



344-Multi Dimensional Arrays or Data Structures

HARN

11-Jun-21

In Axcess programming, you can create a 2-dimensional array to store and access data. For example, you could create an array to store 20 records of 30 names, then another 2- dimensional array to store a matching list of speed-dial numbers. NetLinx gives you the power to expand that list up to 5 dimensions, eg; arrayed by name, department, city and country.

Adding a dimension does not add data — it multiplies memory. The first 2 dimensions create a "sheet" of data, a list of 20 records of 30 names, and reserves 600 bytes in memory. Adding the department dimension with 15 departments creates 15 folders, each set to store up to 20 sheets - 9,000 bytes of memory. Adding another dimension for say, 20 cities creates 20 "file drawers" that hold an equal number of 9,000 byte file folders. Now the array multiplies up to 180,000 bytes. Adding the final array of 10 countries creates 10 "file cabinets" - reserving up to 1.8 MB of memory! And that's just the names — you'll need another 1 MB for phone numbers. Arrays reserve an equal amount of memory for each dimension regardless of actual data requirements so much of the memory is wasted on reserving space for unused data. The result is a sparse array with much wasted space and a significant increase in NetLinx code complexity to manage the data.

In the Define_Type section of NetLinx Studio, you can create a "Structure" called, say "PhoneList". Using Structure, you're building a database with Name, Number, Department, City and Country fields. After defining the reserved characters for each field, each record can hold up to 102 bytes. So if the company has 500 employees, the entire speed-dial database requires only 50,100 bytes, regardless of the number of countries, cities, and departments.

In general, while you may be more familiar with arrays from Axcess, you should use Structures if you go beyond 2dimensional arrays. Arrays work well if each level stores an equal amount of data — which usually isn't the case beyond 2 dimensions. In fact, you might consider using Structures instead of arrays any time your data has multiple fields. This makes your code easier to maintain and more reusable.

About HARMAN Professional Solutions

AKG AMX

HARMAN Professional Solutions is the world's largest professional audio, video, lighting, and control products and systems company. Our brands comprise AKG Acoustics®, AMX®, BSS Audio®, Crown International®, dbx Professional®, DigiTech®, JBL Professional®, Lexicon Pro®, Martin®, Soundcraft® and Studer®. These best-in-class products are designed, manufactured and delivered to a variety of customers in markets including tour, cinema and retail as well as corporate, government, education, large venue and hospitality. For scalable, high-impact communication and entertainment systems, HARMAN Professional Solutions is your single point of contact. www.harmanpro.com

::: DigiTech

XIRI.

exicon

Martin

Soundcraff

STUDER

dbr