Can I Sync Data One-Way or Two-Way Using Exalate?

Last Modified on 02/17/2026 9:29 am EST

Yes, Exalate supports both one-way and two-way synchronization depending on how you configure your sync scripts. The platform's flexibility allows you to design synchronization patterns that match your exact business requirements and governance policies.

Two-Way Synchronization (Bidirectional Sync)

Two-way sync is the default pattern where changes made on either side propagate to the other, keeping both systems continuously updated. This bidirectional approach works well when teams on both sides actively work on synchronized work items and need to see each other's updates in real time.

Common use cases for two-way sync:

- Cross-functional teams collaborating on shared work items across Jira and Azure DevOps
- Development and support teams working together between ServiceNow and Jira
- Product teams coordinating tasks between Asana and Salesforce
- Engineering teams syncing code-related work between GitHub and Jira

In two-way sync, both outgoing and incoming scripts are configured on each side. When a field is updated on one platform, the outgoing script sends that change to the replica, and the incoming script on the other side applies it to the local work item. This creates a continuous feedback loop where both systems stay synchronized.

One-Way Synchronization

One-way synchronization flows data in a single direction, typically used when one system acts as the source of truth and the other consumes updates without sending changes back. You implement one-way sync by configuring only the outgoing script on the sending side and the incoming script on the receiving side, without creating the reverse flow.

Common use cases for one-way sync:

- Development teams pushing status updates to a customer portal that displays information but doesn't update back
- Escalating customer tickets from Zendesk to Jira for internal resolution without syncing internal work back to customers
- Reporting dashboards in Salesforce that consume data from ServiceNow without modifying the source
- Freshdesk or Freshservice feeding support metrics to Azure DevOps for tracking without reverse updates

For true one-way sync, the receiving side simply doesn't populate any replica fields in its outgoing script, so nothing flows back. The sending side defines what information to share in its outgoing script, while the receiving side's incoming script determines how to apply that data locally.

Hybrid Synchronization Patterns

You can create hybrid patterns where some fields sync bidirectionally while others move in only one direction. A connection might sync comments and attachments two-way so teams can collaborate, but only accept status and priority updates from one designated side that controls those attributes.

Example hybrid patterns:

- **Collaboration with controlled workflow:** Comments, attachments, and descriptions sync bidirectionally for team collaboration, but status changes only flow from the project management system (Jira) to the ticketing system (Zendesk) to maintain workflow control.
- **Selective field synchronization:** Custom fields sync one-way from the source system while standard fields like summary and assignee sync bidirectionally.
- **Time-based patterns:** Work items sync bidirectionally during active development, then switch to one-way sync once marked as complete to prevent accidental modifications.

This granular control lets you design synchronization patterns that match your actual business processes and governance requirements. You have full script and operational control over your sync, ensuring each field behaves exactly as intended.

How Autonomous Control Supports Synchronization Patterns

The autonomous control architecture supports these patterns naturally. Each side independently defines what to send in outgoing scripts and what to accept in incoming scripts. The sending side doesn't need to know what the receiving side does with the data, and vice versa.

This approach maintains the flexibility to adjust synchronization direction later without requiring coordination with the other side. If business requirements change and you need to convert a one-way sync to bidirectional, you simply add the appropriate scripting logic on the receiving side without modifying the sender's configuration.

Security controls remain independent as well. Each side maintains role-based access control, encryption of data in transit and at rest, and ISO-certified infrastructure regardless of sync direction.

Getting Started

Start a free trial to configure one-way, two-way, or hybrid synchronization patterns tailored to your integration requirements.

Subscribe for a weekly Exalate hack

Academy

Blog

YouTube Channel

Ebooks

Still need help?

Join our Community

Visit our Service Desk

Find a Partner