

How can I enable multitenancy using domain separation?

Questions addressed:

Page 1:

- What is domain separation?
- What are the benefits of domain separation?

Page 2:

- Is domain separation right for my organization?
- Are there any downsides to domain separation?
- Does each customer require its own domain?
- Can each domain have its own URL?

Page 3:

- How do I get started with domain separation?
- What are the best practices for using domain separation?

[Domain separation](#) (i.e., multitenancy or subtenancy) is a platform function that allows you to separate data, processes, and the user interface into logical groups called domains within one instance. With this function, you can manage demand at scale by using global settings across domains and making minor changes to accommodate domain-specific requirements. Domain separation is most used by [managed service providers](#) (MSPs). Each domain allows for unique configuration across three categories:

Data	Process (process administration)	User interface
<ul style="list-style-type: none"> • Customers, agents, fulfillers, and end users only see data in domains based on their visibility settings. • Data can be partitioned based on most tables in the system (e.g., department or company). 	<ul style="list-style-type: none"> • Business process logic may be overridden and customized for a specific domain. • Service providers can meet contractual obligations that differ from customer to customer. • Examples include assignment and approval rules, SLA management, and client scripts. 	<ul style="list-style-type: none"> • Application menus, lists, forms, homepages, and dashboards can be overridden and customized for a specific domain. • Service providers can alter the branding and UI elements displayed to meet individual customer needs.

What are the benefits of domain separation?

There are benefits for both the instance owner and tenants (customers of the instance owner):

Instance owner	<ul style="list-style-type: none"> • Economies of scale and efficiencies from having strong universal process standards and design, strict governance, purposeful integrations, and centralized instance administration • Ability to provide tiered services and/or serve customers in varying verticals by creating multiple process domains • Unlike alternatives to domain separation (e.g., business rules), no regression testing against new releases required for out-of-the-box configuration since domain separation is built into the platform
Instance tenants	<ul style="list-style-type: none"> • Quick onboarding and return on investment since there is no need to implement a full instance • Reduced administrative overhead and staffing requirements since the instance is centrally administered • Ability to use pre-built processes, features, and business services provided by the instance owner

How can I enable multitenancy using domain separation? (Cont.)

Is domain separation right for my organization?

Common use cases:	Used by customers that require:
<ul style="list-style-type: none">• Global outsourcers and MSPs who support multiple customers in one instance• Global organizations with unique geographical processes and data access requirements• Customers with complex multiple vendor or supplier management requirements	<ul style="list-style-type: none">❑ Contractual agreement fulfillment – Enforcing data separation at an application level between business entities❑ Central administration – Using a single instance to maintain global process and reporting❑ Process consistency – Minor business process or user interface customizations by domain (tenant)

Explore using unique instances or [alternatives to domain separation](#), which may be less complex and easier to maintain, if any of the following are true:

- Tenants want to administer the instance themselves
- Tenants have diverse process and UI needs
- Tenants are large enough that their transactions or data stored would require multiple nodes and/or dedicated hardware
- Your departments in your organization want to isolate records
- There are strict regulatory requirements for data that require separate databases be used (complete physical data separation)

Are there any downsides to domain separation?

- Depending on the setup, domain separation can require significant administrative overhead, such as [consistent monitoring and configuration](#) to ensure users only see what they're supposed to.
- Domain separation is irreversible. While it can be disabled, trickle-down effects, such as adding specific fields to queries are challenging to reverse.
- Custom tables will likely require additional configuration for data separation. Further, [not all applications nor features support domain separation](#) upon initial release.
- Some platform elements are global in scope and can't be separately defined at the domain level (e.g., CMDB data model and search setup).
- [Performance](#) might be affected as each query in the instance will have an additional domain clause. Most performance effects are insignificant.

Does each customer require its own domain?

The best practice is to be consistent with data management and establish one domain per customer regardless of their interaction level with the instance.

For example, if a customer rarely interacts with the domain separated instance today but would like to use it differently in the future, it's best to already have their data separate from other customers.

Can each domain have its own URL?

No. [Custom URLs](#) are supported per service portal but not by domain.

How can I enable multitenancy using domain separation? (Cont.)

How do I get started with domain separation?

1 Enable the plugin.

- Contact your account executive to purchase a subscription and enable the Domain Support – Domain Extensions Installer plugin.
- Try the plugin for *free* in a personal or developer instance.

2 Design and socialize your hierarchy.

- Consult internal and ServiceNow® architects or a [certified partner](#) to design your [domain hierarchy](#).
- Consider both geographic needs and service offerings. Remember that [data rolls up and process flows down](#).
- Socialize the hierarchy to stakeholders and ensure alignment. Adjusting the hierarchy is possible but cumbersome.

3 Build a [governance](#) process.

- Work with your stakeholders to document the process for how to adjust global processes and evaluate demands from tenants.
- Align on technical standards, such as naming conventions for domains.

4 Plan for ongoing maintenance.

- Assign someone to run audits using the [Domain Separation Center](#).
- Determine how domain separation will be enforced—e.g., by assigning users to a domain or to a company, using business rules or using modules to assign domains.

What are the best practices for using domain separation?

- ✓ **Minimize the overall number of domains and enable the [UI16 domain picker](#).** Unnecessary domains can impact instance performance, especially when using the domain picker. The UI16 domain picker will load domains as a user types them instead of loading all domains at once.
- ✓ **Maintain [out-of-the-box configuration](#) in the global processes.** Create a top-level process domain for any necessary overrides for the instance.
- ✓ **[Configure a default domain](#)** to prevent records from falling to the global domain. Admins will need to move records from the default domain.
- ✓ **Use [Visibility and Contains domains](#) thoughtfully.** Excessive use of these can slow performance. Instead, make sure your domain hierarchy is optimal.
- ✓ **Conduct application testing from a domain**, not in the global settings since overrides do not process in global.
- ✓ **Make most configurations in your top-level process domain** as opposed to tenant domains. This will make maintaining and scaling your instance easier.

Related resources

- [Developer Portal – Domain Separation](#)
- [Now Learning – Domain Separation Implementation training](#)
- [Product Docs – Recommended practices for service providers](#)
- [HI – Domain separating 'non out-of-the-box separated' tables](#)
- [Community – Domain Separation concepts](#)
- [YouTube – Domain Separation in ServiceNow](#)

If you have any questions on this topic or you would like to be a contributor to future ServiceNow best practice content, please [contact us](#).