business-user-friendly - Intuitive - Seamless - Low-code Configuration		

Sometimes you need additional tweaking in addition to the predefined interfaces so that you can choose to synchronize in a more flexible manner. So integration tools which allow you to customize the synchronization according to specific business use cases are extremely attractive.

For such customizations, scripting mode is provided only by Exalate. So it can be adopted seamlessly by a variety of users: business (Visual mode) and technical(Visual/ Scripting mode) users alike. It plays an important role in enhancing your overall experience as a user.

Feature Comparison		
Scripting Mode		
Not confined by the UI - Customizable for complex business use-cases		

Here's a peek of what the customers have to say about each of them.



Customer Reviews		
TFS4JIRA is a very convenient way to sync JIRA projects to TFS (and vice versa).	Ease of use	We have a setup for a customer who uses this excellent app to sync between an external Service Desk Jira and an internal Development instance. The first thing that hits you when using this app is how easy it is to get a synchronisation up and running. By default everything works so good.

🔗 Ease of use winner: Exalate

TFS4JIRA vs Exalate: Supported Platforms VIDEO

For an integration solution that handles different work management systems, this question should be on the top of your checklist.

TFS4JIRA only supports integration and synchronization of data between 2 work management systems: Azure DevOps/TFS and Jira.

Exalate on the other hand supports a wider range of work management systems like Jira, Salesforce, Azure DevOps, ServiceNow, Zendesk, GitHub, and HP ALM/QC.

So if you are looking to integrate, for instance, Zendesk to Jira or ServiceNow to Azure DevOps, then you know Exalate is the right solution for you.



Supported Platforms		
✓	Azure DevOps	✓
	ServiceNow	✓
	Zendesk	✓
	GitHub	✓
✓	Jira Cloud	✓
✓	Jira On-Premise	✓
✓	Jira Data Center	✓
	HP ALM/QC	✓
	Salesforce	✓

?Supported integrations winner: Exalate

Exalate vs TFS4JIRA: How flexible is the tool?

An integration solution is always subject to changing business requirements. As such the tool you choose must be able to accommodate these changes.

Sometimes for your business, you might have advanced or complex requirements. It's okay for you to expect that your tool should adapt to them.

At such times, you need something extra from the tool, apart from the pre-defined mappings already present. You need a tool flexible enough to handle your complex business cases with minimal effort.

The scripting mode for Exalate provides an option to customize using Groovy scripting language to help you define your synchronization behavior. Adding such an option to customize gives you the freedom to implement even the most complex business cases.

For instance: You want the data coming from a single issue in Jira to be synchronized under 2 different projects in AzureDevOps. Here, additional scripting is required to make this synchronization work.

Feature Comparison		
Customizability for specific business cases		
Groovy Scripting Language		

Here's what the customers are saying:



Customer Reviews		
The product itself seems to be capable and effective at doing the job we needed of it, yet flexible enough for us to adapt it to our needs.	Flexibility	The configuration using script language offers create flexibility, while plenty of example scripts allow for a quick start. Handling 3rd-party plugin fields was easier than expected.





Flexibility winner: Exalate

TFS4JIRA vs Exalate: Does the tool have built-in autonomy?

As a company, you are always going to be wary about what information goes out of your company borders. Though you are connecting your teams together, you may also have security or legal concerns, especially while handling critical customer data.



So, you need a tool that allows you to retain full control of your environment by specifying what information needs to remain with you and what needs to be passed between systems. This allows the systems to be loosely coupled and independent of each other. For ensuring such autonomy, a tool that has distributed architecture is the best bet.

Exalate is built on top of a distributed architecture, whereas TFS4Jira has a centralized architecture.

Feature Comparison		
Distributed Architecture		
<p>Allowing for a complete decentralized control giving administrators of each system autonomy over:</p> <ul style="list-style-type: none">• What information is shared with the other side• How to interpret the incoming information <p>This ensures data is protected even in complex business scenarios.</p>		

As seen before, to have control over information passed between systems, there must be an efficient mapping mechanism between different fields of work management systems. If such a mapping is in place, it will help you to make better decisions about your critical data.

Almost any type of information can be exchanged using Exalate with the help of the [Custom Keys](#) property. The same can differ slightly for TFS4JIRA.

	Field Synchronization		
✓	Work Item type ↔ Issue type	✓	
✓	Title ↔ Summary	✓	
✓	Description ↔ Description	✓	
✓	Priority ↔ Priority	✓	
✓	State ↔ Status	✓	
✓	Comments ↔ Comments	✓	
✓	Attachments ↔ Attachments	✓	
From TFS to Jira only	Created by ↔ Assignee	✓	
Not the fields provided by an external plugin	Custom Fields ↔ Custom Fields	✓	
From TFS to Jira only	Area Path ↔ String Value	✓	
✓	Assignee	✓	
Subtasks in Jira can be mapped to tasks in TFS	Subtasks	✓	
✓	Labels	✓	

🔗Autonomy Winner: Exalate

Exalate vs TFS4JIRA: Intra-company vs. Cross-company Integration

Both TFS4JIRA and Exalate are software integration tools that allow synchronizing data across different work management systems so that teams within and across company borders can collaborate with each other seamlessly.

Generally, data exchange within teams of the same organization doesn't have strict legal, compliance, or security constraints. But when it comes to sending and receiving data across company boundaries ([B2B integration](#)), you must acknowledge these constraints and choose an integration solution with a little more precaution.

Let's see some examples of what you should expect from an integration tool in cross-company scenarios.

Decentralized control over data

It is important in a cross-company integration scenario that neither of the parties lose control of what information is sent out and how incoming information is processed. Keeping the systems loosely coupled is essential to ensure scalability and maintainability. This means having visibility over your critical data, or deciding what to send over to the other side, and even controlling how you want to interpret the incoming data. This is important if you want to work in your environment and at the same time not worry about critical data being misused, or being sent across without your knowledge.



Using Exalate, you can configure the incoming and outgoing syncs using sync rules on either side without the other party knowing about it. This allows you to have more control over the data being synchronized. This is possible only in a distributed architecture. TFS4JIRA has a centralized architecture where the **TFS4JIRA synchronizer** works as a centralized entity through which all synchronization requests have to pass through. So either of the collaborating parties can make changes to the synchronization profiles, but those changes need to be updated on the synchronizer as well.

Flexibility

As a collaboration with your suppliers, customers, or partners increases, your business is bound to come up with additional complexities or features. So you need a tool flexible enough to adapt to even the most complex B2B integration cases.

Exalate and TFS4JIRA both provide default mapping interfaces that have the most common fields of different work management systems already mapped with each other. For example, suppose you are using Jira and you have outsourced your graphics designing to another company using Azure DevOps. When a bug/ticket-work item is raised in Azure DevOps, it will automatically be synchronized with Jira and a corresponding new issue will be created with the help of these tools. They both come with these predefined basic mappings.

But when you want to use the same tool for advanced cases, for instance: comments need to be handled differently based on the context where they are created (check out this [use case](#) here), then you need something extra.



Exalate's scripting mode enables you to have the flexibility of seamlessly integrating even such advanced scenarios without any additional technical expertise. Your administrators can easily configure using simple scripting snippets/ examples and make configurations of complex cases look extremely simple.

Apart from this, Exalate already integrates with 7 work management systems, which automatically makes it a better choice for flexibility, since you are not limited to only Azure DevOps/TFS and Jira.

Reliability

Sometimes things go wrong even with the most well-architected systems. So you need a solution that handles these problems gracefully. Any errors or changes in the integration should be detected and recovered without manual intervention. This is because even with the most robust systems, downtime is an unavoidable reality. So, your integration tool needs to be able to recover when one or both sides of the integration become unreachable.



Reliability is crucial to ensure any changes happening on one side of the connection, will always be applied to the other side. This, even when a system is being upgraded to a new version and/or a firewall is being reconfigured. Simply put, if a company is facing some downtime, then the other side(s) of the sync should not suffer. They should be rest assured that there is no data loss, or incorrect/ old data is being exchanged.

Both TFS4JIRA and Exalate ensure the data sent across is reliable, not altered or lost, and is up-to-date, but they both differ on how they implement it.

Exalate has an advanced transactional synchronization engine, which queues all the changes that need to be applied in the correct order of their initiation. And breaks it down into atomic steps that can be retried in case of a failure.

Also, Exalate has an integrated retry mechanism for resolving errors. It helps to recover any failure and resume synchronization from the point it has been interrupted by the error.

For TFS4JIRA, if there is a synchronization failure then you need to perform either of the two operations for every synchronization error.

1. Acknowledge to ignore the error.
2. Click on Force resynchronization, which will force the TFS4JIRA Synchronizer to retry synchronization at the next attempt.

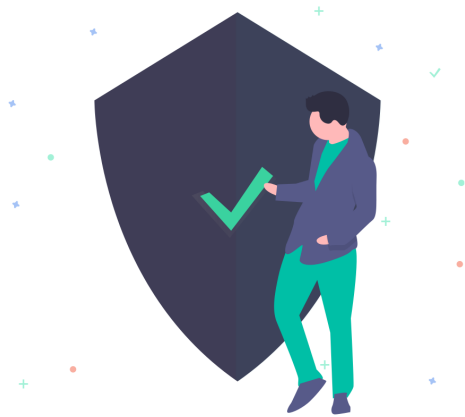
[Click here](#) to learn more about it.

So there is manual intervention needed to reinforce synchronization if anything goes wrong.

Security

For any kind of information exchange (especially in cross-company cases), it's necessary to ensure that it's done in the most secure way possible, such that:

- It's done between all familiar parties.
- Both the sender and receiver want to make sure the information is sent to the right destination.
- It's not tampered with by third parties.
- It's coherent with what the sender wants to send and what the receiver wants to receive.



Potential risks for breach of security can be:

- Man in the middle attack when data is sent across different networks belonging to different companies.
- Data is sent and received by the wrong instance.

To mitigate these risks:

Exalate uses a JWT based approach. Once a connection between 2 instances has been set up, Exalate generates the shared secret. This is used to define a secure connection between both instances. It is shared only once to generate a JWT token. The token is

temporary and is generated for every communication request. It authenticates the request so the destination side can be sure they are getting data from the expected instance.

It also uses access control to ensure that the right people have access to critical information.

To learn more, check the security and architecture whitepaper available [here](#).

TFS4JIRA synchronizer’s access control is based on a standard windows authorization mechanism, where granting or denying access to the application is set up by appropriate read, write permissions.

To learn more about TFS4IRA’s security, read this [document](#).



Intra-company vs. Cross-company Integration

Intra-company vs. Cross-company Integration		
Decentralized control over data For loosely coupled systems in a distributed architecture.	Allows to make decisions about what data should/ should not be synchronized, but is limited by centralized architecture .	Truly distributed architecture for ensuring autonomy on both sides.
Flexibility For the tool being flexible enough to adapt to even the most complex B2B integration cases.	No scripting mode present for advanced use-cases.	Groovy-scripting based engine for handling complex business cases.
Reliability To ensure changes are applied and data is up-to-date after downtime without the need for any manual intervention.	Ensures that data synchronized is reliable (not lost or altered) and up-to-date. Manual intervention needed.	Advanced transactional synchronization engine , which queues all the changes that need to be applied in the correct order of their initiation. An integrated retry mechanism for resolving errors. It helps to recover any failure and resume synchronization from the point it has been interrupted by the error.
Security Important for ensuring data exchanged between systems is secure.	Uses standard windows authentication mechanisms.	Uses JWT token mechanism and has access control.

🔗 Intra-company vs cross-company winner: Exalate







Though both TFS4JIRA and Exalate provide integration for intra- and cross-company scenarios, Exalate was designed with these specific issues in mind. It gives your teams autonomy, enables them to work flexibly, ensures data exchange is secured by advanced mechanisms, and is reliable enough to handle errors automatically.

TFS4JIRA vs Exalate: Are you looking for a migration tool? 📄

Both TFS4JIRA and Exalate support integration and synchronization between different work management systems.

But when it comes to migration, TFS4JIRA is the only one out of the two to offer it. It allows migration from Jira to AzureDevOps/ TFS and vice versa. Exalate has a single-threaded nature because of which it is not meant to do large migrations.



		
Integration and Synchronization Tool		
Migration Tool <small>For migration between Azure Devops/TFS to Jira and vice versa.</small>		

Migration tool Winner: TFS4JIRA











Exalate vs TFS4JIRA: Which tool is better priced?



How do Exalate and TFS4JIRA compare when it comes to pricing? For starters, both Exalate and TFS4JIRA offer a 30-day free trial. The trial rates differ based on the types of work management systems chosen.

Based on our research, here's what you can expect pricing-wise:

Note: Though Exalate supports other work management systems besides Jira and Azure DevOps, for the ease of comparison we have considered only Jira and Azure DevOps/ TFS integration pricing.









For Jira Cloud:



		
Number of Users	Annual	Annual
Up to 75	\$120-\$3,600 	\$3,600-\$4,440 
Up to 100	\$4,800 	\$4,740 
Up to 200	\$8400 	\$5,340 
Up to 10,000	\$54,450 	\$34,890 

 Low
 High









For Jira Server:





Jira Server	Number of Users	Yearly	Yearly
	Up to 75	\$10 - 3,000 	\$3,639 - \$4,240 
	Up to 100	\$5,500 	\$4,690 
	Up to 10,000	\$25,000 	\$10,440 
	+10,000	\$30,000 	\$12,740 

 Low
 High

For Jira Data Center:

Jira Data Center			
	Number of Users	Yearly	Yearly
	Up to 50	\$4,000 	\$5,240 
	Up to 100	\$6,000 	\$5,540 
Unlimited	\$60,000 	\$18,515 	



 Low
 High

Note: The pricing for both Exalate and Jira is inclusive of the Azure DevOps License. For TFS4JIRA, there is no license fee for Azure DevOps. But for Exalate there is a fixed yearly license fee of \$3540.

To learn more about the detailed pricing structure check: [TFS4JIRA](#) and [Exalate](#).

As is evident from the table above, Exalate is cost-effective as compared to TFS4JIRA when the number of users increases.

Here's what the customers have to say about the pricing:



Customer Reviews		
No reviews	Value for Money	I'm a big fan of Exalate and it has proven worth every single cent!

TFS4JIRA vs Exalate: How's the support?

Operations and support play a vital role in software integration tools as a constant need to accommodate specific integration scenarios keeps coming in from companies. More so, you might also have a lot of questions on how certain scenarios or business cases can be synchronized. So having live representatives or online support portals can go a long way in keeping customers happier.

With Exalate and TFS4JIRA you can expect support in the following ways:



Support	Jira	Exalate
Live representative in business hours		
An hour of onboarding call		
Support through service desk or forum		Support for configuration queries through community or professional services.
Online support		

Let's see what the customers have to say about the support they've received:



Customer Reviews		
Support team was outstanding in helping with the synchronize application	Support	The Exalate team have been incredibly responsive to our challenges and issues, providing solutions as well as updating documentation to address our specific use cases.

Bottom Line

Choosing the right software integration solution can save you money, time, and resources. At the same time, you can enjoy the increased collaboration between internal and external teams without having to track everything manually. This leads to seamless integration and increased productivity.

So it's important to choose a tool that checks all your requirements boxes and still provides you with the flexibility you need to do something extra every time.

To summarize:

- Exalate is a hands-down winner for supporting more work management systems as compared to TFS4JIRA
- The scripting mode for Exalate tends to give it an edge over TFS4JIRA for custom fit synchronization scenarios
- Exalate has a distributed architecture that helps control what information is shared across systems, unlike TFS4JIRA
- Exalate doesn't qualify as a migration tool like TFS4JIRA
- Both are cost-effective, but Exalate gives better pricing than TFS4JIRA for more users

Recommended Reads:

- [ServiceNow IntegrationHub or Exalate: How do they Compare?](#)
- [How to set up a Jira Salesforce Integration](#)
- [Jira Azure DevOps Integration](#)
- [How to Set Up an Azure DevOps ServiceNow Integration](#)
- [How to Set up an Azure DevOps GitHub Integration](#)