

Organizing Data using RecordSets

Handling the management of structured data can take on many different forms in the Curl language. Our goal is to obtain content into a form we can easily represent internally for manipulation and display. A primary class for data management is RecordSet, which organizes data in terms of records and fields, and manages events having to do with changes in the data. RecordSets are especially useful for dealing with data derived from one or more databases on a server, or data that is intended to be written to a database. It enables the end user to perform many database-like operations on the data, such as filtering and sorting, directly on the client machine without needing to invoke database functions on the server.

The standard Curl API provides a foundation for representing relationally structured data. RecordSet and related classes provide support for representation, manipulation, display and coordination of such data, thoroughly integrated with Curl graphics. Thus, a RecordSet is the obvious choice for model representation for 'data centric' applications which consume structured datasets from some external (server) provider. Furthermore, because of the breadth of standard functionality, RecordSets should also be considered as the basis for internal structures.

RecordSet Data

A RecordSet holds a collection of Records, each of which contains values for a predefined set of keys (columns). The column structure is specified by field descriptors (using RecordField and RecordFields), which name and characterize the allowable values.

In Curl, there are two ways to easily create a RecordSet:

1. Build the RecordSet programmatically directly in the source code
2. Import data that is structured using comma separated values

Both methods allow you to quickly create data structured in record format. The methods above should be used for small amounts of data, or data that does not change rapidly. You can easily adapt these concepts to your own data, allowing you quickly create a RecordSet structure that can be used as input into grid, charts, and forms.

The Curl API also includes features for creating a RecordSet that is directly connected to a database or remote data file. This is discussed in the <http://developers.curl.com/docs/DOC-1109> Curl Cue.

Programmatically Create a RecordSet

As with most Curl objects, a RecordSet can be specified completely declaratively, using the rest arguments of its constructor.

- The RecordFields object is used to define the collection of RecordField objects, each of which describes the data format and type of one column in the RecordSet.
- The RecordData object is used to organize value tuples which produce successive member Records.

In the example above, we are specifying the item name and type using a RecordField. For example, the first RecordField is defined as {RecordField "Item", domain = String} . This specifies the name of the field to be "Item" and the type of the data in the field to be a String. Each record is specified by a RecordData expression where the value of each field is specified as a keyword argument using the name of the field as the keyword. A RecordGrid is used to automatically provide a tabular display of the RecordSet data.

	Standard Domains are defined for the following basic Curl types: char, String, int, int64, float, double, bool, DateTime, Time, or any
--	--

	For more information regarding RecordSets, please refer to the following section in the Curl Documentation:Curl Developer's Guide > Data Management and Display > Data Records and Grids
--	---

Creating a CsvRecordSet

CSV stands for Comma Separated Values, sometimes also called Comma Delimited. A CSV file is a specially formatted plain text file which stores spreadsheet or basic database-style information in a very simple format, with one record on each line, and each field within that record separated by a comma.

CSV files are often used as a simple way to transfer a large volume of spreadsheet or database information between programs, without worrying about special file types. For example, transferring a home-made address book from Microsoft® Excel into a database program such as Filemaker® Pro could be done by exporting the file as a CSV from Excel, then importing that CSV into Filemaker.

Curl also has APIs for working directly with data files in CSV format. CsvRecordSet is a subclass of RecordSet and it allows you to automatically create one or more Records in a RecordSet.

Let's begin our example with a CSV file. This example uses a data file created using Microsoft ® Excel and saved in CSV format. Each line in the file is a record. Each field in the file is separated from its neighbor by a comma.

We'll use a `CsvRecordSet` to import the contents of our file into `Curl`. `CsvRecordSet` does the following:

- Creates a `Record` from each line of the CSV data
- Parses each field of the data
- Converts each field into a `Curl` datatype

The above example shows how you can quickly bring your data into `Curl`. `CsvRecordSet` automatically attempts to map the CSV content into an equivalent datatype using `DomainDetector`. This automatically detects the type of data in each column of a CSV file.

Alternatively, we can provide a `RecordFields` object that defines the expected structure and domain of the data. In this case, `CsvRecordSet` attempts to parse each field of the data file in terms of the `Domain` specified in the corresponding `RecordField`.

In this case, `CsvRecordSet` parses each field of the data file in terms of the `Domain` specified in the corresponding `RecordField`. In our example, we are specifying the following data types:

- `Item` --> `String`
- `Description` --> `String`
- `Quantity` --> `int`
- `Price` --> `double`

Also note that the column headings for each `Record` are specified by the `RecordFields` object. A caption can also be set for each column, allowing you to keep the internal name and specify a different name for display.

	If the string data in the file cannot be converted to an object of the appropriate type, the code throws an error.
--	--

In summary, a RecordSet enables you to write applications that manage data organized in records and fields.

- Manage events corresponding to data changes
- Perform many database-like operations on the local data, such as filtering and sorting

Two easy ways to create Record data in Curl are to programmatically create a RecordSet or use CsvRecordSet to import CSV data.

	For more information regarding CsvRecordSets, please refer to the following section in the Curl Documentation:Curl Developer's Guide > Data Management and Display > Managing Data from External Sources > Using CSV Data Files
--	---