



ServiceNow Integrations: Integrate ServiceNow and Other Systems Bidirectionally

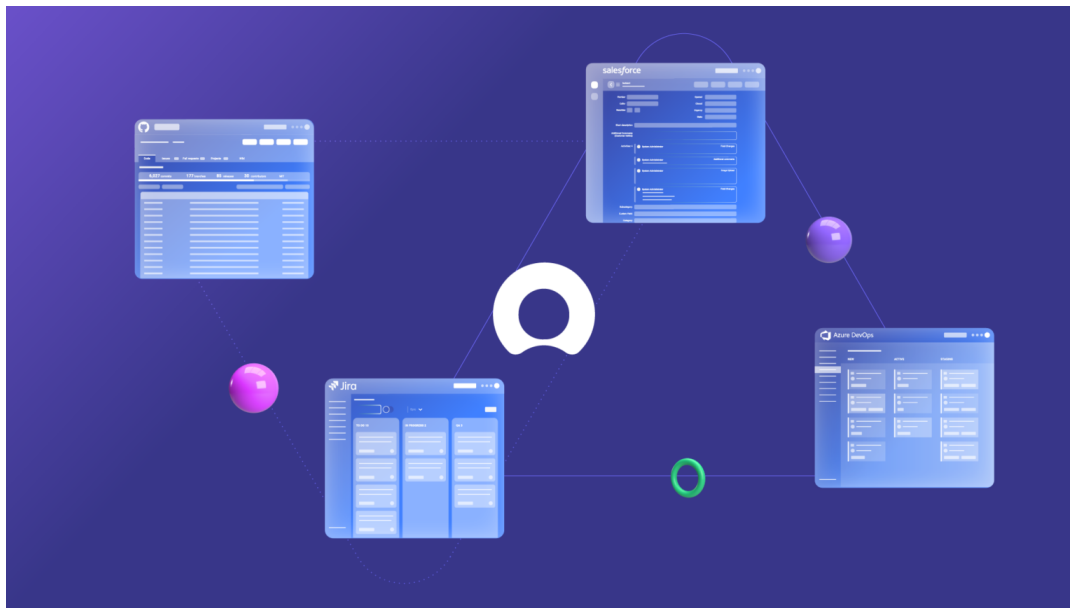


Table of contents

The Need for ServiceNow Integrations

ServiceNow Integrations: An Interconnected ServiceNow Ecosystem

- Benefits of a ServiceNow Integration

Common Use Cases for ServiceNow Integrations

- Integration between ServiceNow and Other Applications
- Integration between Different ServiceNow Instances

Native Ways to Implement ServiceNow Integrations

- Exploring Various Integration Technologies Offered by ServiceNow
- What Integrations Does ServiceNow Provide?
- Challenges of Implementing ServiceNow Native Integrations

Using 3rd-Party Apps for ServiceNow Integrations

- Checklist for Selecting the Right ServiceNow Integrations App
- Exalate: a 3-rd Party Integration Solution for ServiceNow Integrations

How to Set up ServiceNow Integrations with Other Apps or Systems

- Step 1: Install Exalate on ServiceNow and the Other Application



- Step 2: Connect ServiceNow and the Other Application
- Step 3: Configure the Connection to Decide What to Share
- Step 4: Start Automatic Synchronization Using Triggers
- Step 5: Start the Synchronization

Best Practices for ServiceNow Integrations

Conclusion



ServiceNow has grown considerably from IT infrastructure services to supporting security, development, customer service, HR, and other teams. It has set a benchmark for its potential to detect anomalies and automate digital workflows and processes.

ServiceNow Integrations extend the ability of ServiceNow by reducing siloed functional areas to increase productivity and efficiency.

So information can flow from other 3-rd party systems or applications into ServiceNow. At the same time, information from within ServiceNow can be displayed in other apps or systems.

This blog aims to explain why there is a need for ServiceNow integrations and the benefits it brings to teams. It also discusses the technologies ServiceNow natively offers for such integrations and explores other alternatives.

We then see a practical implementation of ServiceNow integrations using a 3-rd party app called [Exalate](#). And finally, we end the blog by discussing a few best practices for ServiceNow integrations.

So let's get started.

Use the list below to jump to the right section:

- [The Need for ServiceNow Integrations](#)
- [ServiceNow Integrations: An Interconnected ServiceNow Ecosystem](#)
- [Common Use Cases for ServiceNow Integrations](#)
- [Native Ways to Implement ServiceNow Integrations](#)
- [Using 3rd-Party Apps for ServiceNow Integrations](#)
- [How to Set up ServiceNow Integrations with Other Apps or Systems](#)
- [Best Practices for ServiceNow Integrations](#)

The Need for ServiceNow Integrations

ServiceNow's bread and butter is incident management and resolution. However, it is not limited to only that. It also extends to other functional areas. So different teams like HR, software development, operations, etc. can use it just as easily.

These teams also need to interact with other teams using completely different applications like Jira, Salesforce, Azure DevOps, GitHub, etc. In fact, they might also want to [interact with other ServiceNow](#) instances their partners or suppliers use.



Maybe your service desk agents using ServiceNow need to escalate incidents to the dev team using Jira after they perform a root cause analysis. You want to resolve incidents faster to be more efficient.

But if you rely on transferring these incidents manually then it's easy to make errors and introduce delays for customers, further impacting your incident resolution times.

So there is a need to integrate ServiceNow with these different applications and automate business functions. All the information exchange is automatic, in real-time, and visible to the correct stakeholders within their own application. It will digitalize workflows and enrich the customer experience.

This is what ServiceNow essentially stands for and believes in the first place.

And who knows, this might just be the beginning! With the right technology and approach such ServiceNow Integrations can enable a global network of [B2B integrations](#) blurring information boundaries.

ServiceNow Integrations: An Interconnected ServiceNow Ecosystem

Remember, the main aim of a ServiceNow integration is not to take the teams away from their original source of information but streamline procedures and workflows to make their day-to-day operations easier.

Essentially, this means pulling data from external systems and displaying it within ServiceNow. It also includes displaying data from ServiceNow into these external systems.



ServiceNow integrations can take place in [different ways](#):

- Data is moved across ServiceNow and different applications such that processes are automated end-to-end. For instance, sending ServiceNow incidents to Jira or Azure DevOps to facilitate the development workflows. ServiceNow coins this as **Process integrations**.
- Simple data exchange between different applications and ServiceNow. The transfer of such data can either be one-way or two-way depending on the requirement. It can be pulling data from a data source (or database) and displaying it within ServiceNow. This is called **Data integrations** by ServiceNow.

- Display publicly available information within the ServiceNow platform. This is an example of **UI Integrations**. In this case, a certain iframe in ServiceNow is launched into another portal or vice versa. For instance, display a live report in another system.

ServiceNow integrations can either be **batch-based** or **event-based**.

Batch-based integrations allow you to exchange information in batches or in bulk. This can include information stored in CSV, Excel, or any other format.

Whereas, event-based integrations allow data to be exchanged whenever a certain event occurs. For instance, when an incident of **urgency=1** and **Priority=High**, belonging to a specific Assignment group is encountered, a ServiceNow record needs to be passed over to the development team using Azure DevOps or Jira.

Let's look at the benefits of ServiceNow integrations.

Benefits of a ServiceNow Integration

If ServiceNow, as an ITSM solution, becomes the single source of truth for your organization connecting different teams and increasing collaborations, it can reap a lot of benefits:

- Easily route or reroute incidents within platforms you use and automate processes across different departments or even across different companies ([cross-company integration](#)).
Give away tedious, manual, repetitive tasks and let go of Excel sheets and to-do lists.
- Better insights and reporting capabilities through data analytics tools that ServiceNow provides, can help track customer requests, identify patterns and deal with problems early on.
It can also measure key performance indicators like problem resolution times and agreed-on SLAs so as to develop an overall rich customer experience. Also, all the team members will have better visibility of every operation and process and can take advantage of this transparency to work efficiently.

- The customization capabilities of ServiceNow are not unknown. It is also scalable. If these advantages are percolated to other platforms through integrations it can increase productivity and enhance its performance further.
- Sometimes you need to [connect multiple ServiceNow instances](#) such that its pre-built applications like ITSM, HRSM, or CSM must integrate. This can bring diverse ServiceNow instances supporting diverse functions to be tied together within a single (or multiple) enterprise(s). This can help MSPs or MSSPs (using these instances) come together under a single workflow and increase productivity.

Further dwelling on the practical aspects of ServiceNow integrations, let's see a few common use cases.

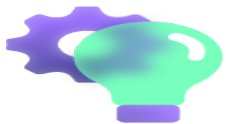
Common Use Cases for ServiceNow Integrations

Integration between ServiceNow and Other Applications

Let's have a look at a few use cases when ServiceNow is connected to other commercially available work management systems like Jira, Azure, GitHub, Salesforce, Zendesk, etc.

Service Desk and Development Teams

Streamlined workflows and faster resolutions between the support and the development teams are the most important measure of delightful customers and increased business. This can happen when all the necessary - and only the required information - is passed between applications they use.



The time taken for an incident to flow across to the development team and be pushed back to the support team for resolution can open up insights. Insights where you can identify bottlenecks and patterns early on and take corrective actions.

Due to this increased transparency, efficiency increases by default. For instance, patch releases, if required, can be expedited keeping the customer at the center of the business.

After all, this is what sound business judgments are all about.

IT and Business Teams

ServiceNow Integrations can provide end-to-end connectivity between IT and business teams. Business teams get valuable feedback (queries, problems, feature requests, etc) from customers, for instance, cases, in Salesforce.



If this feedback is handed over to the IT teams, they can help in making the product more valuable, incorporating proactive improvements. Information can then pass between these multiple applications such that all the teams work towards common business goals.

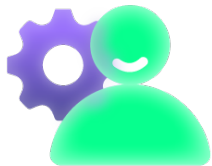
Integration between Different ServiceNow Instances

Sometimes, there is a need to connect multiple ServiceNow instances. Such integrations are often viewed as snaps (duplicate snapshots of 2 ServiceNow instances) of each other but in reality, it is more than that.

Connecting MSPs to Multiple Clients

Managed service providers have to deal with multiple clients often not using the same application as the provider. Maybe they both use different ServiceNow instances. There is a need to pass some (not all) data between them.

Also, multiple clients can have different information requirements. Managing all this information through ServiceNow integrations can strengthen the relationship between the MSPs and their clients and make everyone's life easier.



Having got an idea about the what and why of ServiceNow integrations, let's get into the practical aspects of how such integrations can be achieved.

Native Ways to Implement ServiceNow Integrations

There are different technologies and methods it supports for integrating ServiceNow with external systems.

Exploring Various Integration Technologies Offered by ServiceNow

Depending on the most popular integrations, like Incident Management, CMDB, Problem Management, etc. that are demanded, the following are the technologies ServiceNow offers:

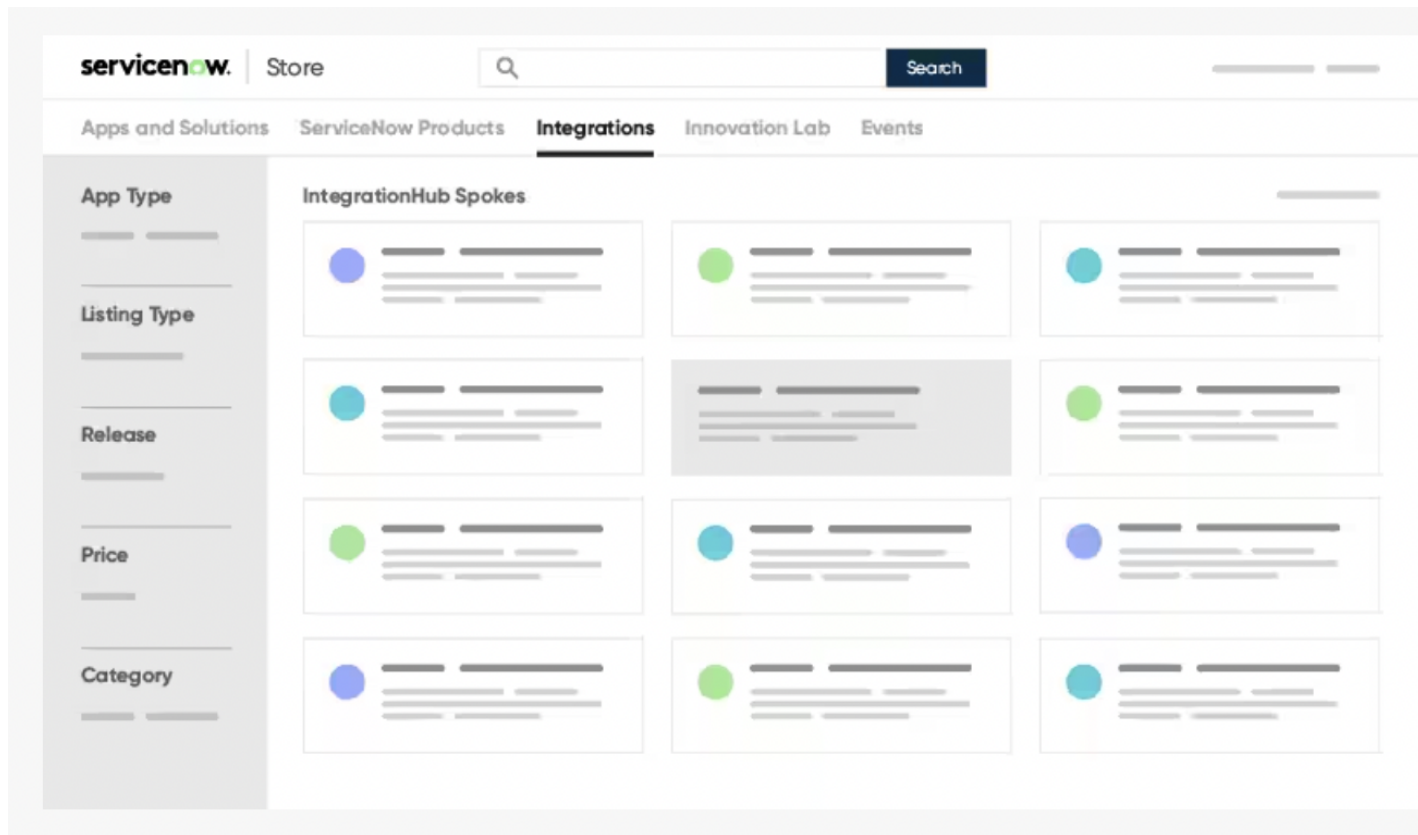
- Web Services
 - ServiceNow allows all data objects to be used as web services for one or two-way data integration. Dynamic modifications to existing and new objects are directly published as Direct Web Service.
 - Another way is through its Mapped Web Service which allows mapping the incoming data into final target tables.
 - There is a Scripted Web Service where data is sent and received between systems through triggers. So it's more of an event-based service like we discussed above.
 - External data can also be loaded via import sets in ServiceNow.
- Information can be pulled via a data source directly using a JDBC connection. Various data sources can be loaded like Excel, CSV files, etc.
- ServiceNow can also pull user data from your instance's existing LDAP server for integration.
- Other industry standard technologies like SOAP, REST, or WSDL are also supported.
- APIs and command-line integrations can also be done using a MID Server.
- A few single-sign-on technologies have been identified and provided integration for. This facilitates faster assimilation into the platform you use. This technique is also customizable through scripting so you can incorporate it for your own SSO environment.
-

Obviously, the list is not exhaustive.

What Integrations Does ServiceNow Provide?

- [ServiceNow provided Integrations](#) that are accessible as a part of the platform. These do not come at an additional cost.

- [ServiceNow Store Integrations](#) that are built by ServiceNow or its partners. The support for the integration is usually provided by the vendor.
- [IntegrationHub](#) allows you to integrate ServiceNow with any API-enabled systems. By far, this is the most popular option for ServiceNow integrations.



It reflects the customization capability of ServiceNow for integrations by extending the functionality of its Flow Designer.

IntegrationHub uses the 3rd party APIs as a part of its flow and triggers information exchange.

An example of such a trigger can be an incident being created or updated. It achieves this in the form of different spokes. Each spoke is application specific. There are spokes for Jira, GitHub, GitLab, Salesforce, etc to connect ServiceNow with these platforms. There are around 175+ pre-built integration spokes.

ServiceNow also realizes the need to connect multiple ServiceNow instances with the help of an [eBonding Spoke](#). This allows you to connect different ServiceNow instances with almost no code and is available OOB (no need for a subscription) with ServiceNow.

- Custom Integrations allow you to use the platform's own interfaces and build an integration from scratch. These are beneficial for very specific use cases but generally need to be maintained for the long term by the customers themselves. So they aren't recommended.

As seen, there are a lot of ways to achieve integrations with ServiceNow natively, so it's safe to say that you can choose to let your ServiceNow instance communicate with any other application of your choice.

But there are also a few caveats we must consider here.

Challenges of Implementing ServiceNow Native Integrations

Though at the onset, it may sound simple to implement the ServiceNow-provided integrations, often it is not.

When it comes to business decisions like these, it is important to have a look at what you expect from such an integration and whether it would be of any value, functionally and economically.

With ServiceNow-provided integrations, ServiceNow often sits at the orchestrating seat and drives them forward. Systems involved in such an integration are tightly coupled creating dependencies and hence additional overheads for maintenance.

Integration requirements that can be fulfilled are limited at times, in terms of capabilities, since they are based on templates and patterns that are inherently provided. So they are perfect if you just want to automatically pull data from an external system and display it within ServiceNow or vice versa.

Nevertheless, for some advanced requirements data mapping can be done using different technologies and methods we saw above. But then doing so requires additional coding efforts and hence additional resources. Not to mention the time taken to even get started is huge. It's like building a bazooka to kill a mosquito!

Sometimes for certain customizations in IntegrationHub a request to the creator of the Spoke needs to be made. So it's not always very simple.

Most of the ServiceNow-provided integrations are available for free but some like IntegrationHub require a separate subscription. At times, the cost for this is not feasible considering the efforts required to set up a simple integration.

Moreover in IntegrationHub, since you aren't able to pick a single integration, you can end up paying for the 175+ integration spokes that you might not even use.

And we still haven't explored the role of 3-rd party vendors who also provide ServiceNow integrations. It's time we do that.

Using 3rd-Party Apps for ServiceNow Integrations

There are a whole host of 3-rd party apps providing different integration options in the ServiceNow store.

They are professionals and experts in ServiceNow integrations and so it makes sense to discuss how we can use them to integrate ServiceNow with other applications.

But we must not be shooting in the dark.

To even reach a conclusion, there must be thorough research and an understanding of what the solution would bring to the table. Keep a few guidelines in your mind.

Checklist for Selecting the Right ServiceNow Integrations App

Security

The most important part of any integration without any doubt is security. This is because the entire essence of an integration lies in sharing data across different applications and you don't want unauthorized access to that data or accidental leaking of information, do you?

So having an integration solution supporting encryption, secure file transfer protocols, role-based access control, etc is a must.



A lot of companies are also concerned about where, what, and how data would reside once it leaves their company or their ServiceNow instance.

Decentralized Integration

An important requirement, for many scenarios, is for the solution to support decentralized integration. All the solutions including the ones provided by ServiceNow are centralized, meaning a single interface works as a controller for the integration.

And trust me, there is nothing wrong with this approach.

But now imagine a scenario where both the integrating platforms have equal and independent rights to choose how they want to send and receive information. This can happen without messing with each other's integration setups or even informing the other side (unless required).



So you simply choose to send (or filter) information that is absolutely necessary to the other side and also decide what you want to receive from the other end **independently**.

Loosely- Coupled Systems

Continuing the point above, the systems in a decentralized integration setup are intrinsically loosely coupled. When systems don't depend on each other (avoiding a single point of failure) it increases the overall scalability.

So it's very easy to add just another point of integration and connect with a new partner.

Flexibility

With integrations, you always have a lot of options. Because there is no “one-size-fits-all” kind of scenario. Requirements change and so does the way in which data exchange is handled. Sometimes, you might have extremely complex needs that have to be implemented by integrating at a deeper level with the platforms.

Now, all of this, as I mentioned before, can also be achieved with ServiceNow-provided integrations or with other solutions in the market.

But the point here is the ease with which this can be achieved.



How much time and effort would the task require? Is it something that the solution can achieve in less time with minimal tweaking and native support for deep integrations? And in the end, is it all even worth it? Would it help in reducing your organization's technical debt?

Having a solution flexible enough to implement any and every integration requirement should be your best shot.

Price

Last but not least, perhaps the most important point businesses want to consider is the price of the integration solution. Please be mindful, this is not the same as the cost of an integration which is a completely different ball game altogether.

Are you paying too much for a simple integration, or are you paying a reasonable price for an advanced integration? These are just some of the few questions you must ask yourself.



Well, are we getting somewhere with all of this?

Yes, I have an integration solution that can be a fit here.

Exalate: a 3-rd Party Integration Solution for ServiceNow Integrations

[Exalate](#) is a one or two-way integration solution with the capability to provide bespoke integrations for different ITSM, ITOM, CRM, or other work management systems.

It supports [integrations](#) for ServiceNow, Jira, GitHub, Azure DevOps, Salesforce, Zendesk, etc. So you can set up a [ServiceNow to Azure DevOps integration](#), a [ServiceNow to ServiceNow integration](#), or even a [Jira to Azure DevOps integration](#) for that matter.

It supports a smaller range of platforms as compared to IntegrationHub but the integration that can be delivered is way deeper. Because it comes with a built-in intuitive scripting engine that allows you to customize mappings and the overall logic to an advanced level.



It is also the only solution that supports decentralized integration, and as we discussed, gives equal and independent rights to either side of the integration.

Plus, it keeps the integrating systems loosely coupled and increases security since both sides decide what information they want to expose.

Let's see a step-by-step walkthrough on how to use Exalate to set up ServiceNow integrations.

How to Set up ServiceNow Integrations with Other Apps or Systems

Note: You can watch the [Exalate Academy](#) tutorials if you are a visual learner.

Step 1: Install Exalate on ServiceNow and the Other Application

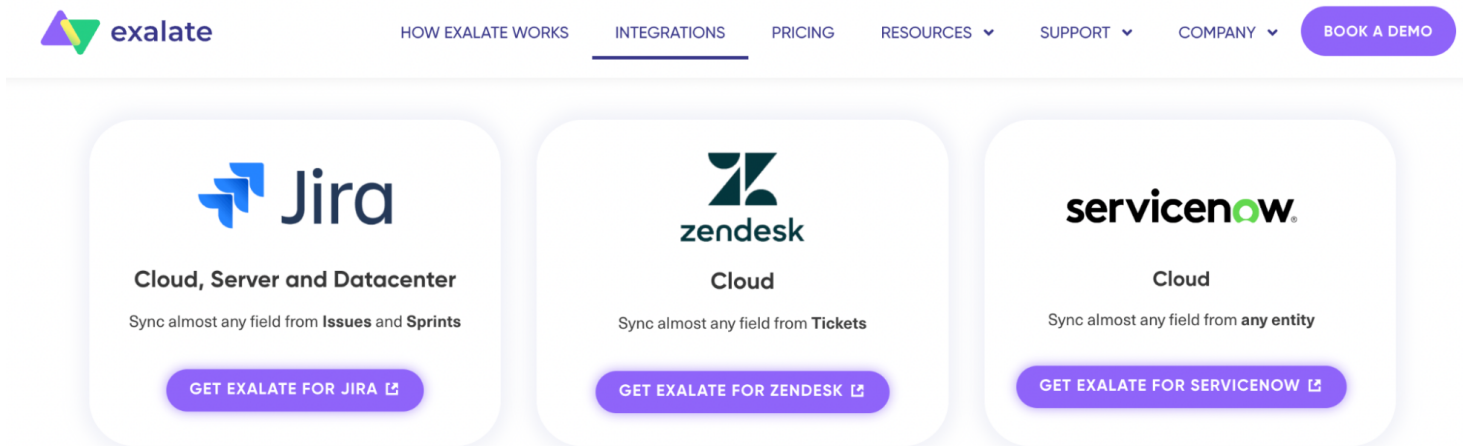
The first step is to install Exalate on ServiceNow and then on the other platform (Jira, Azure DevOps, ServiceNow, GitHub, Zendesk, Salesforce, etc).



We will discuss the steps to installing Exalate on ServiceNow here. For the other platforms, you can refer to the [installation guides](#).

Note: Exalate for ServiceNow can also be installed on [docker](#).

You can request an Exalate instance for ServiceNow from its [integrations](#) page.



A request form will pop up. Fill it in and click “Submit”.

The screenshot shows a request form titled "Request a Free Trial of Exalate for ServiceNow". The form includes the following fields and instructions:

- Organization:** Enter Organization
- First name:** Enter First name
- Last name:** Enter Last name
- Phone number:** Enter Phone number (with a dropdown for country code, currently showing USA +1)
- Email address:** Enter Email address

At the bottom right of the form is a green "SUBMIT" button. A footer note reads: "We respect your privacy - more details [here](#)".

You will then receive an email with the Exalate node URL for ServiceNow along with an evaluation key.

Before you open your email, it's better to create a [Proxy user](#) to enable the correct permissions to use the Exalate console.

Once that's done, open the email you have received, accept the EULA, verify your instance and provide the node with the details it needs to connect to your ServiceNow instance. The steps are easy-to-follow now on.

Step 2: Connect ServiceNow and the Other Application

To start syncing information, you first need to set up a connection between ServiceNow and the other application.

One side initiates the connection and generates an invitation code. The other side accepts the connection invitation using this invitation code. A connection generates a link between the 2 platforms so you can synchronize information.

It doesn't matter which side starts initiating the connection, the Exalate UI remains the same.

Assume we start initiating the connection from ServiceNow.

Navigate to the "Connections" screen in the Exalate console on the left-hand side menu bar.

Click on the "Initiate Connection" button.

The screenshot shows the Exalate console interface. On the left is a navigation menu with options like 'Getting Started', 'General Settings', 'Connections', 'Entity Sync status', 'Triggers', 'Errors', 'Exalate Notifications', 'License Details', 'Bulk Connect', and 'Sync Queue'. The main area is titled 'Connections' and contains a table of active connections. At the top right of the main area are buttons for 'Initiate connection', 'Accept invitation', and a refresh icon.

Connection	Entities under sync	Last sync	Status
MAID_SNOW_TO_SNOW_TEST	12	hc_msg_item HTML... 2 months ago	● Disconnected
SYNC@freespace.to_HLAD@in EPISO connector, please do not delete!	2	incident INCOO... 2 months ago	● Disconnected
INC@out_to_ServiceNow2 Maven	2	incident INCOO... 4 weeks ago	● Disconnected
Mentions between JIRA SNOW	0		● Disconnected


Enter the “Destination Instance URL”. The destination instance URL is the URL of the application you need to sync with. I have considered Jira in this example. But it can be any other application Exalate supports.

Initiate connection ×


Destination instance URL ⓘ

✓ I don't have a URL

Choose the configuration type

 **Basic**

- Automatic configuration of basic fields
- Sync rules cannot be edited
- Only incidents can be synced
- Recommended for use cases of basic complexity

 **Script**

- Groovy-based scripting
- Configure each side of the connection separately
- Recommended for use cases of basic to advanced complexity

Next

You will need to choose between the Basic and the Script mode.

The Basic mode supports pre-defined field mappings between the different systems that cannot be configured. You can sync only incidents in ServiceNow using the Basic mode. It comes with a [Free Plan](#) that allows you 1000 free syncs per month. It lets you test out your sync easily.

The Script mode is where the real power of Exalate lies.

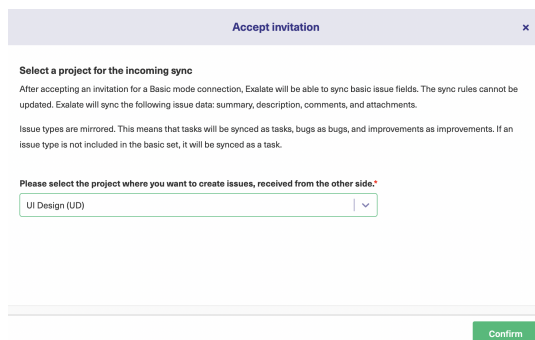
It works on a Groovy-based scripting engine and you can specify and manipulate the fields you want to share. You can also define how to map them and automatically control the sync the way you want.

Let's have a look at the Basic mode first before we go to the Script mode.

The Basic Mode

Select "Basic" in the option above and click "Next".

You will need to verify if you have admin access to the destination instance or not.



Accept invitation

Select a project for the incoming sync

After accepting an invitation for a Basic mode connection, Exalate will be able to sync basic issue fields. The sync rules cannot be updated. Exalate will sync the following issue data: summary, description, comments, and attachments.

Issue types are mirrored. This means that tasks will be synced as tasks, bugs as bugs, and improvements as improvements. If an issue type is not included in the basic set, it will be synced as a task.

Please select the project where you want to create issues, received from the other side.*

UI Design (UD)

Confirm

Once that is done, depending on the destination instance you would either need to select a project in Jira and Azure DevOps or a repository in GitHub, or if no such concept exists then the connection is successfully established.

You can enter the incident number.

Exalate ✕

Connection established successfully ✓

Sync your first incident to see how it works.



Please enter an incident number to proceed



Exalate

And see the result of your synchronization immediately.


Exalate×

Congratulations!


ven03945

INC0011125


idalkoplay

UD-351

Preparing ⓘ Sending ⓘ Waiting for remote ⓘ Finalizing ⓘ Synchronized ⓘ



You can also choose to filter incidents that can be synced using the basic mode automatically with the help of [triggers](#) or you can even [sync entities in bulk](#).

The Script Mode

Choose “Script” and click “Next”.

Enter the connection details. Give a short name to the local instance and a short name to the remote instance. A connection name is automatically generated. You can change it if you want to.

Initiate connection ×

Connection information

Local instance short name*

Remote instance short name*

Connection name*

Description

[< Previous](#) **Initiate**

Click “Next”.

An invitation code is generated. This needs to be copied and pasted into the destination instance.

So click on “Copy invitation code” and save it somewhere safe. Then click “Done”.

Initiate connection ×

On the “Jira” side, you (or their application administrator) need to **Accept the Invitation**.
Use the following invitation code:

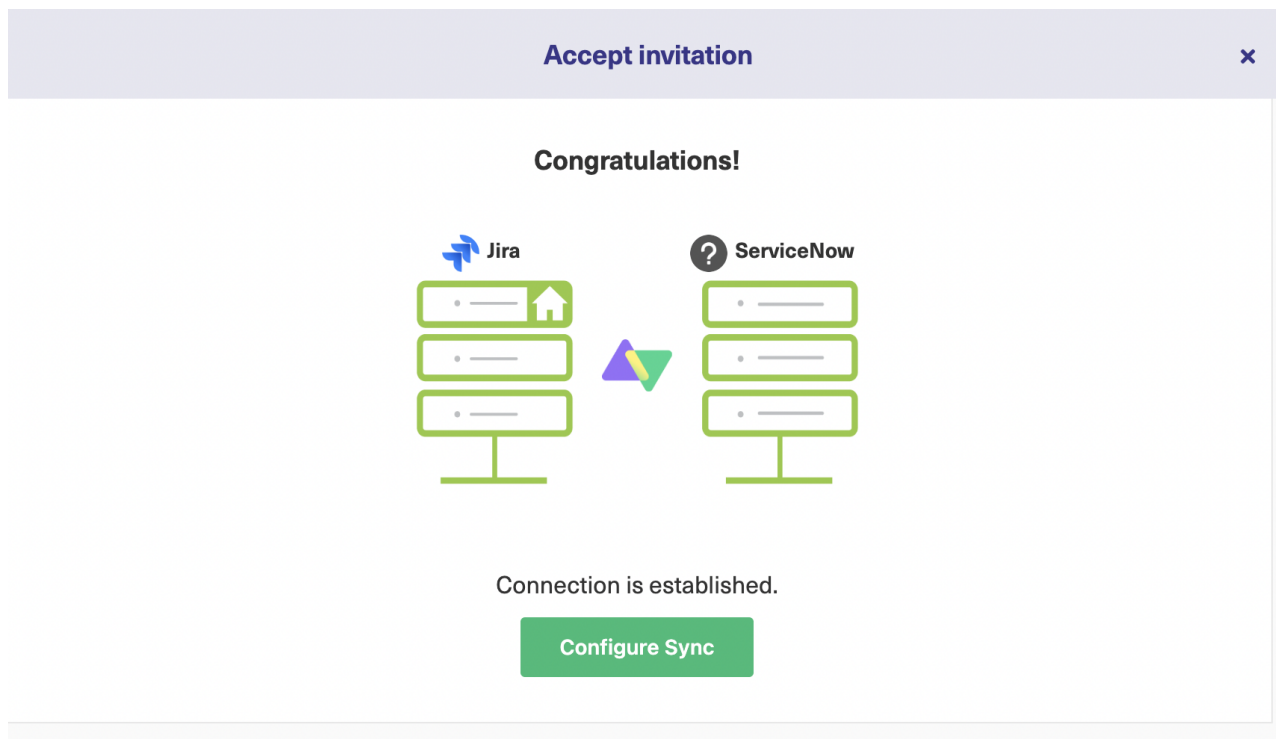
[Copy invitation code](#)

Done

On the destination instance, go to the Exalate console and select the “Connections” tab.

This time around, click the “Accept invitation” button.

Paste the code you have just copied. Again depending on the destination instance, you will either be required to choose a project or a repository or the connection will be successfully established.



Step 3: Configure the Connection to Decide What to Share

On the previous screen, if you click “Configure Sync” you will be shown a panel that looks like this:

The screenshot shows the configuration interface for a connection named "ServiceNow_to_Jira". At the top, there is a breadcrumb "» ServiceNow_to_Jira" with a green dot and the word "Active" below it. To the right are two buttons: "< Back to Connections" and "Publish". Below the breadcrumb are four tabs: "Rules", "Triggers", "Statistics", and "Info". The "Rules" tab is selected. Underneath, there is a section titled "Outgoing sync" with a dropdown arrow and a help icon. This section contains a code editor with the following content:

```
1- if(entity.tableName == "incident") {
2   replica.key      = entity.key
3   replica.summary  = entity.short_description
4   replica.description = entity.description
5   replica.attachments = entity.attachments
6   replica.comments  = entity.comments
7   replica.state    = entity.state
8
9- /*
10  Use a field's internal name to send its value
11  Example: Resolution Notes -> resolution_notes
12  This works for all other entity types as well
13
14  replica.resolution_notes = entity.resolution_notes
15  */
16 }
17 //any other entity can be synced using the table name and the entity variable
18 - if(entity.tableName == "cmdb_ci_business_app") {
19   replica.key      = entity.key
20   replica.summary  = entity.short_description
21   replica.description = entity.description
22   replica.name     = entity.name
23 }
24
```

Below the code editor is a link: "Copy outgoing sync processor to clipboard".

Below that is another section titled "Incoming sync" with a dropdown arrow and a help icon. This section contains a code editor with the following content:

```
1- if(firstSync){
2   //Decide on the first sync, which entity you want to create based on the remote issue type
3   entity.tableName = "incident"
4 }
5
6 - if(entity.tableName == "incident") {
7   entity.short_description = replica.summary

```

At the bottom of the interface, there are links for "Documentation", "EULA", "Support", and "Report a bug". Below these links is the text "Powered by Exalate v. 5.4.10 (Core v. 5.4.10)".

“Rules” and the “Triggers” tabs are the ones we need here.

Rules are Groovy scripts that let you control the information flow.

There are Incoming and Outgoing sync scripts present on both integrating sides. They control the information flowing out and coming into the platforms. Each line corresponds to a particular field that needs to be synced.

In the “Outgoing sync” on the ServiceNow side, the line ***replica.description=entity.description***, means that the description of the ServiceNow entity is copied into something called a replica.

Replica acts like an empty envelope in which you fill the contents you need to transfer to the other side. This content is unraveled at the destination instance and applied according to the rules defined in its “Incoming sync”.

If you don’t want to send a particular field, simply delete the line.

You can also choose to add new lines of code to send or receive new information.

Check out the [Getting started](#) guide on the Exalate documentation to see how these initial steps work out with Exalate or refer to the [Script Helpers](#) for advanced scripting logic.

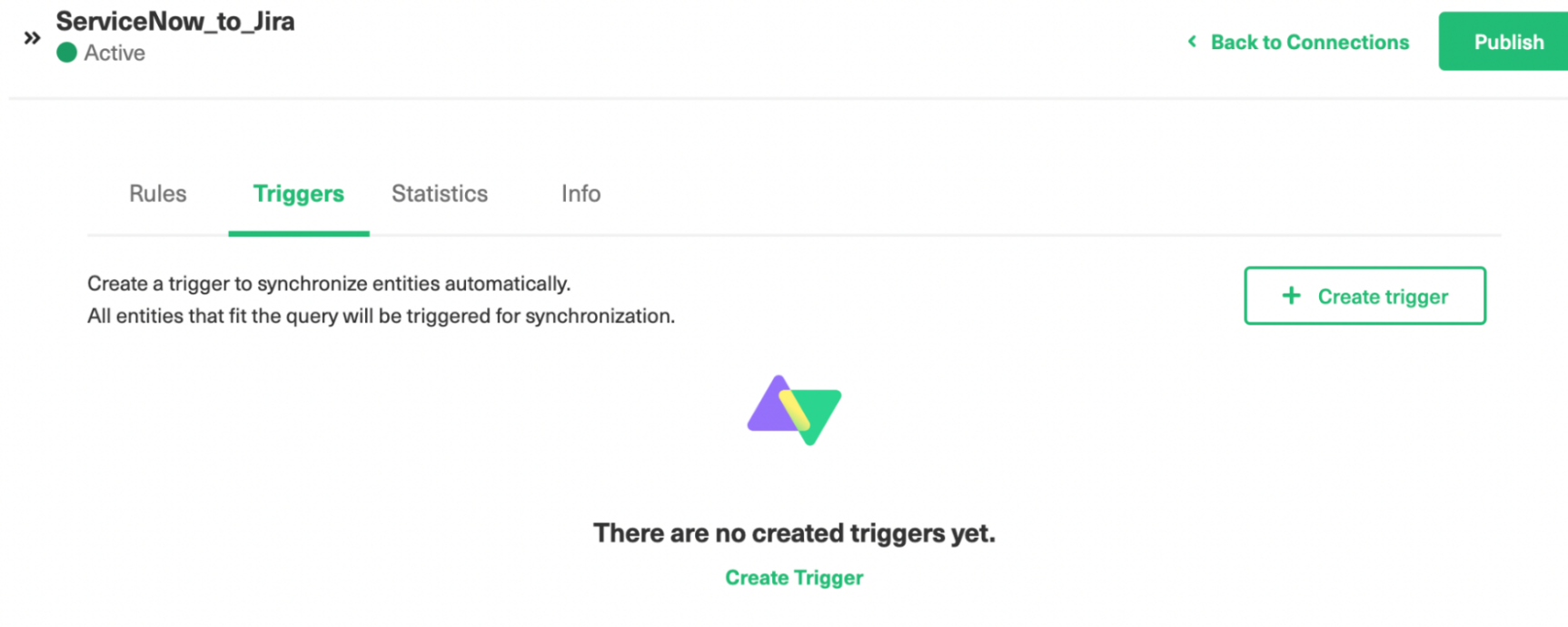
Step 4: Start Automatic Synchronization Using Triggers

Once you have decided what information must be sent and received, you might want to start the synchronization process automatically based on certain events.

This is done via Triggers in Exalate. Based on the conditions (filters) you set, sync will happen in accordance with the Sync Rules.

Triggers are platform-specific and need to be set on both sides of the integration if you want a two-way sync.

To create one, click the “Create Trigger” button.



The screenshot shows the configuration page for a ServiceNow to Jira integration. At the top left, the connection name is "ServiceNow_to_Jira" with a status indicator "Active" (a green dot). To the right, there are two buttons: "Back to Connections" (a green link) and "Publish" (a green button). Below this is a horizontal navigation bar with four tabs: "Rules", "Triggers" (which is selected and highlighted with a green underline), "Statistics", and "Info". The main content area contains the text: "Create a trigger to synchronize entities automatically. All entities that fit the query will be triggered for synchronization." To the right of this text is a button labeled "+ Create trigger". In the center of the page, there is a large, semi-transparent graphic of the Exalate logo (two overlapping triangles, one purple and one green, with a yellow pencil tip) and the text: "There are no created triggers yet." Below this text is a green link labeled "Create Trigger".

Choose the entity you want to set the trigger for.

On the ServiceNow side, you can choose between incidents, RITMs, problems, change requests, etc. In the “If” section, use the ServiceNow search syntax. Entities that meet the conditions are automatically synced.

You can write advanced triggers like syncing incidents assigned to a particular person or syncing incidents that have a particular text in a comment or description.

Add trigger ✕

Specify a ServiceNow search query to synchronize entities automatically. All entities that fit the query will be triggered for synchronization. [Find more details.](#)

Trigger will apply to selected entity type* ⓘ

incident | ▾

If* ⓘ

urgency=1 ↻

Notes

Active?

Add

Once done, leave some notes and activate the trigger.

You can see the trigger you just created in the list.

You can choose to sync all the existing entities that satisfy the trigger conditions using the [“Bulk Exalate”](#) option.

After you have made changes to the Connection in Steps 3 and 4, hit the “Publish” button to apply them.

Step 5: Start the Synchronization

Honestly, there is nothing you need to do in terms of starting the sync.

It will start automatically based on the triggers you have set up. If for some reason it doesn't, grab some coffee and by the time you are back, it will be working just fine.

If there are some issues with the triggers or rules, you can also troubleshoot and find out what's going wrong.

Best Practices for ServiceNow Integrations

The possibilities with integrations are huge and you can go downhill in a short amount of time without realizing what went wrong.

So let's look at a few best practices and see how you can make the most out of your ServiceNow Integration.

To be honest, these are not specific to only ServiceNow, they stand true for all kinds of integrations.

- Set a goal for your integration and define the requirements early on.
Ask questions like:

- Do I really need integration in the first place?
- What other ways are there to connect the platforms I use every day?
- What business value will the integration bring?
- Which processes will the integration have an impact on?

If all the answers point towards integration, you know you have headed the correct way.

- Stressing on the above point a little, make sure you chalk down to the finest details what data needs to be exchanged, at what time, what workflow is expected to be automated, how it is mapped to the destination instance, and so on.
- Involve the right stakeholders in the process from the beginning. Right from the inception, design, and implementation to the maintenance of the integration. They can be the platform admins, the implementation team, data architects, solution architects, information security consultants, or absolutely anyone you think can add value to the project.
- Explore the technologies and options available for integrating your systems. We have discussed ServiceNow integrations here (and I hope it brought some clarity) but they might as well be [Jira integrations](#) or [Salesforce integrations](#). This is an important step to put your integration in full throttle.
Also, check how the integration handles your data and GDPR requirements since it is an important part of an integration.
- Make sure to not rush through the integration. It is a time-consuming process, accept it, especially the trial and error or the POC part. So give it that kind of time. Your integration can be complex and can take time to set up. Acknowledge that instead of rushing through and being left with a clumsily-working high-maintenance result.
- Sometimes, you can have obvious solutions to your problems. For instance, JDBC works really well if you want a one-way integration between a data source and ServiceNow, i.e just pulling the data from the source and displaying it in ServiceNow. For two-way integrations between different platforms, you can use a web service or a MID server, or a third-party app like Exalate.
- Determine how and where the system you want to connect with is located. Is it inside the customer's network and needs update via a MID server or is it across company borders? You can involve a subject matter expert and get the right direction in this case.
- With ServiceNow, you already have basic system integrations like eBonding, google maps, etc. Understand and use these technologies. For advanced integrations, go to the experts.

- Some tools like IntegrationHub support workflow automation and focus on that alone whereas some others like Exalate focus on deeper integrations that involve a lot of advanced integration requirements. Be sure to have this understanding from the beginning. If in doubt, check how you can make the right decision between [IntegrationHub and Exalate](#).

There is a lot that can happen with integrations and the sky's the limit. And with the right approach, it becomes unstoppable. So choose wisely and decide what works best for you.

Conclusion

We have come a long way. Having started with understanding the needs and benefits of ServiceNow Integration, we also saw a few use cases to get a practical view of how such integrations can be envisioned.

You also learned all about the native ServiceNow-provided integration technologies and methods. As opposed to these, we had a look at why 3-rd party applications can also be a good choice.

We then briefly touched on an integration solution called Exalate that has a distributed architecture and supports decentralized integration, an intuitive scripting engine, and so on.

Finally, we saw how ServiceNow integrations can be implemented using Exalate before closing the journey with a few best practices.

There is always more to learn, so feel free to [book a demo](#) with Exalate if you think it can be the right solution for you and see where that leads you.

Recommended Reads:

- [ServiceNow to ServiceNow Integration: The Step-by-Step Guide to Setting up a Two-Way Sync](#)
- [Jira ServiceNow Integration: How to Set up an Integration in 6 Steps](#)

- [Salesforce Integrations: Integrate Salesforce and other Tools](#)
- [Jira Integrations: Integrate Jira and Other Systems Bidirectionally](#)
- [How to Set up a Salesforce ServiceNow Integration \(Step-by-Step Guide\)](#)