Contents

The **client side lifecycle** of a save consists of two events on the *EntityManager*; *Saving* and *Saved*. These are summarized in the table below.

Saving and Saved events

Event	EventArgs	Comments	Typical Uses of the Corresponding Event Handler
Saving	EntitySavingEventArgs	Raised when the EntityManager is preparing to save changes. Provides access to the collection of entities submitted for the save. Permits cancellation of the save.	Modify the object submitted for saving, or refuse the request to perform inserts and/or updates.
Saved	EntitySavedEventArgs	Raised when the EntityManager has completed a save (including a save which was cancelled or failed). Provides access to the collection of entities saved and other status information.	Modify the saved object (which might be different from the object submitted for saving by virtue of triggers that were fired on the back end to modify the latter after it was saved).

The *EntityManager* raises a *Saving* event shortly after the application initiates a save. It raises a *Saved* event upon completion of the save, regardless of whether any entities were saved successfully. We can add our own event handlers to these events.

The Saving event provides the handler with a list of entities that the caller proposes to save. It will calculate that list if the method parameters do not prescribe the list. *SaveChanges()* with no arguments, for example, is a blanket request to save every added, changed or deleted entity in cache. The event handler can scrutinize the list, invoke validation methods on selected entities, clean up others (e.g., clear meaningless error conditions), add additional entities to the list, and even exclude entities from the list. Lastly, it can cancel the save.

The *EntityManager* raises the saved event upon completion of the save. The handler receives a list of the entities that were saved successfully.

In DevForce, saves are always transactional, even across disparate back-end data sources. In transactional saves, either every entity in the save list is saved or none of them are.

Virtual EntityManager methods

Corresponding to each of these events is a protected virtual method on the *EntityManager* that can be used to perform the same task directly in a subclassed *EntityManager*.

Member	Summary
OnSaving	The virtual method that a derived *EntityManager* can override to raise the *Saving* event.
OnSaved	The virtual method that a derived *EntityManager* can override to raise the *Saved* event.

EntitySavingEventArgs

Property	PropertyType	Access	Description
Entities	IList<Object>	get - but the list returned can be modified	The list of entities to be saved.
Cancel	bool	get, set	Allows the save to be cancelled.

EntitySavedEventArgs

Property	PropertyType	Access	Description
Entities	IList<Object>	get	Returns the entities saved.

SaveResult	SaveResult	get	Returns the SaveResult - see SaveResult
Cancelled	bool	get	Whether the save was cancelled.
IsCompleted	bool	get	Whether the operation completed.
CompletedSuccessfully	bool	get	Whether the save was successful.
CompletedSynchronously	bool	get	Whether the save was sync.
HasError	bool	get	Whether an exception occurred.
Error	Exception	get	Non-null if an exception was thrown.
IsErrorHandled	bool	get	True if an EntityServerError handler "handled" the error.