ServiceNow Application Developer

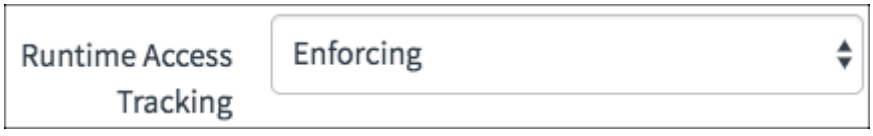
Securing Applications Against Access from Other Applications > Runtime Access Tracking

Use *Runtime Access Tracking* to manage script access to resources from other applications. This is known as cross-scope access because the records are in different application scopes. The settings are:

- ***None***: No authorization required for application scripts to access resources from other applications as long as the other application allows it. No record is created in the *Application Cross-Scope Access* table.
- ***Tracking***: Allows application scripts to access resources from other applications. A record for the access is automatically inserted in the *Application Cross-Scope Access* table with a *Status* value of *Allowed*. This is the default setting.
- ***Enforcing***: Allows application scripts to access resources from other applications only after an admin authorizes the access. A record is automatically added to the *Application Cross-Scope Access* table with a *Status* value of *Requested*.

To see the table of cross-scope authorizations, use the Application Navigator in the main ServiceNow browser window to open **System Applications > Application Cross-Scope Access**.

In this example, the *NeedIt* application *Runtime Access Tracking* is set to *Enforcing*.



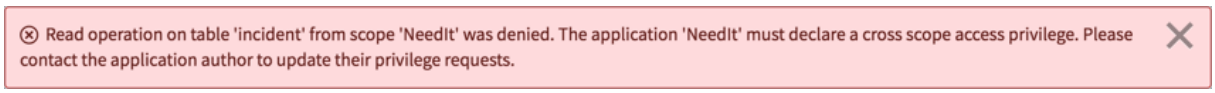
A server-side script in the *NeedIt* application attempts to execute a *GlideRecord* query against the *Incident* table. The *Incident* table is in the *Global* scope.

```
var gr = new GlideRecord('incident');
gr.query();
gs.info("total records in Incident table = " + gr.getRowCount());
```

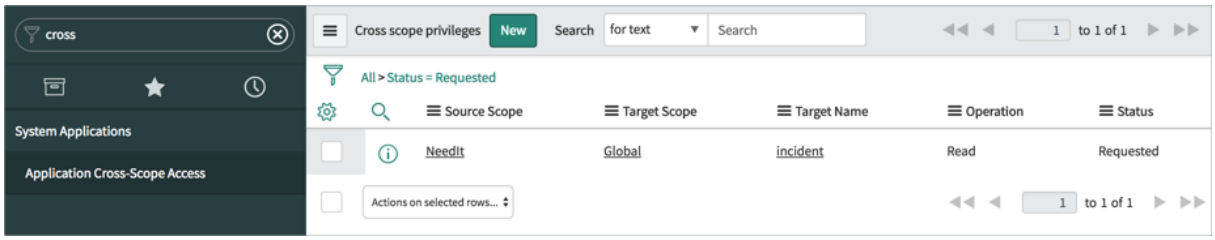
When the script executes, ServiceNow checks to see if the cross-scope access between the *NeedIt* scope and the *Global* scope table is allowed. In this case it is not because the *Enforcing* setting requires an admin to authorize the access. This is a snippet of the error from the System Log:

```
com.glide.script.fencing.access.ScopeAccessNotGrantedException: read access to incident not granted
Caused by error in script at line 2

1: var gr = new GlideRecord('incident');
==> 2: gr.query();
3: gs.info("total records in Incident table = " + gr.getRowCount());

com.glide.script.fencing.ScopedGlideRecord.checkOperationPermitted(ScopedGlideRecord.java:238)
```

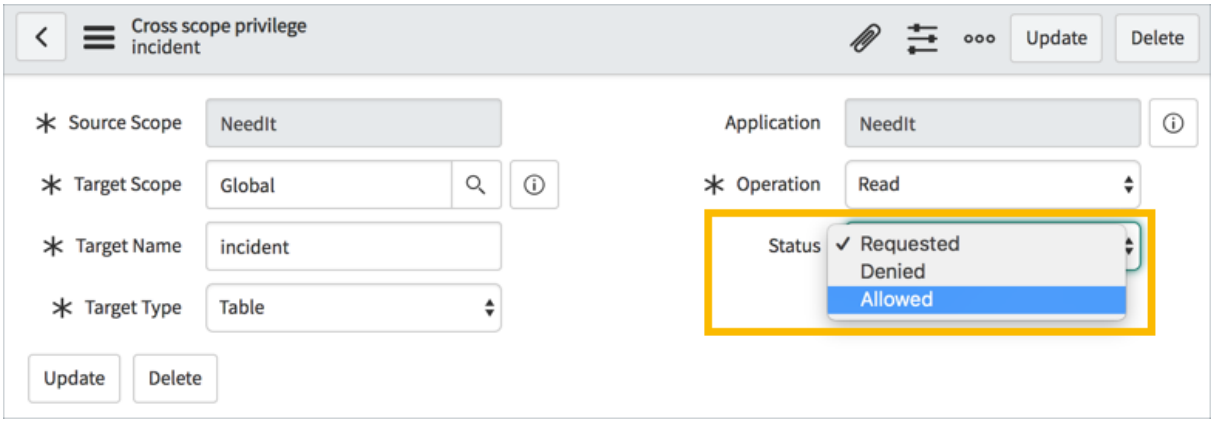
Users with the *admin* role will also see a message on a record form.



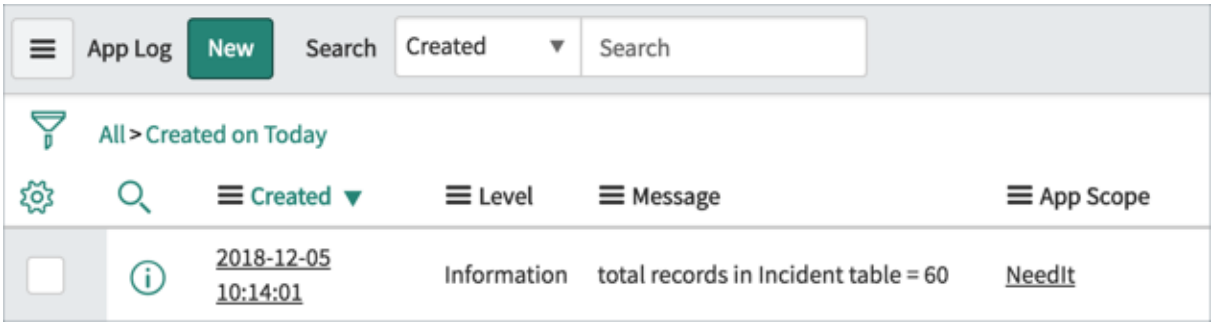
Open the *Application Cross-Scope Access* module and search for all records with a *Status* field value of *Requested*.



To grant access, an admin user must click the **Preview** icon (ⓘ) and then click the **Open Record** button to open the record for editing. Change the *Status* to *Allowed* then click the **Update** button.



The script to count and log the number of incident records now runs without error because the cross-scope access is allowed.



Cross-scope privileges can be granted for:

- ***Table***: Read, write, create, delete records
- ***Script Include***: Execute API
- ***Scriptable (script objects)***: Execute API

The cross-scope privileges must be set before applications are installed on other instances. While applications are under development, be sure to test all script logic to establish the appropriate privileges.