

# Creating an Address Book Database with Thingworx.

One of the most basic things that ThingWorx can do is store data. Not just from your devices but any type of data you might want to display in a mashup. This example assumes you are familiar enough with the composer to create basic models including DataShapes and DataTables.

As a review, a DataShape in this capacity is like a traditional database schema describing what each row in your table should look like. A DataTable thing is a table consisting of rows that are described by this DataShape.

In this example we will create and search simple address book database.

Follow These Steps.

1. Create a DataShape. For this example, lets call it an AddressBookShape. This will define what each row of your database will look like. Designate the CustomerId as the primary key.

The screenshot shows the ThingWorx DataShape editor for 'AddressBookShape'. The interface includes a top navigation bar with 'AddressBookShape', 'DataShape', 'Save', 'Cancel Edit', and 'To Do' buttons. A left sidebar contains 'ENTITY INFORMATION' with sub-sections: 'General Information', 'Field Definitions' (selected), 'PERMISSIONS' (Visibility, Design Time, Run Time), 'CHANGE HISTORY' (Change History), and 'DEPENDENCIES'. The main area is titled 'Fields' and contains a table with the following data:

	Edit		Name
<input type="checkbox"/>			-T- Lastname
<input type="checkbox"/>			-T- Firstname
<input type="checkbox"/>			-T- Company
<input type="checkbox"/>			-T- Telephone
<input type="checkbox"/>			-T- CustomerId

2. Create a DataTable, call it AddressBook. Add the AddressBookShape as this DataTable's Data Shape.

AddressBook **Thing** [?](#) [Edit](#)

ENTITY INFORMATION

- General Information**
- Properties
- Services
- Events
- Subscriptions
- Configuration
- Home Mashup

PERMISSIONS

- Visibility
- Design Time
- Run Time

CHANGE HISTORY

- Change History

DEPENDENCIES

- Entity Depends On
- Uses This Entity

### General Information

Name	AddressBook
Description	
Tags	
Thing Template	<a href="#">DataTable</a>
Implemented Shapes	

Data Shape	<a href="#">AddressBookShape</a>
Persistence Provider	<a href="#">NeoPersistenceProvider</a>
Documentation	

3. Add your data. Use the AddDataTableEntries service to enter information into the table.

AddressBook **Thing** [?](#) [Edit](#)

ENTITY INFORMATION

- General Information
- Properties
- Services**
- Events
- Subscriptions
- Configuration
- Home Mashup

PERMISSIONS

- Visibility
- Design Time
- Run Time

CHANGE HISTORY

- Change History

### Services

My Services

Action	Service Name	Test	Service T
No Services			

**Click here** →


Then here →

### DataTable (ThingTemplate) - Services

Action	Service Name	Test	Service T
<a href="#">?</a>	AddDataTableEntries	<a href="#">Test</a>	<a href="#">Local</a>
<a href="#">?</a>	AddDataTableEntry	<a href="#">Test</a>	<a href="#">Local</a>
<a href="#">?</a>	AddOrUpdateDataTableEntries	<a href="#">Test</a>	<a href="#">Local</a>
<a href="#">?</a>	AddOrUpdateDataTableEntry	<a href="#">Test</a>	<a href="#">Local</a>
<a href="#">?</a>	AssignDataTableEntries	<a href="#">Test</a>	<a href="#">Local</a>


- A form will then come up, Select the edit button.

### AddDataTableEntries - Test Service

 Please be careful. Only execute services and queries where you understand


**Inputs:**

sourceType

values  
 **Click here** 

location

source

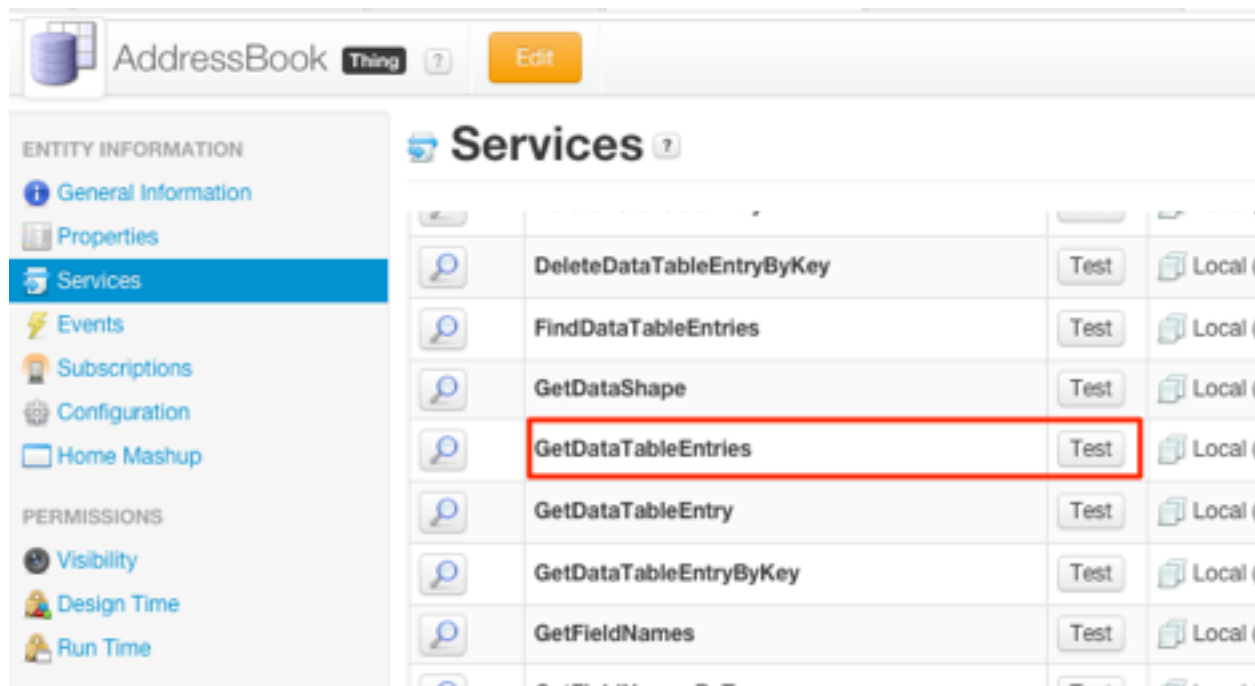
tags  
 

- Add as many new rows as you want. Hit save when done. Then hit the execute service button to finish adding your values.

#### Editing Infotable

	Lastname	Firstname	Company	Telephone	CustomerId
<input type="checkbox"/>	<input type="text" value="Sivers"/>	<input type="text" value="Sivers"/>	<input type="text" value="Neuron"/>	<input type="text" value="221-3224"/>	<input type="text" value="3321"/>
<input type="checkbox"/>	<input type="text" value="McNaughton"/>	<input type="text" value="Joe"/>	<input type="text" value="PSE"/>	<input type="text" value="223-2332"/>	<input type="text" value="1122"/>

6. To see all your values, use the GetDataTableEntries service. Hit the test and then execute buttons. You will now see all the records you have created.



7. To access your records you can either write a custom service or use existing services to query your data table. Try the GetDataTableEntryByKey service and type in a customerID to get back just that specific row.

Remember that any of the services on your DataTable can be called from any mashup, via REST or using one of the ThingWorx SDKs. Try experimenting with the other data access services provided by your address book DataTable to find, modify or delete rows in your DataTable.

Your Data can be exported as a JSON file by using the Export menu and selecting “Export to file”, “Single Data” and the name of your DataTable.

To import large amounts of csv data, use the Thingworx parser extension <http://marketplace.thingworx.com/downloads/csv-parser/> . It allows you to reference CSV files as data tables. You can then use these tables or use a javascript to copy the data from one table to another.