Introduction to Scripting with HEScript

Scripting with HEScript · https://www.exeoutput.com/help/scripting

ExeOutput for PHP applications are script-driven and feature a **built-in script engine**. An application is managed by a collection of scripts that are generated and compiled into p-code by ExeOutput for PHP. When the application runs, its runtime module executes these scripts to simulate a web browser and respond to user actions.

You can, therefore, extend the functionality of your applications by **writing and calling your own HEScript functions**. This feature offers a high degree of flexibility.

Info

Note that you are not required to work with scripting to use ExeOutput for PHP and compile applications. Scripting is an advanced feature for users who want full control over their applications.

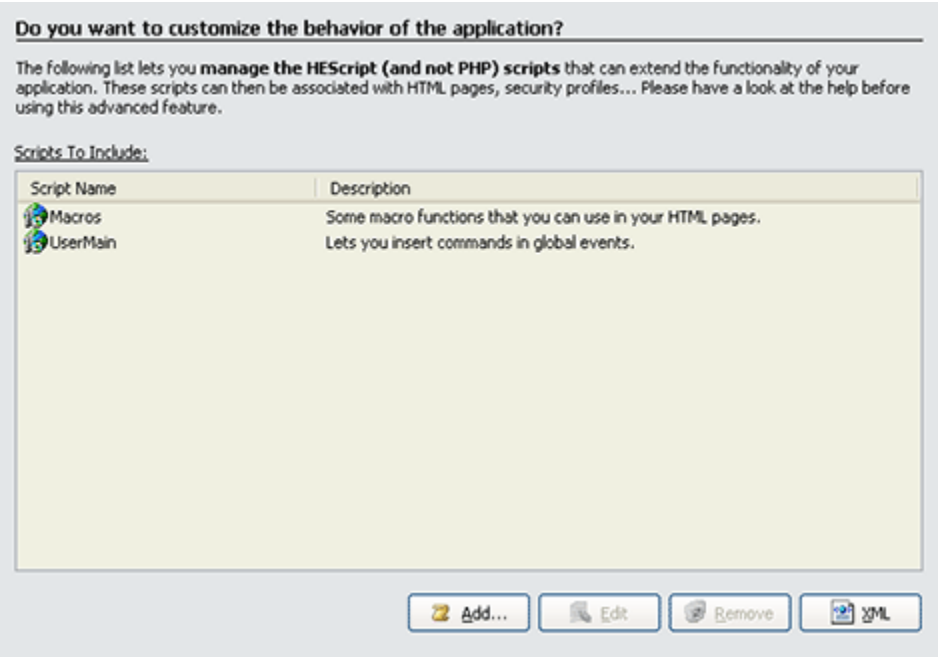
About the HEScript Language

The script language used by ExeOutput for PHP is called **HEScript**. It is based on the Object Pascal language syntax (similar to Embarcadero® Delphi and FreePascal) with some minor changes.

Unlike JavaScript and PHP, you cannot write HEScript functions directly into HTML or PHP pages. HEScript code must first be compiled into p-code. For this reason, you must use the **User Script Manager** to write and manage your scripts.

You can use HEScript in addition to PHP and JavaScript. In fact, all three can be easily combined.

The User Script Manager

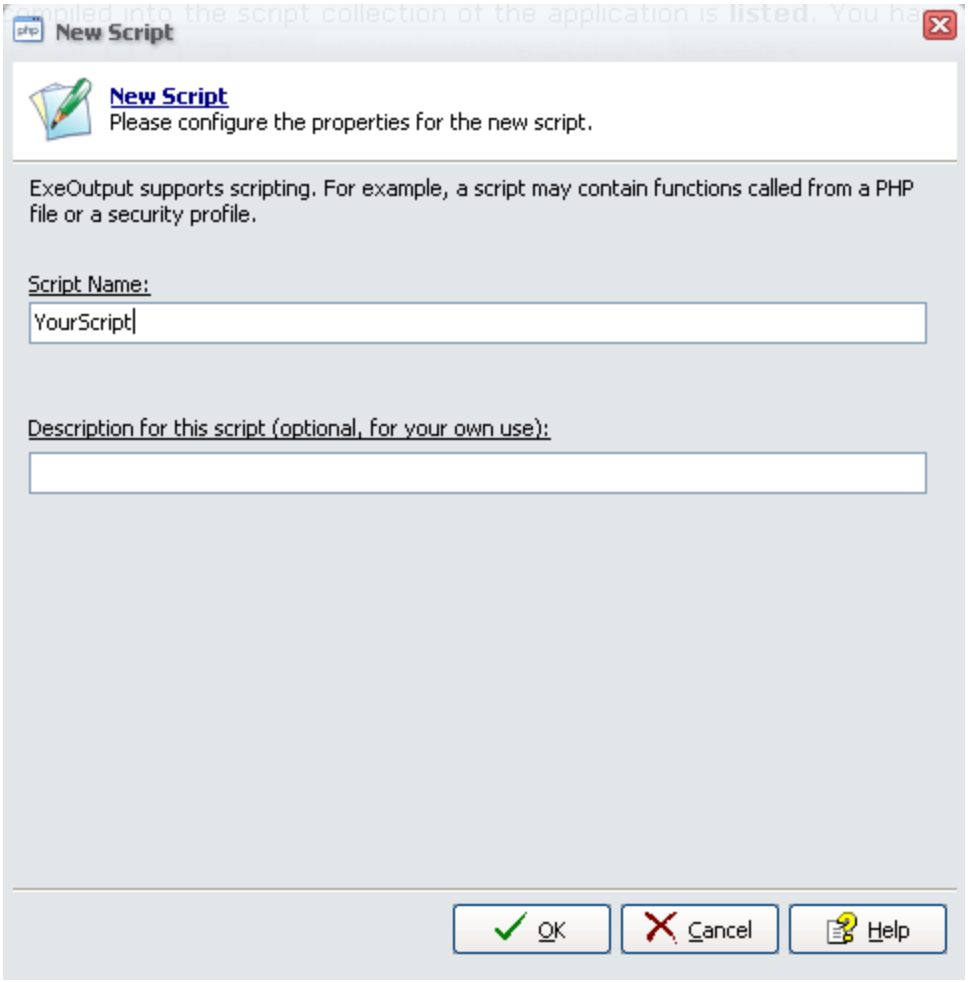


Each script compiled into the application is **listed** with its **name** and an optional **description**. Each script must have a **unique name**, which acts as a namespace.

Quickly Add HEScript Code

To add a new script:

1. Press **Add**.
2. Provide a **name** (alphanumeric only, no spaces or special characters, 255-character maximum).
3. Optionally, enter a description.



Press **OK** to create your script. The script editor will open, allowing you to start coding. Alternatively, you can press **Save** and return to it later.

More Information About the Script Editor

More Information About the Predefined `UserMain` Script

Managing Scripts

- **Edit**: Select a script and press **Edit** (or double-click it).
- **Remove**: Select a script and press **Remove**. (The `UserMain` script cannot be removed).
- **Import/Export**: Select a script and press **XML → Import/Export**. Scripts are stored in XML format and can be edited with an external editor. Imported scripts are automatically checked for syntax errors.

Example: The `UserMain` Script

By default, ExeOutput for PHP automatically adds a **UserMain** script when you create a project. This script contains predefined global events related to the application.

```
// UserMain
// This script contains special functions related to some of the events triggered by the
application.
// You can then optionally add new commands.

function OnBeforeNavigate(NewURL, TargetFrame: String): Boolean;
begin
  // Before the application displays a page.
  // Set Result to True to stop the operation.
  Result := False;
end;

procedure OnNavigateComplete;
begin
  // When a page has been displayed.
end;
```

A script file is, therefore, a **group of procedures and functions**. Each procedure or function name must follow these rules:

- Use only alphanumeric characters (no spaces).
- Each name must be unique within its script file.
- However, different script files can contain procedures or functions with the same name.

Important Notes and Best Practices

1. Script files act as **namespaces**. When calling or assigning a procedure/function to an event, use the following syntax: `ScriptName.FunctionName` (Example: `UserMain.OnNavigateComplete`).
2. Each script is managed by an **independent script engine**. Use global variables to exchange data between scripts.
3. Using local variables is possible but **not recommended**.

4. Scripts are stored inside the project file; no external files are required.
5. You can call HEScript functions from your PHP code, HTML links, and even from JavaScript.