Article
Alessandro Marin · Jan 8, 2018 4m read

DeepSee: Setting up security - Part 5 of 5

In part 1, part 2, and part 3 parts of this series we set up three user types. In part 4 we saw how to secure model elements and DeepSee items. In this last part of the tutorial we conclude with some remarks on DeepSee security and troubleshooting tips. In particular, we see how pivot tables in User Portal can be "hidden".

Remarks

Controlling pivot visibility in User Portal

A common application requirement is to hide all pivot tables from users accessing the User Portal. This can be done, but not using security. Instead, pivot tables (as well as other folder items) have a Public setting that can be checked or unchecked when saving. Marking a folder item as Public or Private will show or hide the folder item in User Portal to users other than the owner, respectively. Please refer to the documentation Visibility of Folders and Folder Items.

Role organization

It is recommended to divide the roles in two categories. Roles that affect **functionalities** (e.g. Read-only access to Architect, Read-Write permission on User Portal, etc.) and **item visibility** (e.g. access to the INVOICES cube or the Users folder in User Portal). In this tutorial we created three functionality roles (DSUser, DSPowerUser, and DSAdmin) and one item visibility role (DSInvoices).

Resources controlling item visibility

Item visibility roles gather resources that secure DeepSee items, cubes and subject areas, etc. These resources can be assigned to DeepSee items such as folders (such as fAdmin in this tutorial), pivot tables, dashboards, KPIs, listings, listing groups, etc. The suggested naming conventions for Resources are:

Cubes	c<cube name>
Subject Areas	s<subject area name>
KPI	k<KPI name>
User custom resource	u<user name>

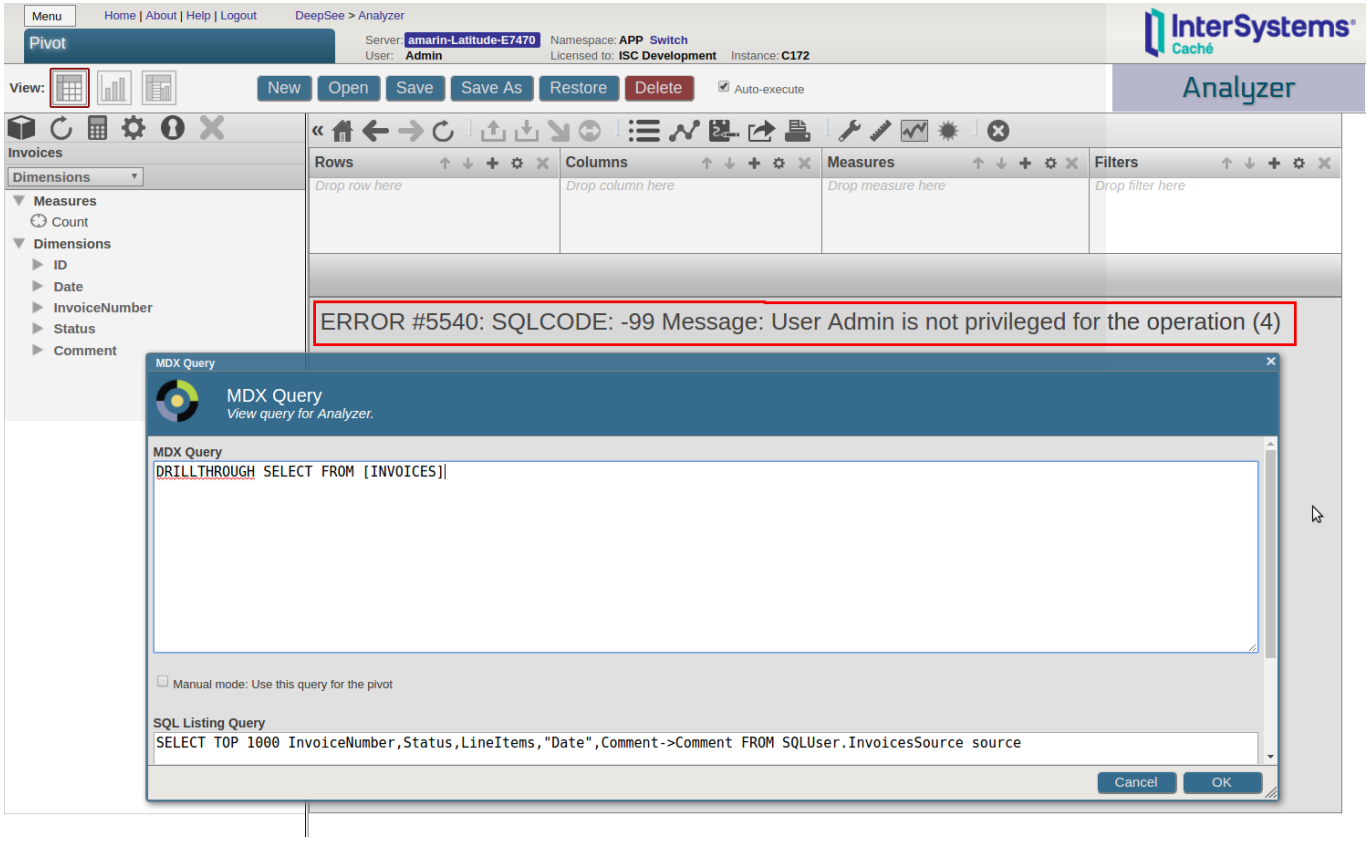
In this tutorial we created two resources: cInvoices secured a cube and the pivot tables based on it, whereas fAdmin controlled the visibility of the Admin folder in User Portal.

Remote databases

Caché and DeepSee can use remote databases. If the remote system has security set up, the users' roles will have to grant permissions on the resources needed on the remote system. This means that you might have to create resources on the local system having the same name as the ones needed on the remote system.

Listings

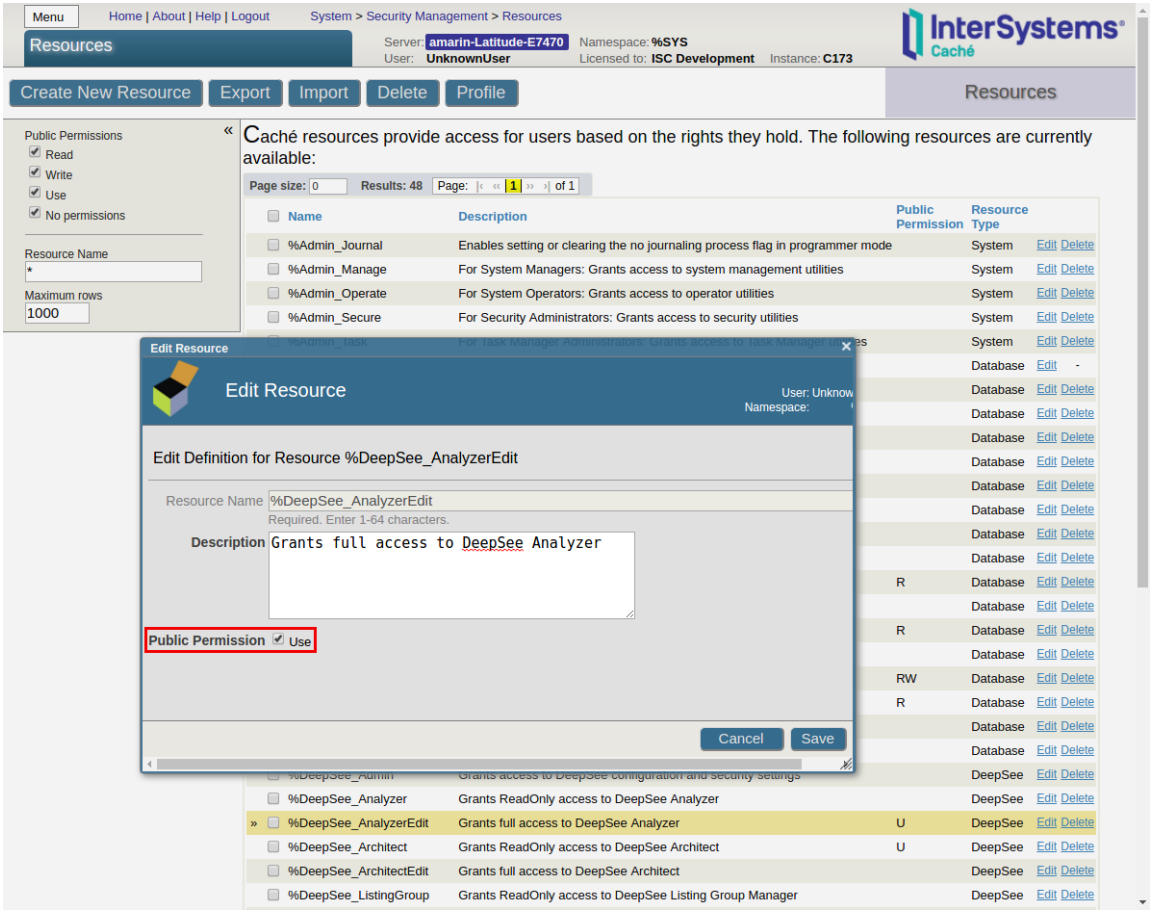
It is recommended to test listings in pivot tables and dashboards. Listings pull data from source tables. When users have no access to the needed resource, it is possible to see an error about missing privileges for the operation as in the following screenshot:



This post shows how to troubleshoot this error.

Tips to troubleshoot issues

- Check the public permissions for the relevant DeepSee resources. Public permissions will be granted to all users. For example, let's suppose that in the Resources page the %DeepSeeArchitect resource shows "U" as public permissions. This means that all users will have Use permissions to Architect and will be able to access it.



- Enable auditing to troubleshoot issues. In the Auditing page > Configure System Events enable %System/%admin /Login, Logout, and LoginFailure to audit log in/log out events, %System/%Security/Protect to audit access to protected resources and troubleshoot protect errors. Then go to Auditing page > Enable Auditing to enable auditing. Go to Auditing page > View Audit Database to see the logged events. Note however that not all errors are covered by audit events. As in the example on listings above, failure to access SQL tables is often not logged in the Audit page.

Conclusions

In this post we set up a simple security model. simpleuser is allowed to use dashboards but not to edit or save them. By using two visibilities roles we made sure that simpleuser can neither access dashboards/widgets based on a certain cube, nor see in User Portal a particular folder containing pivot tables and dashboards that would otherwise be visible to all users.
poweruser is granted full access to pivot tables in Analyzer and dashboards, meaning that poweruser can see and edit all pivot tables and dashboards including a DeepSee folder protected by a custom resource.
Finally, the Admin user has also access to Architect, Folder Manager, Cube Manager, Query Tool, and Settings pages.
The number and types of users needed for an application can vary. This post outlines some guidelines on how to implement a basic security model, offers some tips for the implementation, and shows how to troubleshoot typical issues with security models. While there are different ways to achieve the same functionalities, this post is hopefully a practical starting point to implement security.

#Access control #Beginner #Security #InterSystems IRIS BI (DeepSee)