Contents

A single *EntityManager* save changes call will save all adds, changes and pending deletions, whether only a single entity should be saved or thousands.

All *EntityManager* **saves are transactional** by default. When the developer saves more than one entity at a time, DevForce processes them together as a single unit of work. Either every save succeeds, or they are all rolled back.

SaveOptions.TransactionSettings

The transactional nature of any save is controlled by the value of the *TransactionSettings* property on the *SaveOptions* class. Recall that every save has a *SaveOptions* instance applicable to it. This is either the *EntityManager's DefaultSaveOptions* or a standalone *SaveOptions* instance specified within the *SaveChanges* call.

The *TransactionSettings* property is of type *TransactionSettings* and contains the following properties, all of which can be set.

Property	PropertyType	Default value	Description
IsolationLevel	IsolationLevel	ReadCommitted	Determines the transactions locking behavior. Please consult your database documentation for more info.
Timeout	Timespan	TimeSpan(0, 1, 0) - 1 minute	The timeout period for the transaction.
UseTransactionScope	bool	true	It is highly recommended that this be set to true (default). It is this setting that causes a transaction that crosses database boundaries to escalate to use the DTC (Distributed Transaction Coordinator).

Most SQL databases define different degrees of consistency enforcement called "isolation levels". Each database vendor has a different default isolation level and a proprietary syntax to change it. Setting the *IsolationLevel* above is dependent upon the support of both the database vendor and the .NET ADO.NET provider being used. In some cases, settings may need be made within the database itself or with proprietary information embedded in the connection string. Consult the database vendor's documentation.

Distributed transactions

By default, DevForce also provides transactional integrity when saving entities to two or more data sources. These data sources can be of different types from different vendors. Their data source managers must support the X/Open XA specification for distributed transactions.

By setting the *UseTransactionScope* property mentioned above to "true" (the default), the developer is instructing DevForce to use the .NET Enterprise Services (AKA COM+) Distributed Transaction Coordinator (DTC) to handle transaction management when more than one database is involved. If you will be saving entities across databases in a single call, be sure to enable DTC in your system. (Control Panel > Administrative Tools > Services)