

Creating and using 'Plug-Ins'

About Plug-ins

Suppose you have an irrigation system, and a rain gauge, and you want to want to incorporate it into your XTension system.

These are closely related systems, and much can be done to daily modify the amount of water that your plants need. But it may take you some time to create all of the scripts and units and scheduled events that would make it really useful.

Much of the work could have already been done for you by someone who created such a system and then 'exported' all of the elements that you would have to create yourself.

You will probably have to modify any of these plug-ins to adapt to the addresses and names that you use in your system already, but it is easier to modify something than to create it from scratch.

A 'plug-in' is a file that includes a selection of units, scripts, views, lists, and scheduled events which represent all or most of what is relevant to a particular 'sub system' such as "Irrigation System".

Some plug-ins may assume that you have already imported others. Like "Irrigation System" may assume that you already have imported the "Rain Gauge" plug-in, or vice-versa.

After having created and tested a 'sub-system', you might want to publish it, share it, sell it, or just save it for yourself.

Creating Plug-ins

By selecting "Export ..." from the Database menu, you will be presented with a series of selection dialogs.

You will be asked to choose which elements of your system that you want to include in the 'plug-in'.

These elements include :

- Views
- Groups
- Events

- Units
- Global scripts
- the name of the 'plug-in'

As you choose from the views, groups and events, each of the 'units' that are called-upon in those, will be automatically selected as individual units to be exported.

When you get to the 'Units' selection page, you will be presented with a choice of all database units which are not already included by some previous view, group or event.

Unit scripts will be automatically exported with their respective units, if they exist.

Global scripts are selected individually, and it is assumed that they contain references to units and scripts which are already included in the plug-in, or that they are common to all systems.
ie : daylight, or time/date etc.

Finally, you will be asked for a name of the plug-in, and where you want to save it.

You should always create a help file for a plug-in. Tell others what addresses and names you assumed, and perhaps any startup instructions.

You should have a good idea of what you want to export before you start the process, but do remember that you can cancel out at any time. There is no way to add elements to a plug-in once it has been created.

Importing Plug-ins

After receiving an XTension plug-in from a friend, and you have read the 'help file' for it, simply pull down the Database menu and select "Import...".

You will be asked to select the plug-in file, and if it is the proper format, XTension will then ask you:

Do you want to overwrite all units in your receiving database which have names which are identical to names in the 'plug-in' ?

Some of the units in the plug-in may be redundant, and you may have the choice to keep some of your 'old faithful's .

You can choose to :

- Always use new
- Always keep the old
- Let me choose as conflicts occur

Once you have imported a plug-in, you will see a new 'List' window which includes all of the units which were imported with the plug-in. It will also have the same name as the plug-in. Remember that you can always change the name of a List.

You will most likely have to change the addresses of any X-10 units, but you can either change the database, or change the modules directly.

You may also want to change the times and weekdays for some of the scheduled events to match your preferences.

Plugs-ins are not expected to be perfectly effortless to import, however they are a great way to get started, and a great way to share.

DO REMEMBER to save a copy of your current database before you import any plug-in (as before any major change to your system).

There is no way to UNDO an import other than to revert to a saved database copy.