

Core idea:  
Intuitively separate type definitions from type  
declarations

## Definition of MyType

```
// Definition

typedef struct {
    int count;
    char *array;
} MyType;

// other definition syntax
struct MyType {
    int count;
    char *array;
};
```

## Declaration of MyType

```
// Declaration

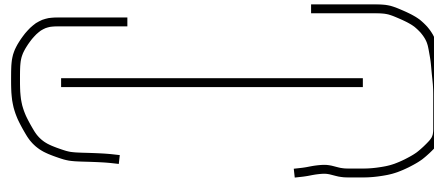
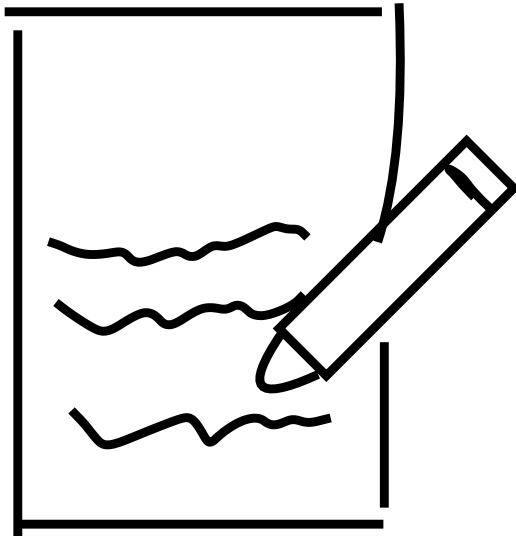
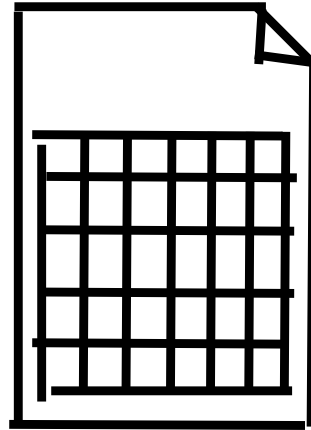
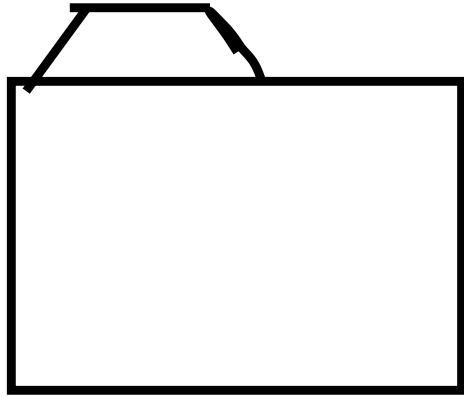
struct SomeTypedef {
    MyType declaration_of_mytype; // declaration occurs here
    int count;
};

// Anonymous Declaration
// a declaration with an inline definition
struct SomeDefinition {
    struct {
        int count;
        int *array;
    } declaration_of_anon_type;
};
```

## Quick terminology

Kind refers to dataset, group, attribute or link

In the GUI kinds use icons instead of full name.  
Hover over an icon it's full name.



## Simplified Landing Page

Note: Only one active button (the add type button)

A hand-drawn sketch of a web page layout. The page is divided into a header and a main content area. The header contains the text "Create Extended NWB Types" followed by a right-pointing arrow. The main content area is split into a left sidebar and a large right section. The sidebar is titled "My Types" and contains a button with a plus sign. The right section is empty.

Create Extended NWB Types →

My Types

+

After pressing add,  
user hits the top left icon dropdown to select dataset.

App presents user with  
DatasetTypeDef options

The sketch depicts a user interface for managing dataset types. On the left, a sidebar titled "My Types" contains a "+" button. The main area features a modal dialog box titled "New Dataset Type Name". Inside the dialog, there is a dropdown menu with a document icon and a downward arrow, which is highlighted by an arrow from the text "App presents user with DatasetTypeDef options". Below the dropdown is a text input field labeled "Describe this dataset type.". To the right of the description field are two input fields: "Name" and "Default Name" with a checkbox. Below these fields are three stacked input fields labeled "shape", "labels", and "dtype". At the bottom of the dialog is a button labeled "Pick a Dataset" and the word "extends" next to it. Below the dialog is a small icon of a document with a pencil and a "+" button. In the bottom right corner of the main interface is a "save" button.

User fills in DatasetTypeDef options  
And hits add note (attribute) to create an attribute declaration.

The type of the attribute declaration is implicit (anonymous).

The mockup shows a main window titled "My Types" with a "+" button. A secondary window titled "My Lab Meta Data" is open, showing a list of attributes and a "dt type" field. A third window titled "Lab Temp" is open, showing a list of attributes and a "dt type" field. A "+" button is next to the "Lab Temp" window. A "Save" button is at the bottom right.

My Types

+

My Lab Meta Data

Advanced

Name

Meta Data

Default Name ☒

shape

Labels

dt type

extends Lab Meta Data

Lab Temp

Temperature of Lab

dt type

Advanced

Required ☐

Value

Is default ☐

shape

dt type

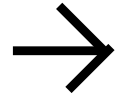
+

Save

User adds more attributes  
with plus button.  
The full shape of the dataset  
typedef is always visible.

User hits save when ready for validation

## Create NWB Extended Types



My Types

 MyLab Meta Data



User can now click on a type to re-edit,

Or use arrow to move on and  
define their namespace

<typedef-builder>  
 <Group-Typedef-builder>  
 </typedef-builder>

Group typedef

extends Pick a Type

Note:

Definiton will always have

<anon-dataset-dec>

Anonymous type dataset declaration

or

< Named type dataset declaration