

LiveCode Datastore API – Developer Reference Manual

Complete reference for all public datastore functions and commands.

Table of Contents

1. Overview
2. Collections API
3. Schema Management
4. Records API
5. Current Record Management
6. Dataview API
7. Container API
8. Databinding Events
9. Actions API
10. Asset API

1. Overview

This manual documents all public datastore API calls.

All asynchronous operations dispatch:

dispatch pCallbackHandler to pCallbackStack with pError, pData, pContext

All record arrays are sequenced arrays (1..N).

All schema structures follow IDE■formatted dictionary form.

2.1 datastoreGetCollections

Fetches or returns cached collection metadata.

Parameters:

- pUseCache – if true, returns cached array.
- pCallbackStack – stack to receive callback.
- pCallbackHandler – invoked with pError, pData, pContext.

Returns:

- If pUseCache: array of collections.
- Else async callback.

```
get datastoreGetCollections(true)
datastoreGetCollections false, the long id of me, "onCollectionsLoaded"
```

2.2 datastoreGetCollectionNames

Returns cached return■delimited list of collection names.

```
put datastoreGetCollectionNames() into tNames
```

2.3 datastoreGetCollectionsWithFieldType

Returns names of all collections containing at least one field of type pType.

```
put datastoreGetCollectionsWithFieldType("enum") into tCols
```

2.4 datastoreGetNumberOfRecordsInCollection

Returns cached record count.

```
put datastoreGetNumberOfRecordsInCollection("Tasks") into tCount
```

2.5 datastoreGetCollectionSchema

Fetches schema for a collection.

Parameters:

- pUseCache – if true, returns schema synchronously.
- pCollectionName
- pCallbackStack, pCallbackHandler, pContext

Returns formatted schema (field → {type, options}).

```
put datastoreGetCollectionSchema(true, "Tasks") into tSchema
datastoreGetCollectionSchema false, "Tasks", the long id of me, "onSchema", empty
```

3.1 datastoreAddCollection

Creates a new collection with schema.

```
put "text" into tS["title"]["type"]
datastoreAddCollection "Tasks", tS, the long id of me, "onAdded", empty
```

3.2 datastoreDeleteCollection

Deletes a collection if not used by dataviews.

```
datastoreDeleteCollection "Tasks", the long id of me, "onDelete"
```

3.3 datastoreAddFieldToCollection

Adds a new field to an existing collection.

```
datastoreAddFieldToCollection "Tasks", "priority", "number", empty, the long id of me,
"onFieldAdded"
```

3.4 datastoreRemoveFieldFromCollection

Removes a field from schema.

```
datastoreRemoveFieldFromCollection "Tasks", "priority", the long id of me,
"onFieldRemoved"
```

3.5 datastoreUpdateCollectionFields

Batch update of schema fields.

```
put "enum" into tU["status"]["type"]
put "Open,Closed" into tU["status"]["options"]
datastoreUpdateCollectionFields "Tasks", tU, the long id of me, "onSchemaUpdated"
```

4.1 datastoreCreateNewRecord

Creates a new empty record based on schema.

```
datastoreCreateNewRecord "Tasks", true, the long id of me, "onNewRecord"
```

4.2 datastoreCreateNewRecordInCollectionWithData

Creates a record with initial values.

```
put "Do something" into tD["title"]
datastoreCreateNewRecordInCollectionWithData "Tasks", tD, true, the long id of me,
"onRecordCreated"
```

4.3 datastoreBulkCreateNewRecordInCollectionWithData

Creates many records in one call.

```
put "A" into tArr[1]["title"]
put "B" into tArr[2]["title"]
datastoreBulkCreateNewRecordInCollectionWithData "Tasks", tArr, the long id of me,
"onBulkCreated"
```

4.4 datastoreGetDataOfCollection

Fetches records for a collection. Supports synchronous and async modes.

```
put datastoreGetDataOfCollection(true,false,"Tasks") into tRecs
datastoreGetDataOfCollection false,true,"Tasks", the long id of me, "onGotRecs", empty
```

4.5 datastoreGetDataOfRecordOfCollection

Returns record data for given record number.

```
put datastoreGetDataOfRecordOfCollection("Tasks", 1) into tRec
```

4.6 datastoreUpdateRecordOfCollection

Updates a record by number.

```
put datastoreGetDataOfRecordOfCollection("Tasks",1) into tR
put "Closed" into tR["status"]
datastoreUpdateRecordOfCollection "Tasks",1,tR,the long id of me,"onUpdated",empty
```

4.7 datastoreDeleteRecordOfCollection

Deletes record by number.

```
datastoreDeleteRecordOfCollection "Tasks",1,the long id of me,"onDeleted"
```

4.8 datastoreComputeFieldOptions

Returns unique values across all records for a field.

```
put datastoreComputeFieldOptions("Tasks","status") into tOptions
```

5.1 datastoreSetCurrentRecordOfCollection

Sets current record index.

```
datastoreSetCurrentRecordOfCollection "Tasks", 3
```

5.2 datastoreCurrentRecordOfContainer

Returns current record for a container.

```
put datastoreCurrentRecordOfContainer("collection","Tasks") into tCur
```

6.1 datastoreGetDataviews

Fetches list of dataviews.

```
datastoreGetDataviews false, the long id of me, "onDVLoaded", empty
```

6.2 datastoreAddDataview

Creates a dataview.

```
put "Tasks" into tI["collection-name"]  
put ("title","status") into tI["field-names"]  
datastoreAddDataview "TasksView", tI, the long id of me, "onDVAdded"
```

6.3 datastoreGetDataOfDataview

Fetches dataview data after applying filters and sort.

```
datastoreGetDataOfDataview false, "TasksView", the long id, "onDVD", empty
```

7. datastoreGetDataOfContainer

Unified API to fetch data from either collection or dataview.

```
datastoreGetDataOfContainer false, "collection", "Tasks", the long id of me,  
"onLoaded", empty
```

8. databinding events

These fire automatically on data or schema changes.

- databindingDataChanged
- databindingRecordChanged
- databindingSchemaChanged

9.1 revDatastoreSetDataviewFilter

Sets filter value for a dataview.

```
revDatastoreSetDataviewFilter "TasksView", "statusFilter", "Open"
```

9.2 revDatastoreSetDataviewSort

Sets sort for a dataview.

```
revDatastoreSetDataviewSort "TasksView", "sortByTitle", "ascending"
```

9.3 revDatastoreRefreshDataview

Forces a dataview to re-evaluate records.

```
revDatastoreRefreshDataview "TasksView"
```

10.1 revDatastoreAddAssetToCollection

Opens file dialog and assigns asset to field.

```
revDatastoreAddAssetToCollection "Tasks", "attachment", empty
```

10.2 revDatastoreDownloadAssetFromCollection

Downloads asset from current record.

```
revDatastoreDownloadAssetFromCollection "Tasks", "attachment"
```

10.3 revDatastoreDeleteAssetFromCollection

Clears asset field in current record.

```
revDatastoreDeleteAssetFromCollection "Tasks", "attachment"
```