

Sequence Number Inheritance

Document ID: Q000072

Last Revised On: August 21, 2008

This article applies to the following:

Product Version: IssueNet 5.1 and later

Component(s): Architect

Solutions(s): All

Sequence numbers are the sequential IDs assigned to every IssueNet object. These IDs are an easy to identify supplement to the GUID Object IDs the IssueNet Platform uses to identify objects.

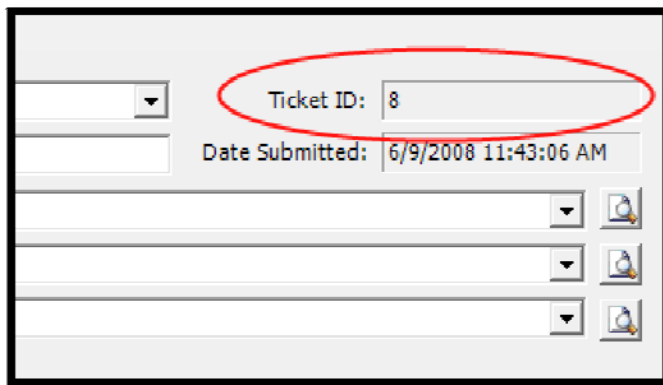


Figure 1: The sequence number provides an easily identifiable ID for any item.

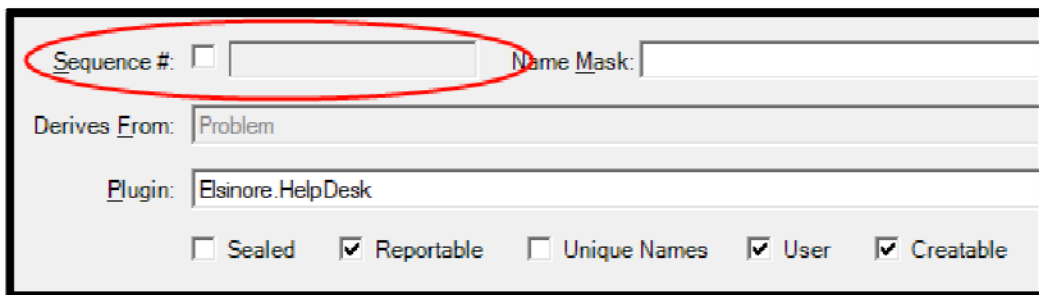
In versions of the IssueNet Platform prior to 5.1 sequence numbers were always assigned in series which were unique per class. So, for example, Customer Support Tickets would always draw sequence numbers from a series distinct from the sequence numbers assigned to other issue types such as Software Defects.

While this approach to numbering had its own advantages and flexibility, a number of customers requested that issues pull their IDs from a common pool. The primary consideration was that their users were already familiar and comfortable with systems in which all IDs were sequential. To answer these requests, Elsinore has implemented new sequence numbering options in version 5.1 which allow one to choose whether an item type has its own unique sequence numbers or pulls them from a series it shares with other classes.

This new feature leverages the fact that item types in IssueNet are implemented as classes which inherit properties from their parent classes. In IssueNet 5.1 you can specify whether a class will assign its own sequence numbers or inherit the sequence numbering of its parent class.

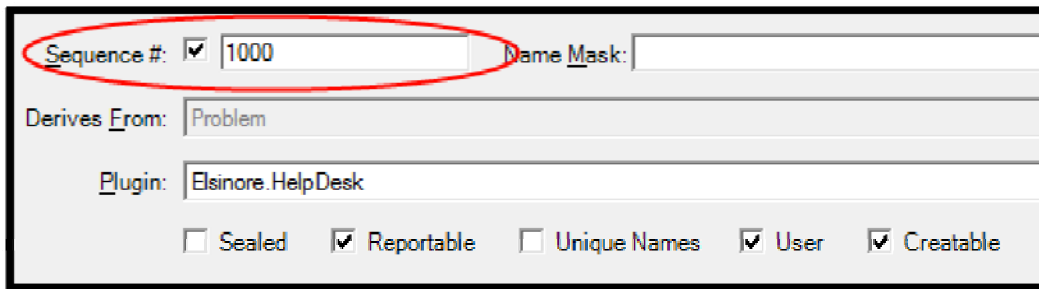
To better understand how the feature works, open the IssueNet Architect, expand the class editor and open a class. You can open any class. Sequence numbers are assigned to members of all classes - not just issues. Once you have a class open, look at the **Sequence #** field on the general tab. You will notice a new check box next to the box for the sequence number value.

If the box is unchecked, the class will assign sequence numbers based on the settings of its parent class, and the box for the sequence number will be disabled. When the box is checked, the class will assign its own sequence numbers and you are required to enter a sequence number starting value.



The screenshot shows the 'General' tab of the class editor. The 'Sequence #' field is highlighted with a red oval. It consists of a checkbox that is unchecked and a text box that is disabled (grayed out). To the right is the 'Name Mask' field. Below this are the 'Derives From' and 'Plugin' fields, and a row of checkboxes for 'Sealed', 'Reportable', 'Unique Names', 'User', and 'Creatable'.

Figure 2: When unchecked a class will assign sequence numbers based on the settings of its parent class.



The screenshot shows the 'General' tab of the class editor. The 'Sequence #' field is highlighted with a red oval. It consists of a checkbox that is checked and an active text box containing the value '1000'. To the right is the 'Name Mask' field. Below this are the 'Derives From' and 'Plugin' fields, and a row of checkboxes for 'Sealed', 'Reportable', 'Unique Names', 'User', and 'Creatable'.

Figure 3: When checked a class will assign its own sequence numbers and a starting value must be supplied.

With the settings clearly illustrated, let's review a practical example of their use. All new databases created with version 5.1 or greater are set so that classes inherit sequence numbers from the base framework class. All issue types, for example, inherit sequence numbers from the base class issue. However, let's suppose you have a database created using an earlier version and want to set all issue classes to assign IDs from the same sequence. To do this, uncheck the Sequence Number box for all issue classes except the base class of issue. If you have existing issues with sequence numbers set the starting sequence number to a value greater than the highest existing issue ID to avoid creating duplicates. Save the changes you have made to the classes and the changes will take effect. If you

create a new issue class and want it to have its own set of IDs, simply check the box and enter the starting sequence number.

In addition to allowing you to have common or unique sequence number series per class, this new feature allows you to revise the inheritance setting at any time as well as change the starting sequence number when you need to. Together these new sequence numbering features give you precise control over the IDs your user rely on when identifying issues and other custom item types.