

Contents

- [FetchStrategies](#)
 - [FetchStrategies when the client is disconnected from the data source](#)

DevForce uses the [FetchStrategy](#) property of the [QueryStrategy](#) class to **control where the query executes**: in the cache, in the datasource, or in some combination of the two.

FetchStrategies

Five [FetchStrategies](#) are available in DevForce:

Strategy	Action
CacheOnly	Apply this query against the cache only, returning references only to entities already there. Do not consult the data source. (Note that this query leaves the cache unchanged.)
DataSourceOnly	Retrieve matching entries from the datasource into the entity cache . Return references only to those entities retrieved from the data source. A result set returned from a query using this FetchStrategy would <i>not</i> include locally added entities that had not yet been persisted to the data source.
DataSourceThenCache	First retrieve matching entries from the datasource into the entity cache. Discard all references to entities retrieved in this step. Resubmit the same query against the updated cache. Return references only to entities matched by this second, CacheOnly query.
DataSourceAndCache	First retrieve matching entries from the datasource into the entity cache. Retain references to entities retrieved in this step. Resubmit the same query as <i>CacheOnly</i> . Combine (union) the references obtained in this second, <i>CacheOnly</i> query with those obtained in the data source retrieval step.
Optimized	Check the <i>query cache</i> to see if the current query has previously been submitted (and, if necessary, inverted) successfully. If so, satisfy the query from the <i>entity cache</i> , and skip the trip to the datasource. If the query cache contains no query matching or encompassing the current query, then determine if all entities needed to satisfy the query correctly from the cache can be retrieved into the cache. If so, apply the <i>DataSourceThenCache FetchStrategy</i> . Otherwise, apply the <i>DataSourceOnly FetchStrategy</i> . See the discussion on query inversion for more detail.

FetchStrategies when the client is disconnected from the data source

If the client is disconnected from the data source, the *DataSourceOnly*, *DataSourceThenCache*, and *DataSourceAndCache* strategies will throw an *InvalidOperationException*. The *Optimized* strategy will behave as a *CacheOnly* query. It will not throw an exception, even if no matching query exists in the query cache.