

REVELATION

S O F T W A R E

Controlling the Development Tools in an OpenInsight 9.x Network User License

Locking down your deployed systems using the Developer based engine supplied with the Network User Engine.

Adapted from an original paper authored by Mike Ruane, Revelation software, Inc.

24th December 2008

Table of Contents

INTRODUCTION	3
A WARNING!!	3
HOW DOES OPENINSIGHT HANDLE THE LOCKING DOWN OF ENTITIES AND SYSTEM COMPONENTS?	4
USER ACCESS LEVEL	5
STOP SHARING THE DEVELOPMENT TOOLS BETWEEN APPLICATIONS	6
USING THE REPOSITORY TO LIMIT USER ACCESS	9
IN CONCLUSION	13

COPYRIGHT NOTICE

© 2008 Revelation Software Limited, revelation software, Inc. All rights reserved.

No part of this publication may be reproduced by any means, be it transmitted, transcribed, photocopied, stored in a retrieval system, or translated into any language in any form, without the written permission of Revelation Software Limited.

TRADEMARK NOTICE

OpenInsight is a registered trademark of Revelation Technologies, Inc. Advanced Revelation is a registered trademark of Revelation Technologies, Inc. OpenInsight for Workgroups is a registered trademark of Revelation Technologies, Inc. Report Builder+ is a trademark of Revelation Technologies, Inc.

Microsoft, MS, MS-DOS, Windows, Vista are registered trademarks of Microsoft Corporation in the USA and other countries. All other product names are trademarks or registered trademarks of their respective owners.

Introduction

Since the release of 7.2.1 back at the beginning of 2006, Revelation have provided their developers with the option to deploy OpenInsight application as locked down runtime licenses or with development capabilities using the Developer Class license. From version 9.0, all multi-user deployed systems will be supplied using a 'Network User License' which is the new name for the old 'Developer Class License'. The 'Single User Runtime' remains unaffected.

Whereas the traditional Runtime license did not allow users (either by license or by configuration) to make use of some of the development tools, such as the Form Designer, Compiler or the Dictionary Builder, the Network User License has no such limitations. These tools and others are available to users of an application that is deployed under 9.x using tools out of the box and without attention to locking down certain components.

That said, some limitations do still apply to the Network User License though, most notably the ability to deploy applications using the Runtime Deployment Kit or any other deployment means.

There will be many instances where a system developer will not want to expose all or some of those tools to their users or make them available through any means. Fortunately, OpenInsight contains the capabilities to enable the developer to restrict access to those tools as required and this document will describe how to limit access to the OpenInsight Development tools.

A Warning!!



Do Not Do This in the SYSPROG Account (Application).

It is most important that the changes that are discussed in this document are NEVER applied to the SYSPROG account in OpenInsight. As with all Pick-based systems, the SYSPROG account or application is a special 'System Programmer Account' which is equivalent to a Windows 'Administrator' user, or a Linux or UNIX 'root' user.

Making any changes to the SYSPROG account, whether correctly applied or otherwise, may cause OpenInsight to stop working correctly. For instance, by making a few wrong settings, you could make changes that result in the entire suite of development tools being disabled for every user in every application. By leaving the SYSPROG account alone, you will always have a master account to fall back to, should you lock down your application too tightly.

Note: Within OpenInsight, we refer to a collection of logically related entities and data files as an 'Application'. Most of the MultiValue world refers to this collection as an 'Account'. For purposes of this document, the two terms shall be used interchangeably.

How does OpenInsight handle the locking down of entities and system components?

Central to OpenInsight is a central repository that consists of tables, indexes and entity relationships. The repository maintains information about all of the entities (items/components) created within an OpenInsight application. This includes the application's forms, the form executables, pop-ups, stored procedures, stored procedure executables, dictionary items if that option is set for the application and more.

The repository maintains relationships between those entities for both a 'uses' relationship (as in the BOOKS form 'uses' the BOOKS pop-up) and a 'used-by' relationship (as in the CUSTOMERS dictionary item CUSTOMER_NAME is 'used-by' the CUSTOMER monthly report).

In addition to maintaining the entity relationships, the repository also regulates access to entities in an application. It is therefore possible to use the repository settings to limit user access to specific entities within an application, including the development tools.

Over the next few pages, we will firstly take a look at a way to limit access based on user access levels and then we will explore solutions using the Repository itself.

User Access Level

Application Users are created within the Database User Management utility, which can be found under the Database Manager's, 'Database->User Management...' menu option. During this process users are assigned a privilege setting, being 'User', 'Administrator' or 'System Administrator', as seen in Figure 1. In the example, I have set myself up with System Administrator access level.

Users that are configured with 'User' level access do not have access to the Application Manager. The Application Manager is the default developer's access point to an OpenInsight Application and it is this interface that provides the developer with access to the entire OpenInsight toolset. It should also be noted that by setting a user to have a 'User' Access Level, they will not have access to the development tools. When they go to open the application, they will be presented with the application entry screen, so it looks just as if they are running the application.

It therefore follows, that if your application does not allow access to the development tool or TCL, the users should be prevented from easy access to the development tools within OpenInsight. It also goes without saying, that a suitable Application Entry point should be defined within the Application Manager.



Figure 1 – The user access levels are circled in red.

Stop Sharing the Development Tools between Applications

User access to the development tools can also be restricted by simply not sharing the development tools between applications. For example, the Table Builder screens (TB_MAIN, TB_CHILD, etc.) are written using the SYSPROG application, but they can be used by all other applications. This is made possible because of a setting in the Repository that indicates that a particular entity can be shared with other applications.

By simply turning the 'Sharable' attribute on or off from the SYSPROG application (**violating the warning made earlier in this document**), a development tool can be excluded from an entire application.

For example, the entity name for the Table Builder is an OpenInsight Form Executable named TB_MAIN. The following steps would be undertaken to stop sharing that entity with any other application. OpenInsight version 9.0 features a brand new IDE and it is that IDE that I shall use for this example. The old IDE is also available and it can be used in exactly the same way.

Firstly, we run up OpenInsight, log into the SYSPROG application and locate the OpenInsight Executables in the Repository outline view, as shown in Figure 2.

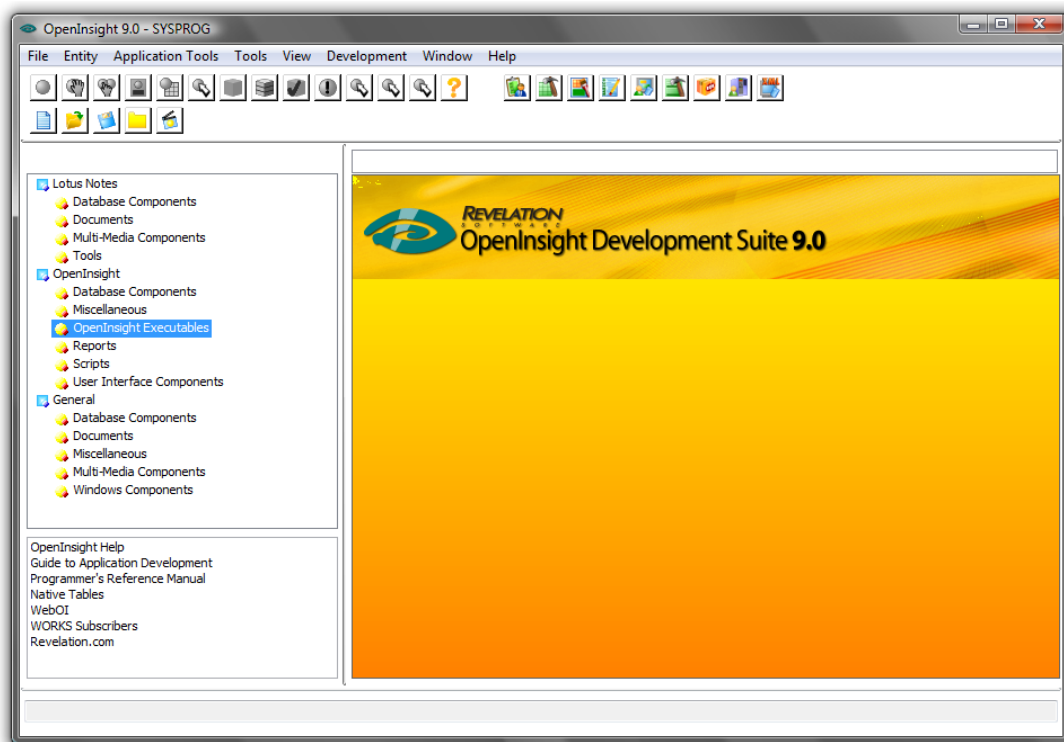


Figure 2 – The new IDE in OpenInsight version 9.0

We need to drill down through the hierarchical list to view the OpenInsight Form Executables node and then scroll down through the list of all the OpenInsight windows until the TB_MAIN entity is visible in the repository outline. We then select TB_MAIN from the list. It should be highlighted as shown in figure 3 overleaf.

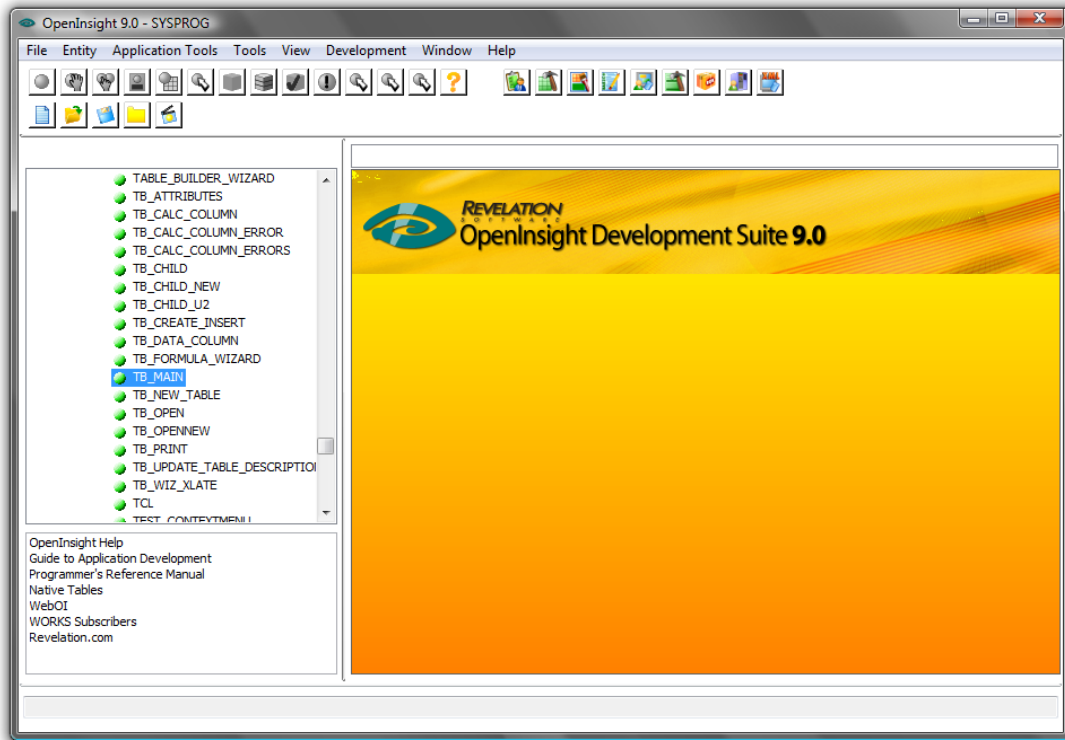


Figure 3 – The TB_MAIN form executable entity is selected in the Repository Outline View.

Once the entity has been selected (highlighted), its properties can be accessed from the Entity menu, see figure 4. It can also be accessed by selecting the TB_MAIN entity and then pressing key combination Alt-F1.

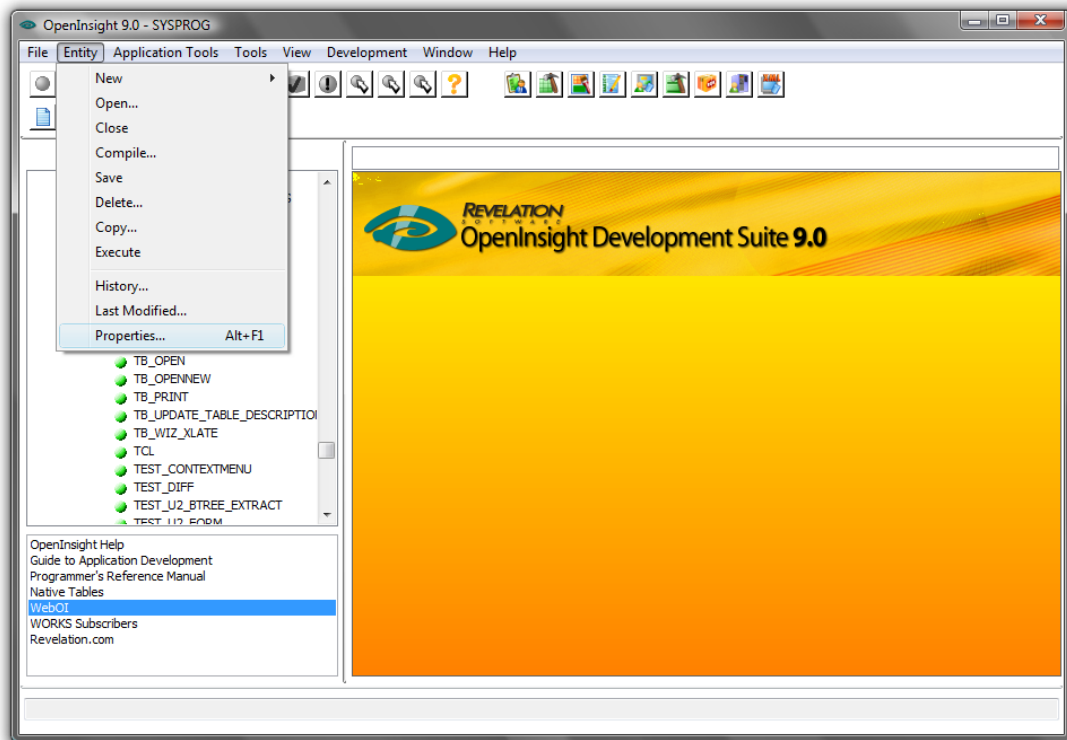


Figure 4 – Selecting the Properties window for the TB_MAIN entity

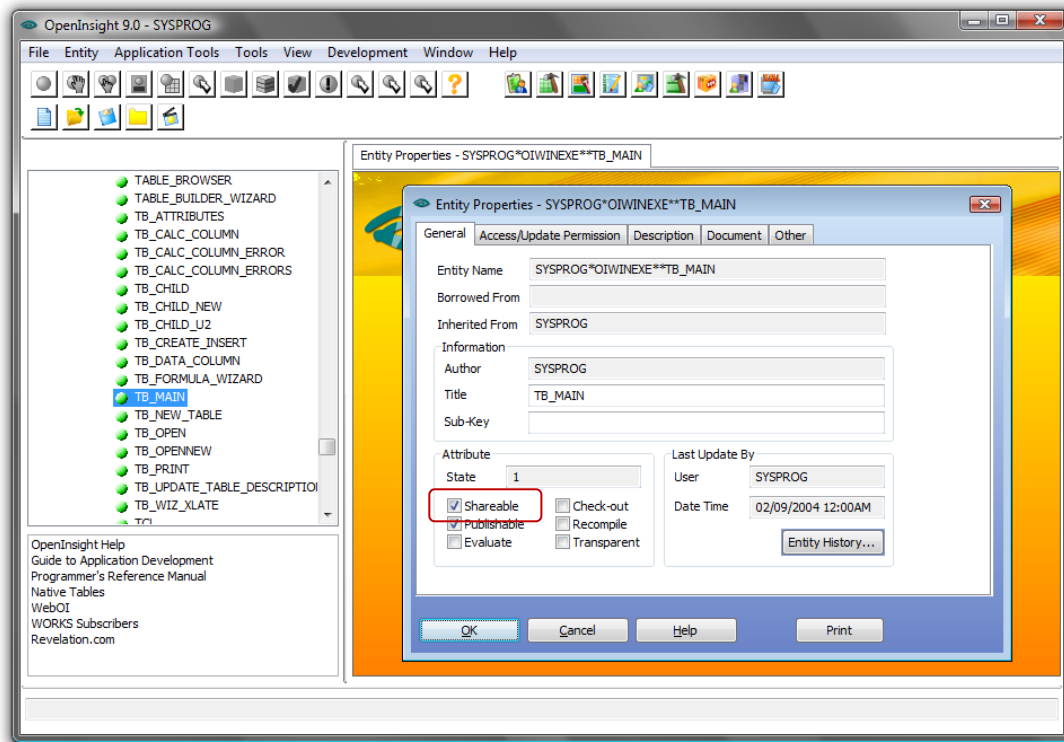


Figure 5 – The Properties screen for the TB_MAIN Form Executable Entity.

The Properties screen contains a lot of information, which is more neatly accessed in version 9.0 through the tab control. Clicking the help button will provide full information regarding the capabilities of this screen. However, we are only interested in the Shareable checkbox (highlighted) for the purposes of this solution.

Checked by default, leaving this checkbox checked allows all new applications to make use of this form. This is possible as all applications inherit sharable entities from the main SYSPROG application.

Clearing the Shareable checkbox renders the entity unavailable to applications inherited from SYSPROG and therefore the entity can **only** then be accessed from and by the SYSPROG application. As shown in figure 6, we are presented with an error when we then try to access the TB_MAIN entity from another application. Of course, full access will be granted to any user trying to access the entity from within the SYSPROG application.

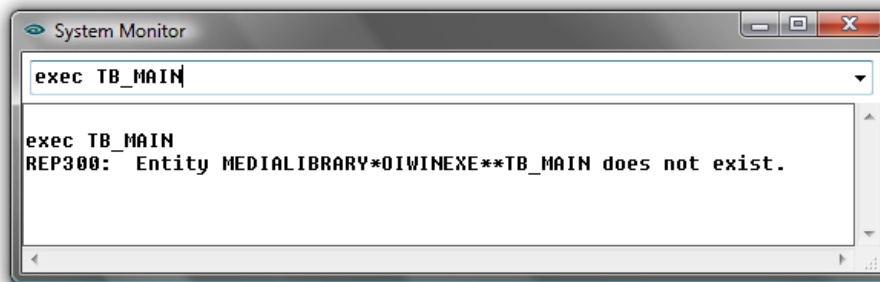


Figure 6 – An error is received when trying to launch a non-sharable entity from another application.

Using the Repository to Limit User Access

The methods of restricting user access to development tools and other entities discussed this far are a little extreme and very inflexible. For instance, in the last example, even the application author would not have access to the TB_MAIN entity, or any tool marked as non-shareable.

In the real world, developers will more often than not wish to give their users access to some development tools and hide others. This is the topic of this section in which we will explore the method of limiting Access and Update privileges on a user level.



IMPORTANT – If you are working through this document following and checking the options in OpenInsight, please make sure that you do not have any instance of OpenInsight logged into the SYSPROG application. It is strongly recommended that you use a test application for this purpose.

Open up the application in which you need to limit user access to the development tools, and or other entities. I have used an application simply called TEST for the writing of this document. Then navigate through the Repository Outline to select the entity that you wish to limit access to (TB_MAIN in this example) as detailed above and open up that entities Properties screen.

Instead of un-checking the Shareable entity (make sure that it is checked), click on the 'Access/Update Permission' tab to view the permissions tab, as shown in figure 7 below.

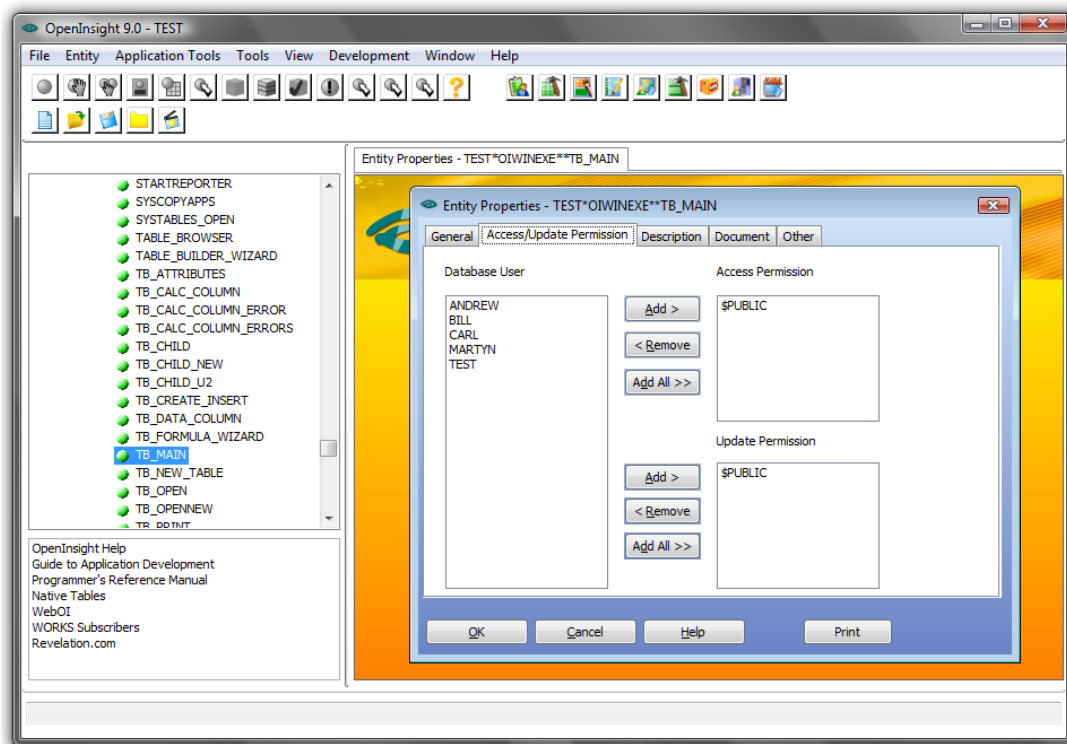


Figure 7 – The entity permission tab/screen

To the left of the buttons Figure 7 shows a list of users that are defined for the current application. To the right of the buttons are two boxes that show the users with Access Permission and Update Permission. In the screen shot, \$PUBLIC is defined for both, meaning that all registered users (Andrew, Bill, Carl, Martyn and Test) have Access and Update permissions for the TB_MAIN entity that we are working with.

In this example we want to limit access to the TB_MAIN entity and only provide access to the entity by MARTYN as a trusted power user on our system. To achieve this, we simply need to select the \$PUBLIC entry under the Access Permission and click the uppermost Remove button and then do the same to remove \$PUBLIC from the Update Permission box using the lower Remove button. The screen should now look similar to that in figure 8

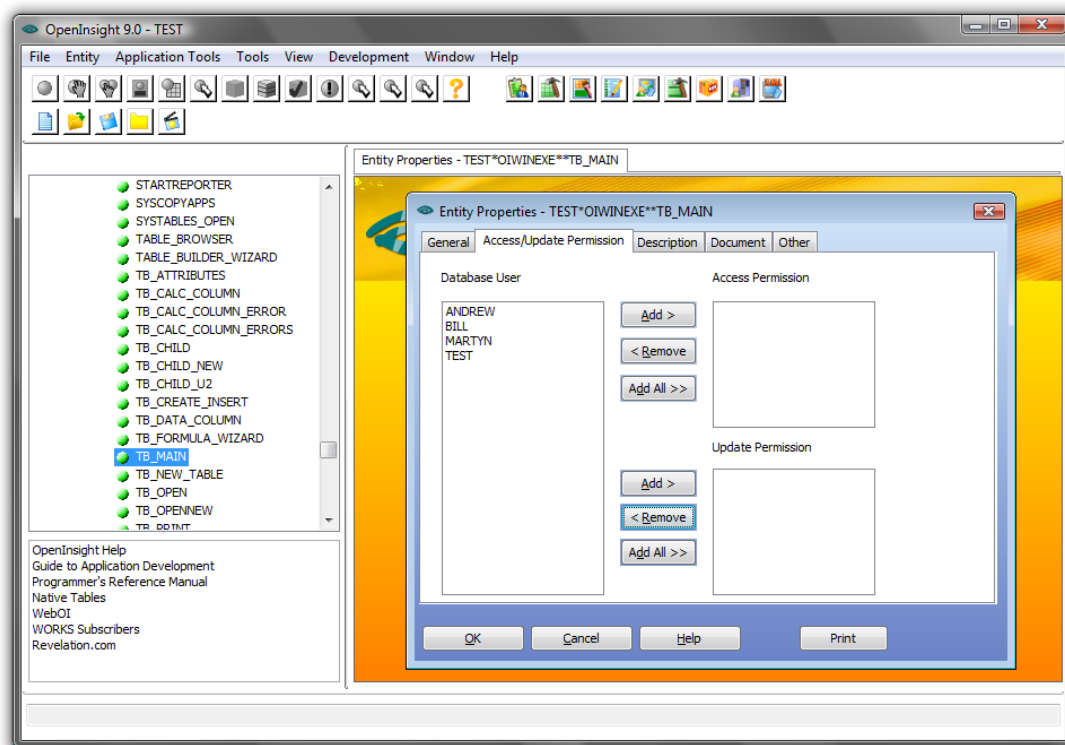


Figure 8 – Showing the \$PUBLIC entries removed from the Access and Update Permission boxes.

Confirm that the Access and Update Permission boxes are both now empty and then select MARTYN (or your chosen user) from the list of available Database Users. Once highlighted, click the Add button to move MARTYN to the Access Permissions box and do the same to move Martyn into the Update Permissions box.

Note that you need to add MARTN to BOTH boxes; otherwise the TB_MAIN entity will still be available and updateable by other users.

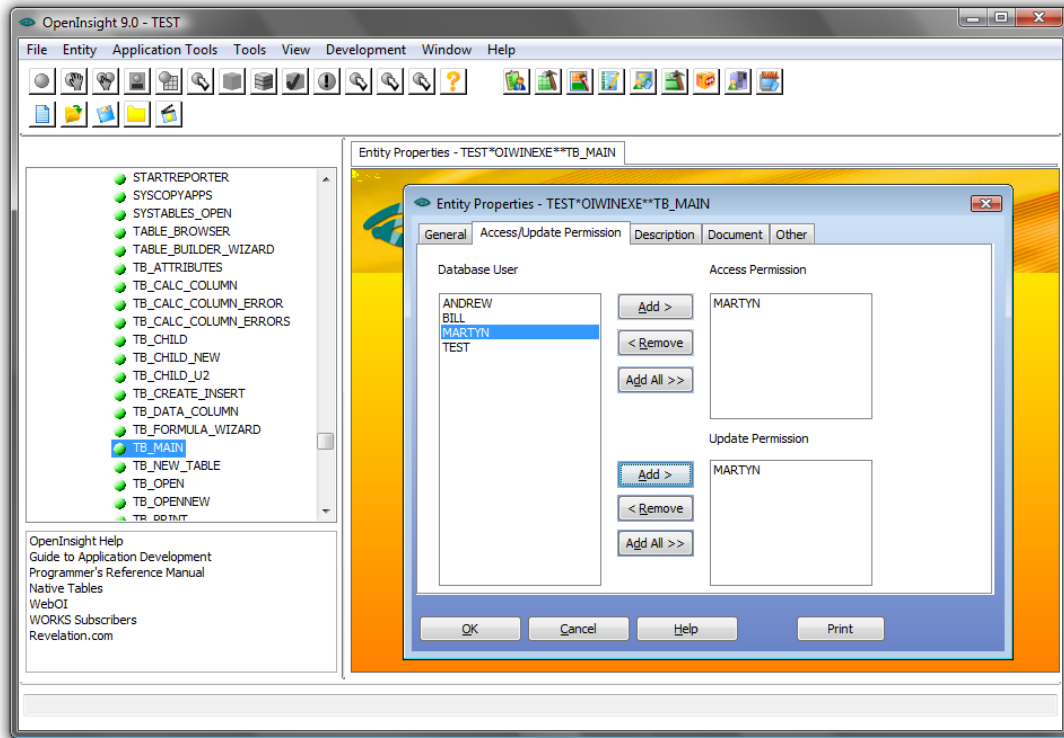


Figure 9 – Only user MARTYN has Update Permissions for the TB_MAIN entity.

As figure 9 shows, MARTYN is now the only user in the application who can use or update the TB_MAIN screen. In fact, if we log into the TEST application as anyone other than MARTYN, we will not see the TB_MAIN entity in the list of application entities displayed under the Repository view in the Application Manager and as shown earlier in figure 3.

If a user tries to be smart and launches the System Monitor or TCL to get around the Application Manager, they will be presented with an error message similar to that shown in figure 10.

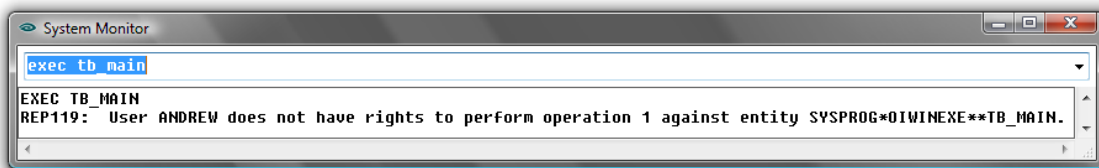


Figure 10 – Andrew is blocked from accessing the TB_MAIN entity that only Martyn has access to.

Please note that by setting the Access and update Permissions at the entity level in this way we have also set the same within SYSPROG. The result is that only MARTYN is able to access and update TB_MAIN in ALL applications inherited from SYSPROG in this copy of OpenInsight.

It should also be noted that OpenInsight behaves differently for users within the system with different User Access levels. As an example, with Martyn set as the only user with Access and Update Permissions on the TB_MAIN entity, the following applies:

User	Access Level ¹	Repository ²	System Monitor Menu Button ³
TEST	System Author	Not viewable	Executable & Updateable
MARTYN ⁴	System Administrator	Executable & Updateable	Executable & Updateable
ANDREW	System Administrator	Not viewable	No access
CARL	Administrator	Not viewable	No access
BILL	User	Not viewable	No access

¹ Access Level set in the Database Manager, User Management window.

² Entity is viewable in the Application Manager's repository view.

³ Entity can be executed from the System Monitor, a defined button and the application menus.

⁴ Remember that Martyn is the named user with permissions to access and update TB_MAIN.

The biggest part of setting access permissions at the entity level is deciding which entities should be exposed and which should not. In the case of our Table Builder (TB_MAIN) example, we have thus far limited access only to the main Table Builder window. This might be enough, but I'd personally go further and limit access to other entities such as TB_NEW_TABLE. Leaving user access to this entity will enable users to open up the create table window and go on to create and save new empty tables within the application – not something that would normally be acceptable or desirable.

It should also be noted that following the methods in this document will not grey out menus or disable buttons to identify that they are not available. This can be handled through training and there are audible notifications of the utilities not being available. However, it might be worthwhile giving due consideration to programmatically disabling the buttons and menu items for limited controls.

It goes without saying that you will be best advised to make these entity limitation settings in a copy of your application which you are preparing for deployment, rather than in your main development system.

In Conclusion

As this document shows, there are several ways to limit user's access to OpenInsight system tools in an OpenInsight 9.x Network User License. In truth, there are more ways than those defined in this document, but these techniques make good use of the tools already provided within OpenInsight.

Your mileage will vary, and you can decide which users get access to which entities or tools without having to go to the expense of writing and maintaining bespoke application code. As mentioned earlier, the hardest part that you will encounter will probably be deciding which tools and entities should be limited and which should be made available to your users.

If this is the hardest decision, then the team at Revelation have done their job and you will be able to concentrate on your business, whilst Revelation make the system design and implementation as easy as it can be.

As always, please ensure that you thoroughly test your 'ready to deploy' application before actually giving it to anyone. In this context, this means checking it for user access levels and the ability for people to only get to those components that you want those users to get to.

We trust that the new licensing model, introduced with version 9.0, will provide developers and users with even more flexibility than before within their applications. In doing so, we hope that those applications will continue to benefit their users for another decade or two.