### Build list from file

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Command group** | **Flag affected** | **Reversible** | **Execute on client** | **Platform(s)** |
| [Lists](../lists.html) | YES | NO | NO | All |

#### Syntax

**Build list from file** on field-name ([Exact match][,Use search][,Use sort])

#### Options

|  |  |
| --- | --- |
|  |  |
| Exact match | If specified, the index value of the field in suitable records must equal the current value |
| Use search | If specified, the command uses the current search to select data |
| Use sort | If specified, the command uses the current sort field(s) to order the data |

#### Description

This command builds a list of data from the main file using a specified index field. The records are selected and corresponding field values added to the list in the order of the specified index field. You must set the main file before using the command.

If the Exact match option is specified, only records matching the current value of the specified field are added to the list. Similarly, if the Use search check box is selected, only records matching the current search class are added. In both cases, an error occurs if neither a field nor a search class is specified.

When large files are involved, that is, those that may require more than the maximum number of available lines (the value of LIST.*$linemax*), you can use the flag false condition to detect when an incomplete list is built.

Building a list using this command does not affect the current record buffer and does not clear ‘Prepare for update’ mode.

The Use sort option lets you use the database records in sorted order without first having to load them into a list. You use [Set sort field](set_sort_field.html) to specify a sort field after which **Build list from file** (Use sort) creates a sorted table of records in memory before loading them into the list. The main advantage of this method is that the sort fields do not have to be read into the list at all. The Sort field order overrides the index field order but if the sort field is non-indexed, the index is used as the order in which to gather up records before sorting. Multi-level sorts are possible by using repeated [Set sort field](set_sort_field.html) commands to accumulate the required sorting order. Since sort levels are cumulative you should first clear any existing ones with [Clear sort fields](clear_sort_fields.html).

#### Example

###### # This example compiles a list of all records sorted in order of descending fCustomers.Surname # and within each value, in increasing fCustomers.FirstName order

###### Set current list iMyList Set main file {fCustomers} Define list {fCustomers.Surname,fCustomers.FirstName} Clear sort fields Set sort field fCustomers.Surname (Descending) Set sort field fCustomers.FirstName

###### # Note fCustomers.CustomerID is not in the list

###### **Build list from file** on fCustomers.CustomerID (Use sort)