
Article

[Pravin Barton](#) · May 12, 2022 1m read

Making an existing property Required in a Persistent class

Say I have a persistent class in IRIS with an optional property EmailOptIn:

```
Class Person Extends %Persistent
{
Property Name As %String;
Property EmailOptIn As %Boolean;
}
```

I later realize that I'm doing a lot of null-checking on this property where I shouldn't need to. The solution is to make this a required property:

```
Class Person Extends %Persistent
{
Property Name As %String;
Property EmailOptIn As %Boolean [ Required ];
}
```

When I make this change I'll need to update all the existing data to set a reasonable default where it is null. Otherwise I'm in a bad state where updating the Name of a Person will fail if their EmailOptIn is null. I might expect to fix this in SQL as follows:

```
&sql(update Person set EmailOptIn = 0 where EmailOptIn is null)
```

However, running this query will leave me with no rows affected. I can also use a SELECT statement to verify that the WHERE condition matches no rows. I interpret this as SQL assuming none of the entries are null since the property is required. The workaround is a little unintuitive. Even though no index exists, using the %NOINDEX keyword in the condition forces the update to actually check for null entries.

```
&sql(update Person set EmailOptIn = 0 where %NOINDEX EmailOptIn is null)
```

This edge case tripped me up recently. Hopefully this article can save another developer a little time.

[#SQL](#) [#InterSystems IRIS](#)

Source URL: <https://community.intersystems.com/post/making-existing-property-required-persistent-class>