



Restriction Rules Developer Guide

Version 63.0, Spring '25



CONTENTS

Chapter 1: About This Guide	1
Restriction Rules	2
Quick Start: Control Which Tasks the Sales Team Can Access	4
Before You Start	5
Create a Restriction Rule Using the Tooling API	5
Create the Restriction Rule Using the Metadata API	7
Considerations	9
Example Scenarios	12
Allow Users to See Only Specified Record Type	12
Allow Users to See Only Records That They Own	13
Allow Users to See Only Records Owned by Same Role	14
Allow Users to See Only Records Owned by Same Profile	15
Allow Users to See Records Based on a Custom Field	16
Allow Users to See an External Object's Records	16
Provide User Access With Multiple String or ID Values in Record Criteria	17
Tooling API Reference	19
RestrictionRule	19
Metadata API Reference	23
RestrictionRule	24

CHAPTER 1 About This Guide

In this chapter ...

- [Restriction Rules](#)
- [Quick Start: Control Which Tasks the Sales Team Can Access](#)
- [Considerations](#)
- [Example Scenarios](#)
- [Tooling API Reference](#)
- [Metadata API Reference](#)

Restriction rules are a record-level access control mechanism that allows you to grant granular visibility to specified users. This guide outlines the different use cases of restriction rules and describes how to set them up using the Tooling or Metadata API.

Restriction rules are available in Lightning Experience in Enterprise, Performance, Unlimited, and Developer editions.

Before creating restriction rules, turn off Salesforce Classic for your org. Find instructions at [Turn Off Salesforce Classic for Your Org](#).

Set up restriction rules in Salesforce by navigating to a supported object in the Object Manager.

You can provide feedback and suggestions for restriction rules in the [Restriction Rules](#) group in the Trailblazer Community.

SEE ALSO:

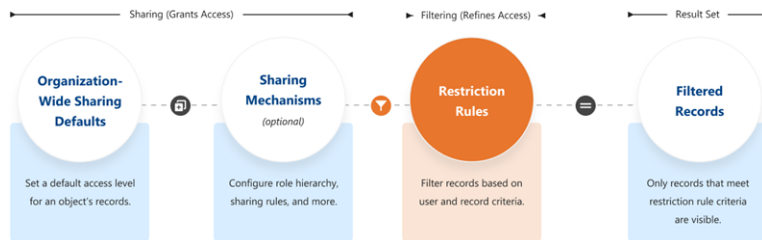
[Salesforce Help: Restriction Rules](#)

Restriction Rules

Restriction rules let you enhance your security by allowing certain users to access only specified records. They can prevent users from accessing records that can contain sensitive data or information that isn't essential to their work. Restriction rules are available for custom objects, external objects, contracts, events, tasks, time sheets, and time sheet entries and can be configured in the Object Manager or through the Tooling or Metadata API.

EDITIONS

Available in: Lightning Experience in **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions



When a restriction rule is applied to a user, the records that the user is granted access to via org-wide defaults, sharing rules, and other sharing mechanisms are filtered by criteria that you specify. For example, if users navigate to the Today's Tasks tab or to a list view for activities, they see only the records that meet the restriction rule's criteria. If a user has a link to a record that is no longer accessible after a restriction rule is applied, the user sees an error message.

Note: Before setting up a restriction rule on an external object, review these considerations.

- Restriction rules for external objects don't include organization-wide defaults or sharing mechanisms.
- Only external objects created using the Salesforce Connect: OData 2.0, OData 4.0, and Cross-Org adapters support restriction rules.
- External objects created using the Cross-Org adapter don't support search or SOSL when a rule is applied to a user. Salesforce returns only search results that match the most recently viewed records.
- Disabling search on external objects is recommended.
- External objects created using the Salesforce Connect custom adapter aren't supported.



When Do I Use Restriction Rules?

Use restriction rules when you want certain users to see only a specific set of records. Restriction rules can simplify controlling access to records with sensitive or confidential information. Access to contracts, tasks, and events can be difficult to make truly private using organization-wide defaults, making restriction rules the best way to configure this visibility.

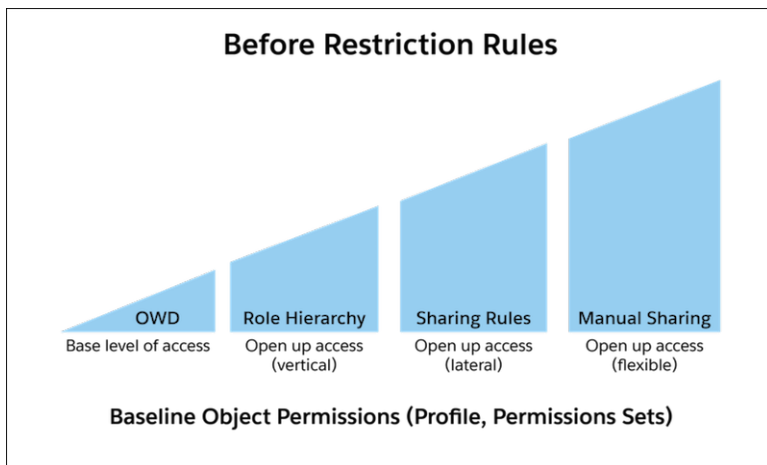
For example, you have competing sales teams that can't see each other's activities, even though these activities are on the same account. With restriction rules, you can make sure that sales teams see only activities that belong to them and are relevant to their work. Or, if you provide confidential services to various individuals, use restriction rules so that only team members responsible for supporting these individuals can see related tasks.

When creating more than one restriction or scoping rule, configure the rules so that only one active rule applies to a given user. Salesforce doesn't validate that only one active rule applies for a given user. If you create two active rules, and both rules apply to a given user, only one of the active rules is observed.

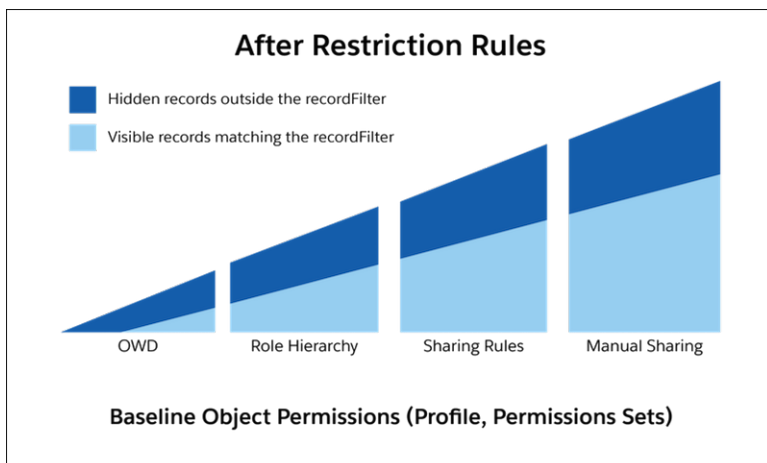
Before creating restriction rules, we recommend that you [Turn Off Salesforce Classic for Your Org](#). Salesforce can't guarantee that restriction rules work as intended for end users who are in the Salesforce Classic experience.

How Do Restriction Rules Affect Other Sharing Settings?

Users get access to records based on your organization-wide defaults and other sharing mechanisms, such as sharing rules or enterprise territory management.



When a restriction rule is applied to a user, the data that they had read access to via your sharing settings is further scoped to only records matching the `recordFilter`. This behavior is similar to how you can filter results in a list view or report, except that it's permanent. The number of records visible to the user can vary greatly depending on the value that you set in the `recordFilter`.



How Do I Configure Restriction Rules?

You can create and manage restriction rules by navigating to a supported object in the Object Manager or using either the Tooling API or Metadata API. You can create up to two active restriction rules per object in Enterprise and Developer Editions and up to five active restriction rules per object in Performance and Unlimited Editions.

Where Are Restriction Rules Available?

Restriction rules are available for custom objects, external objects, contracts, events, tasks, time sheets, and time sheet entries. Restriction rules are applied to the following Salesforce features:

- List Views
- Lookups
- Related Lists
- Reports
- Search
- SOQL
- SOSL

SEE ALSO:

[Salesforce Help: Restriction Rules](#)

Quick Start: Control Which Tasks the Sales Team Can Access

In this Quick Start, we create a restriction rule that controls which tasks members of the Sales Team can access.

[Before You Start](#)

Before you create the restriction rule, make sure you have the needed permissions and tools.

[Create a Restriction Rule Using the Tooling API](#)

Create a restriction rule that controls which tasks members of the Sales Team can access using the RestrictionRule Tooling API object.

[Create the Restriction Rule Using the Metadata API](#)

Create a restriction rule that controls which tasks members of the Sales Team can access using the RestrictionRule Metadata API type.

EDITIONS

Available in: Lightning Experience in **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Before You Start

Before you create the restriction rule, make sure you have the needed permissions and tools.

Before creating restriction rules, we recommend that you [Turn Off Salesforce Classic for Your Org](#). Salesforce can't guarantee that restriction rules work as intended for end users who are in the Salesforce Classic experience.

You can use either the [Tooling](#) or [Metadata](#) API to create, retrieve, update, and delete restriction rules. In this Quick Start, we show you how to use both.

SEE ALSO:

[Salesforce Developers Blog: Explore the Salesforce APIs with a Postman Collection](#)

Create a Restriction Rule Using the Tooling API

Create a restriction rule that controls which tasks members of the Sales Team can access using the RestrictionRule Tooling API object.

! **Important:** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

You can create up to two restriction rules per object in Enterprise and Developer Editions and up to five restriction rules per object in Performance and Unlimited Editions.

1. Set a value for the `FullName` value (the full name of the associated metadata object in Metadata API). We don't support two consecutive underscores in the `FullName` field.
2. Include all other required fields. For more information, see the reference topic [RestrictionRule](#).

For our example, we set the fields as follows:

```
{
  "FullName": "restrictionrulesalesteam",
  "Metadata": {
    "active": true,
    "description": "Sales team can see only task records with specified record type",
    "enforcementType": "Restrict",
    "masterLabel": "Sales Team Record Type",
    "recordFilter": "recordTypeId = '011xxxxxxxxxxxxx'",
  }
}
```

EDITIONS

Available in: Lightning Experience in **Enterprise, Performance, Unlimited,** and **Developer** Editions

USER PERMISSIONS

To create and manage restriction rules:

- Manage Sharing

To view restriction rules:

- View Setup & Configuration AND View Restriction and Scoping Rules

EDITIONS

Available in: Lightning Experience in **Enterprise, Performance, Unlimited,** and **Developer** Editions

USER PERMISSIONS

To create and manage restriction rules:

- Manage Sharing

To view restriction rules:

- View Setup & Configuration AND View Restriction and Scoping Rules

```

    "targetEntity": "Task",
    "userCriteria": "$User.ProfileId = '00xxxxxxxxxxxx'",
    "version": 1
  }
}

```

3. Use a POST request to create the restriction rule.
POST /services/data/v55.0/tooling/subjects/RestrictionRule
4. Copy your restriction rule definition into the request body.
5. Execute your request. Note the ID returned for the newly created restriction rule for later reference.

Retrieve and Update Information

Use the GET, PATCH, and DELETE methods to retrieve, update, and delete restriction rules.

Retrieve and Update Information

Use the GET, PATCH, and DELETE methods to retrieve, update, and delete restriction rules.

! **Important:** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

EDITIONS

Available in: Lightning Experience in **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Retrieve

To retrieve information about your restriction rule, use the GET method.

Example HTTP Method and URI:

GET

/services/data/v55.0/tooling/query/?q=SELECT+id,+targetEntity,+enforcementType,+recordFilter,+userCriteria+FROM+RestrictionRule

Update

To update the restriction rule, use the PATCH method.

We recommend that you don't update the value of `targetEntity` after the restriction rule is created. Instead, delete the restriction rule and create another one with the correct values.

Example HTTP Method and URI:

PATCH /services/data/v55.0/tooling/subjects/RestrictionRule/0eYxxxxxxxxxxxxx2AY

Replace `0eYxxxxxxxxxxxxx2AY` with the ID returned when creating the restriction rule.

Example Request Body:

All Metadata fields must be included, even if you aren't updating them. Specify the `FullName` value only if you're changing this field.

In this example, we deactivate the restriction rule by setting `active` to `false`.

```

{
  "Metadata": {
    "active": false,
    "description": "Sales team can see only task records with specified record type",

```

```

    "enforcementType": "Restrict",
    "masterLabel": "Sales Team Record Type",
    "recordFilter": "recordTypeId = '01lxxxxxxxxxxxx'",
    "targetEntity": "Task",
    "userCriteria": "$User.ProfileId = '00exxxxxxxxxxxx'",
    "version": 1
  }
}

```


Delete

To delete a restriction rule, use the DELETE method.

Example HTTP Method and URI:


```
DELETE /services/data/v55.0/tooling/subjects/RestrictionRule/0eYxxxxxxxxxxxxx2AY
```

Replace 0eYxxxxxxxxxxxxx2AY with the ID returned when creating the restriction rule.

 **Note:** If you include Salesforce org IDs in your `userCriteria` or `recordCriteria` fields, you must modify these IDs before deploying to the target org if different from the org where the restriction rules were retrieved.

Create the Restriction Rule Using the Metadata API

Create a restriction rule that controls which tasks members of the Sales Team can access using the RestrictionRule Metadata API type.

 **Important:** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

You can create up to two restriction rules per object in Enterprise and Developer Editions and up to five restriction rules per object in Performance and Unlimited Editions.

1. Set up the `package.xml` manifest file and your directory.

Example `package.xml` file:

```

<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>*</members>
    <name>RestrictionRule</name>
  </types>
  <version>55.0</version>
</Package>

```

EDITIONS

Available in: Lightning Experience in **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create and manage restriction rules:

- Manage Sharing

To view restriction rules:

- View Setup & Configuration AND View Restriction and Scoping Rules

Example directory:

```
myPackage/package.xml
myPackage/restrictionRules
myPackage/restrictionRules/Rule1.rule
myPackage/restrictionRules/Rule2.rule
```

2. Include all required fields. For more information, see the reference topic [RestrictionRule](#).

For our example, we set the fields as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<RestrictionRule xmlns="http://soap.sforce.com/2006/04/metadata">
  <active>true</active>
  <description>Sales team can see only task records with specified record
type</description>
  <enforcementType>Restrict</enforcementType>
  <masterLabel>Sales Team Record Type</masterLabel>
  <recordFilter>recordTypeId = '011xxxxxxxxxxxxx'</recordFilter>
  <targetEntity>Task</targetEntity>
  <userCriteria>$User.ProfileId = '00xxxxxxxxxxxxx'</userCriteria>
  <version>1</version>
</RestrictionRule>
```

3. Zip your directory and deploy your changes. For more information, see [Deploying and Retrieving Metadata](#) in the Metadata API Developer Guide.
4. Note the ID returned for the newly created restriction rule for later reference.

Retrieve and Update Information


Use the `deploy()` and `retrieve()` calls to move metadata (XML files) between a Salesforce organization and a local file system.

Retrieve and Update Information

Use the `deploy()` and `retrieve()` calls to move metadata (XML files) between a Salesforce organization and a local file system.

For more information, see [Deploying and Retrieving Metadata](#) in the Metadata API Developer Guide.

If you include Salesforce org IDs in your `userCriteria` or `recordCriteria` fields, you must modify these IDs before deploying to the target org if different from the org where the restriction rules were retrieved.

-  **Note:** We recommend that you don't update the value of `targetEntity` after the restriction rule is created. Instead, delete the restriction rule and create another one with the correct values.

EDITIONS

Available in: Lightning Experience in **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Considerations

Keep these considerations and limitations in mind while using restriction rules.

Available Objects

- Before creating restriction rules, we recommend that you [Turn Off Salesforce Classic for Your Org](#). Salesforce can't guarantee that restriction rules work as intended for end users who are in the Salesforce Classic experience.
- Restriction rules are available for custom objects, external objects, contracts, events, tasks, time sheets, and time sheet entries.
- In calendars, if the Show Details access level is selected, users can see the subject of all events, regardless of the restriction rules created. For more information, see [Share Your Calendar in Lightning Experience](#) in Salesforce Help.

EDITIONS


Available in: Lightning Experience in **Enterprise, Performance, Unlimited, and Developer** Editions

Applicable Features

- Restriction rules are applied to the following Salesforce features:
 - List Views
 - Lookups
 - Related Lists
 - Reports
 - Search
 - SOQL
 - SOSL
- Restriction rules support custom picklist values in record and user criteria. If you delete a custom picklist value used in a restriction rule, the rule no longer works as intended.
- Use the Activity Timeline instead of activity related lists, such as Open Activities or Activity History. If you use activity related lists, create rules on task or event objects using fields that are only available in the related lists.
- If you use Open Activities and Activity History related lists, when restriction rules are applied, it's possible that fewer than 50 records are displayed when more activities exist that the user has access to. This behavior occurs because these lists display at most 50 records, and restriction rules are applied after. This behavior is related to the known issue, [Limit of Fifty Records Visible in Related List View](#).
- Users can still see records they previously had access to in the search box shortcuts list or in the Recently Viewed list view after restriction rules are applied. When users click the record name, they can't access the record and get an error.
- Users can see their subordinates' events in calendars even if the users have an active restriction rule applied.
- If a user creates an event or a task record using the Chatter publisher, the record name is visible in the related Chatter post. Restriction rules don't restrict visibility to these record names.
- Users can't clone records that have a lookup to a record that they can't see due to a restriction rule. For example, you have a restriction rule that prevents a user from seeing a specific contract record, and the user tries to clone an order record that has a lookup to the contract record. The user gets an error, preventing the clone operation from succeeding.
- Restriction rules aren't applied for code executed in System Mode.
- Users with the View All Records or View All Data permissions can view all records regardless of restriction rules. Users with the Modify All Records or Modify All Data permissions can view, edit, and delete all records regardless of restriction rules.

- A user with a restriction rule applied might not find all possible matching results when searching for a record. For performance reasons, search crowding applies limits to the number of search results. The record the user is looking for can fall outside those limits. Learn how to adjust your searches for the best results at [How Search Crowding Affects Search Results](#).
- The [UserRecordAccess](#) object doesn't consider whether a user's access is blocked due to a restriction rule. If a user's access is blocked even though query results state that they should have access, check to see if a restriction rule on the object prevents the user's access.

Creating Restriction Rules

- You can create up to two active restriction rules per object in Enterprise and Developer Editions and up to five active restriction rules per object in Performance and Unlimited Editions.
 - Create only one restriction or scoping rule per object per user. In other words, for a given object, only one restriction or scoping rule at most should have the `userCriteria` field evaluate to `true` for a given user.
 - Creating a restriction rule for an object doesn't automatically restrict access to its child objects. For example, if you create a restriction rule for the Contract object, the access doesn't change for notes that are associated with the affected contract records. To secure these child objects, you must use other sharing mechanisms.
 - We recommend that you don't edit the `targetEntity` field after the restriction rule is created. Instead, delete the existing rule and create a restriction rule with the correct values.
 - When you reference the `Owner` field, you must specify the object type in your syntax. For example, the `Owner` field on an Event object can contain a user or a queue, but queues aren't supported in restriction rules. So it's necessary to specify `Owner:User` in the `recordFilter` syntax when the filter allows only users.
 - If you reference IDs in the `recordFilter` field, use 15-character IDs instead of 18-character IDs.
 - You can reference another object's field using dot notation in the `recordFilter` field. You can use only one "dot" (one lookup level from the `targetEntity`). For example, `Owner.UserRoleId`. See [Allow Users to See Only Records Owned by Same Role](#) and [Allow Users to See Only Records Owned by Same Profile](#) for examples.
 - We support these data types in the `recordFilter` and `userCriteria` fields:
 - boolean
 - date
 - dateTime
 - double
 - int
 - reference
 - string
 - time
 - single picklist
-  **Note:** Comma-separated ID or string values are supported in the Record Criteria field.
- Including a null or blank value in record criteria isn't supported and can result in unexpected behavior.
 - Restriction rules support only the EQUALS operator. The AND, OR, or any other operators aren't supported.
 - The use of formulas isn't supported.
 - Don't create rules on `Event.IsGroupEvent`, which indicates whether the event has invitees.
 - You can use a change set or unlocked package to move restriction rules from one org to another.

- If you include IDs in your `recordFilter` or `userCriteria` fields that are specific to your Salesforce org (such as a role, record type, or profile ID), you must modify these IDs in the target org if different from the org where the restriction rules were originally created. Keep this consideration in mind if deploying rules between sandboxes or to a production org.

Restriction Rules and External Objects

- Only external objects created using the Salesforce Connect: OData 2.0, OData 4.0, and Cross-Org adapters support restriction rules.
- External objects created using the Cross-Org adapter don't support search or SOSL when a rule is applied to a user. Salesforce returns only search results that match the most recently viewed records.
- External objects created using the Salesforce Connect custom adapter aren't supported.
- External object record data is stored outside Salesforce. Admins are responsible for ensuring that rules they create on external objects don't negatively impact performance in Salesforce or in the external system.

Important:

- Editing or deleting a restriction rule on an external object causes an additional database call, which can result in additional billing when the external data source bills per call.
- When search is enabled for external object records, searching requires additional database calls each time. Avoid additional charges by turning off search for external object records.

As with all restriction rules, using only object fields that are indexed is recommended, especially in record criteria.

- Using external IDs in record criteria isn't recommended.
- Restriction rules for external objects don't include organization-wide defaults or sharing mechanisms.
- External objects don't appear in Object Manager. To navigate to an external object, enter *External Data Sources* in the Quick Find box in Setup, then select **External Data Sources**. Select an external object from the list view on this page.

 **Note:** You can also find external objects in the Most Recently Used list in Setup.

Performance Considerations

Restriction rules were built to support sharing needs in a performant way. Your data volume and architecture are also factors in rule performance.

- To test a rule's performance impact, take the record criteria to your API client of choice and run the query. If it's fast for a given user, the rule is likely to run efficiently. For objects with large data volumes, add three to five percent overhead to the record filter's performance.
- If it isn't performant, isolate the field that is slowing performance. Work with Salesforce customer support to find out if the field can be indexed.

SEE ALSO:

[Knowledge Article: Improve Performance of SOQL Queries using a Custom Index](#)

Example Scenarios

Refer to these sample restriction rules that fulfill different access requirements. To review how these example scenarios are built in Salesforce, see [Restriction Rule Example Scenarios](#) in Salesforce Help.

[Allow Users to See Only Specified Record Type](#)

This restriction rule allows the designated users to see only contracts that have a specified record type.

[Allow Users to See Only Records That They Own](#)

This restriction rule allows users with the designated profile to see only the tasks that they own.

[Allow Users to See Only Records Owned by Same Role](#)

This restriction rule allows active users to see only the events owned by users that have the same role. You can use dot notation to traverse object and field relationships in the `recordFilter` field.

[Allow Users to See Only Records Owned by Same Profile](#)

This restriction rule allows active users to see only the events owned by users that have the same profile. You can use dot notation to traverse object and field relationships in the `recordFilter` field.

[Allow Users to See Records Based on a Custom Field](#)

This restriction rule allows high-volume users to see only the contracts where the user's department matches the contract's department. This rule uses a custom field, `Department__c`, that must have the appropriate value set through Apex, Process Builder, workflows, or flows.

[Allow Users to See an External Object's Records](#)

This restriction rule allows active Salesforce users to see the records of an external object called Purchase Order. The rule uses a field called `IsClosed` on Purchase Order records in its record criteria.

[Provide User Access With Multiple String or ID Values in Record Criteria](#)

This restriction rule allows active users to see records whose `Name__c` field matches the rule's record criteria value. The record criteria contains strings separated by a comma. ID values are also supported. Double-quotes specify that the value inside the quotes isn't considered a delimiter.

EDITIONS


Available in: Lightning Experience in **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

SEE ALSO:

[Salesforce Help: Restriction Rule Example Scenarios](#)

Allow Users to See Only Specified Record Type

This restriction rule allows the designated users to see only contracts that have a specified record type.

 **Important:** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

EDITIONS

Available in: Lightning Experience in **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Tooling API

```
{
  "FullName": "Contract restriction rule on RecordType",
```



```

"Metadata": {
  "active":true,
  "description":"View Contracts with RecordType = Internal.",
  "enforcementType":"Restrict",
  "masterLabel":"RR for Internal Contracts",
  "recordFilter":"RecordTypeId = '012xxxxxxxxxxxx'",
  "targetEntity":"Contract",
  "userCriteria":"$User.UserRoleId = '00Exxxxxxxxxxxx'",
  "version":1
}
}

```

Metadata API

```

<?xml version="1.0" encoding="UTF-8"?>
<RestrictionRule xmlns="http://soap.sforce.com/2006/04/metadata">
  <active>true</active>
  <description>View Contracts with RecordType = Internal.</description>
  <enforcementType>Restrict</enforcementType>
  <masterLabel>RR for Internal Contracts</masterLabel>
  <recordFilter>RecordTypeId = '012xxxxxxxxxxxx'</recordFilter>
  <targetEntity>Contract</targetEntity>
  <userCriteria>$User.UserRoleId = '00Exxxxxxxxxxxx'</userCriteria>
  <version>1</version>
</RestrictionRule>

```

Allow Users to See Only Records That They Own

This restriction rule allows users with the designated profile to see only the tasks that they own.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

EDITIONS

Available in: Lightning Experience in **Enterprise, Performance, Unlimited,** and **Developer** Editions

Tooling API

```

{
  "FullName":"restriction_rule_tasks_you_own",
  "Metadata": {
    "active":true,
    "description":"Allows users of a specific profile to see only tasks that they own.",
    "enforcementType":"Restrict",
    "masterLabel":"Tasks You Own",
    "recordFilter":"OwnerId = $User.Id",
    "targetEntity":"Task",
    "userCriteria":"$User.ProfileId = '00exxxxxxxxxxxx'",
    "version":1
  }
}

```

Metadata API

```
<?xml version="1.0" encoding="UTF-8"?>
<RestrictionRule xmlns="http://soap.sforce.com/2006/04/metadata">
  <active>true</active>
  <description>Allows users with a specific profile to see only tasks that they
own.</description>
  <enforcementType>Restrict</enforcementType>
  <masterLabel>Tasks You Own</masterLabel>
  <recordFilter>OwnerId = $User.Id</recordFilter>
  <targetEntity>Task</targetEntity>
  <userCriteria>$User.ProfileId = '00exxxxxxxxxxxxx'</userCriteria>
  <version>1</version>
</RestrictionRule>
```

Allow Users to See Only Records Owned by Same Role

This restriction rule allows active users to see only the events owned by users that have the same role. You can use dot notation to traverse object and field relationships in the `recordFilter` field.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

EDITIONS

Available in: Lightning Experience in **Enterprise, Performance, Unlimited,** and **Developer** Editions

Tooling API

```
{
  "FullName": "restrictionruleeventsrole",
  "Metadata": {
    "active": true,
    "description": "Allows active users to see only events owned by users of the same
role.",
    "enforcementType": "Restrict",
    "masterLabel": "Events Owned by Same Role",
    "recordFilter": "Owner:User.RoleId = $User.RoleId",
    "targetEntity": "Event",
    "userCriteria": "$User.IsActive = true",
    "version": 1
  }
}
```

Metadata API

```
<?xml version="1.0" encoding="UTF-8"?>
<RestrictionRule xmlns="http://soap.sforce.com/2006/04/metadata">
  <active>true</active>
  <description>Allows active users to see only events owned by users of the same
role.</description>
  <enforcementType>Restrict</enforcementType>
  <masterLabel>Events Owned by Same Role</masterLabel>
```


```

<recordFilter>Owner:User.UserId = $User.UserId</recordFilter>
<targetEntity>Event</targetEntity>
<userCriteria>$User.IsActive = true</userCriteria>
<version>1</version>
</RestrictionRule>

```

Allow Users to See Only Records Owned by Same Profile

This restriction rule allows active users to see only the events owned by users that have the same profile. You can use dot notation to traverse object and field relationships in the `recordFilter` field.

 **Important:** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

EDITIONS

Available in: Lightning Experience in **Enterprise, Performance, Unlimited,** and **Developer** Editions

Tooling API

```

{
  "FullName": "restictionruleeventsprofile",
  "Metadata": {
    "active": true,
    "description": "Allows active users to see only events owned by users of the same profile.",
    "enforcementType": "Restrict",
    "masterLabel": "Events Owned by Same Profile",
    "recordFilter": "Owner:User.ProfileId = $User.ProfileId",
    "targetEntity": "Event",
    "userCriteria": "$User.IsActive = true",
    "version": 1
  }
}

```

Metadata API

```

<?xml version="1.0" encoding="UTF-8"?>
<RestrictionRule xmlns="http://soap.sforce.com/2006/04/metadata">
  <active>true</active>
  <description>Allows active users to see only events owned by users of the same profile.</description>
  <enforcementType>Restrict</enforcementType>
  <masterLabel>Events Owned by Same Profile</masterLabel>
  <recordFilter>Owner:User.ProfileId = $User.ProfileId</recordFilter>
  <targetEntity>Event</targetEntity>
  <userCriteria>$User.IsActive = true</userCriteria>
  <version>1</version>
</RestrictionRule>

```

Allow Users to See Records Based on a Custom Field

This restriction rule allows high-volume users to see only the contracts where the user's department matches the contract's department. This rule uses a custom field, `Department__c`, that must have the appropriate value set through Apex, Process Builder, workflows, or flows.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

EDITIONS

Available in: Lightning Experience in **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Tooling API

```
{
  "FullName": "Contract restriction rule for Customer Community",
  "Metadata": {
    "active": true,
    "description": "Show high-volume user by department",
    "enforcementType": "Restrict",
    "masterLabel": "RR for Internal Contracts",
    "recordFilter": "Department__c = $User.Department",
    "targetEntity": "Contract",
    "userCriteria": "$User.UserType = 'CSPLitePortal'",
    "version": 1
  }
}
```

Metadata API

```
<?xml version="1.0" encoding="UTF-8"?>
<RestrictionRule xmlns="http://soap.sforce.com/2006/04/metadata">
  <active>true</active>
  <description>Show high-volume user by department</description>
  <enforcementType>Restrict</enforcementType>
  <masterLabel>RR for Internal Contracts</masterLabel>
  <recordFilter>Department__c = $User.Department</recordFilter>
  <targetEntity>Contract</targetEntity>
  <userCriteria>$User.UserType = 'CSPLitePortal'</userCriteria>
  <version>1</version>
</RestrictionRule>
```

Allow Users to See an External Object's Records

This restriction rule allows active Salesforce users to see the records of an external object called Purchase Order. The rule uses a field called `IsClosed` on Purchase Order records in its record criteria.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Note: Only external objects created using the Salesforce Connect: OData 2.0, OData 4.0, and Cross-Org adapters support restriction rules. Find out more in Restriction Rule Considerations.

EDITIONS

Available in: Lightning Experience in **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Tooling API

```
{
  "FullName": "purchase_orders_restrictionrule",
  "Metadata": {
    "active": true,
    "description": "Allows accounting department users to access open purchase order records from external system.",
    "enforcementType": "Restrict",
    "masterLabel": "OpenPurchaseOrderRecords",
    "recordFilter": "IsClosed__c = 'false'",
    "targetEntity": "PurchaseOrder__x",
    "userCriteria": "$User.Department = 'Accounting'",
    "version": 1
  }
}
```

Metadata API

```
<?xml version="1.0" encoding="UTF-8"?>
<RestrictionRule xmlns="http://soap.sforce.com/2006/04/metadata">
  <active>true</active>
  <description>Allows accounting department users to access open purchase order records from external system.</description>
  <enforcementType>Restrict</enforcementType>
  <masterLabel>OpenPurchaseOrderRecords</masterLabel>
  <recordFilter>IsClosed__c = 'false'</recordFilter>
  <targetEntity>PurchaseOrder__x</targetEntity>
  <userCriteria>$User.Department = 'Accounting'</userCriteria>
  <version>1</version>
</RestrictionRule>
```

SEE ALSO:

[Salesforce Help: Restriction Rule Considerations](#)

[Salesforce Help: Define External Objects](#)

Provide User Access With Multiple String or ID Values in Record Criteria

This restriction rule allows active users to see records whose Name__c field matches the rule's record criteria value. The record criteria contains strings separated by a comma. ID values are also supported. Double-quotes specify that the value inside the quotes isn't considered a delimiter.

This rule uses a custom object called Agent__c with a custom text field called Name__c.

EDITIONS

Available in: Lightning Experience in **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Tooling API

```
{
  "FullName": "Agent records matching name field",
  "Metadata": {
    "active": true,
```

```

    "description": "Show Records Matching Name__c field",
    "enforcementType": "Restrict",
    "masterLabel": "Records Matching Name__c field",
    "recordFilter": "Name__c='Tom, Anita, \"Torres, Jia\"'",
    "targetEntity": "Agent__c",
    "userCriteria": "$User.IsActive=true",
    "version": 1
  }
}

```

Metadata API

```

<?xml version="1.0" encoding="UTF-8"?>
<RestrictionRule xmlns="http://soap.sforce.com/2006/04/metadata">
  <active>true</active>
  <description>Show Records Matching Name__c field</description>
  <enforcementType>Restrict</enforcementType>
  <masterLabel>Records Matching Name__c field</masterLabel>
  <recordFilter>Name__c='Tom, Anita, \"Torres, Jia\"</recordFilter>
  <targetEntity>Agent__c</targetEntity>
  <userCriteria>$User.IsActive=true</userCriteria>
  <version>1</version>
</RestrictionRule>

```

This restriction rule allows active users to see records owned by two different managers. In this example, the rule's record criteria contains IDs separated by a comma.

Tooling API

```

{
  "FullName": "Records Owned By Managers",
  "Metadata": {
    "active": true,
    "description": "Displays records owned by two department managers",
    "enforcementType": "Restrict",
    "masterLabel": "RR for manager records",
    "recordFilter": "Agent__c.Owner:User.ManagerId=001xx000003HNy7, 001xx000003HNut",
    "targetEntity": "Agent__c",
    "userCriteria": "$User.IsActive=true",
    "version": 1
  }
}

```

Metadata API

```

<?xml version="1.0" encoding="UTF-8"?>
<RestrictionRule xmlns="http://soap.sforce.com/2006/04/metadata">
  <active>true</active>
  <description>Displays records owned by two department managers</description>
  <enforcementType>Restrict</enforcementType>
  <masterLabel>RR for manager records</masterLabel>

```

```
<recordFilter>Agent__c.Owner:User.ManagerId=001xx000003HNY7,
001xx000003HNut</recordFilter>
<targetEntity>Agent__c</targetEntity>
<userCriteria>$User.IsActive=true</userCriteria>
<version>1</version>
</RestrictionRule>
```

Tooling API Reference

This section provides more information on the RestrictionRule Tooling API object used to create restriction rules.

EDITIONS

Available in: Lightning Experience in **Enterprise**, **Performance, Unlimited**, and **Developer** Editions

RestrictionRule

Represents a restriction rule or a scoping rule. A restriction rule has `EnforcementType` set to `Restrict` and controls the access that specified users have to designated records. A scoping rule has `EnforcementType` set to `Scoping` and controls the default records that your users see without restricting access.

RestrictionRule

Represents a restriction rule or a scoping rule. A restriction rule has `EnforcementType` set to `Restrict` and controls the access that specified users have to designated records. A scoping rule has `EnforcementType` set to `Scoping` and controls the default records that your users see without restricting access.

This object is available in API version 52.0 and later.

Supported SOAP API Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API Methods


`DELETE`, `GET`, `HEAD`, `PATCH`, `POST`, `Query`

Special Access Rules

Only users with the View Restriction and Scoping Rules permission can view restriction rules and scoping rules via the API. Only users with the Manage Sharing permission can view, create, update, and delete restriction rules and scoping rules.

Fields

Field	Details
Description	Type textarea

Field	Details
	<p>Properties Filter, Group, Nillable, Sort</p> <p>Description Required. The description of the rule.</p>
DeveloperName	<p>Type string</p> <p>Properties Filter, Group, Sort</p> <p>Description The unique name for the RestrictionRule object.</p> <p>This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is automatically generated, but you can supply your own value if you create the record using the API.</p> <p> Note: Only users with View DeveloperName OR View Setup and Configuration permission can view, group, sort, and filter this field.</p>
EnforcementType	<p>Type picklist</p> <p>Properties Defaulted on create, Filter, Group, Restricted picklist, Sort</p> <p>Description Required. The type of rule.</p> <p>Possible values are:</p> <ul style="list-style-type: none"> • <code>FieldRestrict</code>—Don't use. • <code>Restrict</code>—Restriction rule. • <code>Scoping</code>—Scoping rule.
FullName	<p>Type string</p> <p>Properties Create, Group, Nillable</p> <p>Description Required. The full name of the associated RestrictionRule in Metadata API. The full name can include a namespaceprefix.</p> <p>Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.</p>

Field	Details
IsActive	<p>Type boolean</p> <p>Properties Defaulted on create, Filter, Group, Sort</p> <p>Description Indicates whether the rule is active (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code>.</p>
Language	<p>Type picklist</p> <p>Properties Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort</p> <p>Description The language of the rule. The value for this field is the language value of the org.</p>
MasterLabel	<p>Type string</p> <p>Properties Filter, Group, Sort</p> <p>Description Label for the rule.</p>
Metadata	<p>Type <code>mns : RestrictionRule</code></p> <p>Properties Create, Nillable, Update</p> <p>Description The restriction rule's metadata.</p> <p>Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.</p>
RecordFilter	<p>Type textarea</p> <p>Properties Create, Filter, Group, Sort, Update</p> <p>Description Required. The criteria that determine which records are accessible via the rule.</p>
TargetEntity	<p>Type picklist</p>

Field	Details
	<p>Properties Filter, Group, Restricted picklist, Sort</p> <p>Description Required. The object for which you're creating the rule. We recommend that you don't edit this field after the rule is created.</p> <p>If <code>EnforcementType</code> is set to <code>Restrict</code>, custom objects, external objects, and these objects are supported:</p> <ul style="list-style-type: none"> • Contract • Event • Task • TimeSheet • TimeSheetEntry <p>If <code>EnforcementType</code> is set to <code>Scoping</code>, custom objects and these objects are supported:</p> <ul style="list-style-type: none"> • Account • Case • Contact • Event • Lead • Opportunity • Task
UserCriteria	<p>Type textarea</p> <p>Properties Create, Filter, Group, Sort, Update</p> <p>Description Required. The users that this rule applies to, such as all active users or users with a specified role or profile.</p>
Version	<p>Type int</p> <p>Properties Filter, Group, Sort</p> <p>Description Required. The rule's version number.</p>

Usage

The following is an example of a `RestrictionRule` representing a restriction rule.

```
{
  "FullName": "restriction_rule_tasks_you_own",
  "Metadata": {
    "active": true,
    "description": "Allows users of a specific profile to see only tasks that they own.",
    "enforcementType": "Restrict",
    "masterLabel": "Tasks You Own",
    "recordFilter": "OwnerId = $User.Id",
    "targetEntity": "Task",
    "userCriteria": "$User.ProfileId = '00exxxxxxxxxxxxx'",
    "version": 1
  }
}
```

The following is an example of a `RestrictionRule` representing a scoping rule.

```
{
  "FullName": "Department A contact scoping rule",
  "Metadata": {
    "active": true,
    "description": "View contacts from Department A.",
    "enforcementType": "Scoping",
    "masterLabel": "SR for Department A",
    "recordFilter": "Department=$User.Department",
    "targetEntity": "Contact",
    "userCriteria": "$User.UserRoleId = '00Exxxxxxxxxxxxx'",
    "version": 1
  }
}
```

Metadata API Reference

This section provides more information on the `RestrictionRule` Metadata API type used to create restriction rules.

[RestrictionRule](#)


Represents a restriction rule or a scoping rule. A restriction rule has `enforcementType` set to `Restrict` and controls the access that specified users have to designated records. A scoping rule has `enforcementType` set to `Scoping` and controls the default records that your users see without restricting access. This type extends the `Metadata` metadata type and inherits its `fullName` field.

EDITIONS

Available in: Lightning Experience in **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

RestrictionRule

Represents a restriction rule or a scoping rule. A restriction rule has `enforcementType` set to `Restrict` and controls the access that specified users have to designated records. A scoping rule has `enforcementType` set to `Scoping` and controls the default records that your users see without restricting access. This type extends the `Metadata metadata` type and inherits its `fullName` field.

 **Important:** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

File Suffix and Directory Location

RestrictionRule components have the suffix `.rule` and are stored in the `restrictionRules` folder.

Version

RestrictionRule components are available in API version 52.0 and later.

Special Access Rules

Only users with the View Restriction and Scoping Rules permission can view restriction rules and scoping rules via the API. Only users with the Manage Sharing permission can view, create, update, and delete restriction rules and scoping rules.

Fields

Field Name	Field Type	Description
<code>active</code>	boolean	Indicates whether the rule is active (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> .
<code>description</code>	string	Required. The description of the rule.
<code>enforcementType</code>	EnforcementType (enumeration of type string)	Required. The type of rule. Valid values are: <ul style="list-style-type: none"> <code>FieldRestrict</code>—Don't use. <code>Restrict</code>—Restriction rule. <code>Scoping</code>—Scoping rule.
<code>masterLabel</code>	string	Required. The name of the rule.
<code>recordFilter</code>	string	Required. The criteria that determine which records are accessible via the rule. For picklist fields, you can now use the OR operator to specify multiple values for a single picklist field.— For example: <pre>OR(ISPICKVAL(Status,'Draft'), ISPICKVAL(Status,'Activated'), ISPICKVAL(Status,'Negotiating'))</pre> <p>This enhancement applies to both Restriction Rules and Scoping Rules. Support for multiple picklist values using the OR operator was introduced in API version 60.0 and later.</p>

Field Name	Field Type	Description
targetEntity	string	<p>Required. The object for which you're creating the rule. We recommend that you don't edit this field after the rule is created.</p> <p>If <code>enforcementType</code> is set to <code>Restrict</code>, custom objects, external objects, and these objects are supported:</p> <ul style="list-style-type: none"> • Contract • Event • Task • TimeSheet • TimeSheetEntry <p>If <code>enforcementType</code> is set to <code>Scoping</code>, custom objects and these objects are supported:</p> <ul style="list-style-type: none"> • Account • Case • Contact • Event • Lead • Opportunity • Task
userCriteria	string	Required. The users that this rule applies to, such as all active users or users with a specified role or profile.
version	int	Required. The rule's version number.

Declarative Metadata Sample Definition

The following is an example of a `RestrictionRule` component representing a restriction rule.

```
<?xml version="1.0" encoding="UTF-8"?>
<RestrictionRule xmlns="http://soap.sforce.com/2006/04/metadata">
  <active>true</active>
  <description>Allows users with a specific profile to see only tasks that they
own.</description>
  <enforcementType>Restrict</enforcementType>
  <masterLabel>Tasks You Own</masterLabel>
  <recordFilter>OwnerId = $User.Id</recordFilter>
  <targetEntity>Task</targetEntity>
  <userCriteria>$User.ProfileId = '00exxxxxxxxxxxxxx'</userCriteria>
  <version>1</version>
</RestrictionRule>
```

The following is an example of a `RestrictionRule` component representing a scoping rule.

```
<?xml version="1.0" encoding="UTF-8"?>
<RestrictionRule xmlns="http://soap.sforce.com/2006/04/metadata">
  <active>true</active>
```

```
<description>View tasks contacts from Department A.</description>
<enforcementType>Scoping</enforcementType>
<masterLabel>SR for Department A contacts</masterLabel>
<recordFilter>Department=$User.Department</recordFilter>
<targetEntity>Contact</targetEntity>
<userCriteria>$User.UserRoleId = '00Exxxxxxxxxxxxx'</userCriteria>
<version>1</version>
</RestrictionRule>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>*</members>
    <name>RestrictionRule</name>
  </types>
  <version>55.0</version>
</Package>
```