Contents

The **PropertyInterceptorManager** is the container managing property interceptor actions within DevForce. A developer accesses the *PropertyInterceptorManager* via its *CurrentInstance* property.

Dynamic property interception

The *CurrentInstance* holds all currently registered interceptor actions. You can query for existing actions, and add and remove property interceptor actions.

To directly add actions to the *PropertyInterceptorManager*, use the *AddAction* method. For example, this adds a "before set" auditing method to every entity property:

```
PropertyInterceptorManager.CurrentInstance.AddAction(
 new PropertyInterceptorAction<PropertyInterceptorArgs<Entity,Object>>(
  typeof(Entity), null, PropertyInterceptorMode.BeforeSet,
   (args) => Debug.WriteLine("Entity BeforeSet"), 0.0, "A")
);
```

```
PropertyInterceptorManager.CurrentInstance.AddAction( _
 New PropertyInterceptorAction(Of PropertyInterceptorArgs(Of Entity, Object)) _
   (GetType(Entity), Nothing, PropertyInterceptorMode.BeforeSet, _
   Function(args) Debug.WriteLine("Entity BeforeSet"), 0.0, "A"))
```

Recursive interceptor actions

Recursion within a property interceptor can occur when the interceptor action invokes another property get or set which triggers execution of the same interceptor - this might occur, for example, when an interceptor action accesses the same property on a super- or sub-class. By default, recursive execution is disabled.

For use cases which require recursive execution of an interceptor, set the *AllowRecursiveIntereceptors* flag to true. For example:

```
PropertyInterceptorManager.CurrentInstance.AllowRecursiveInterceptors = true;
```