Script Structure

Script structure

A script in ESL language (source text) consists of the initialization part and procedures. The initialization part is activated after the script initiation. Basically, it is an output tag on which the script control is devolved. Within the initialization part, procedures defined in the script may be called and thereby an own algorithm can be executed. The presence of procedures in a script is not obligatory. Virtually the whole algorithm may be executed within the initialization part.
When using HIS server, the initialization parts of both Local and Remote script are executed at the same time.

Rules for writing script actions:

- Each action (an executive item of the language) is written separately in one line. Then any written algorithm is a sequence of actions (lines), which are executed in successive steps. The sequence of actions execution may be changed by actions such as: GOTO, ON GOTO, GOSUB, CALL, IF GOTO, IF THEN ELSE,...
- A comment always begins with the character ';' and continues to the end of the line (multi line comments are not allowed).

Initialization part begins:

1. explicitly with BEGIN action
2. with any action, except the declaration of a local variable or the procedure body.

The following example sets the value of 1 of the object U.Int, within the initialization part (rule 2).

```
U.Int := 1     ; sets the value of the variable to 1
```

In ESL language, an assignment is known as an action. In source text, there is an assignment followed by a comment. According to the rule (for the beginning of script initialization part), assignment action is executed in the initialization part. The following example is semantically (with its meaning) identical with the example above:

```
BEGIN
U.Int := 1
END
```

The action END explicitly terminates the script execution. In this case, it is not necessary to do it either, because the script execution is terminated after the last action has been executed.

In the following example, the value setting is moved from the initialization part to the procedure, that is called:

```
PROCEDURE Init_U_Int
U.Int := 1
END Init_U_Int

BEGIN
CALL Init_U_Int
END
```

The script execution starts in the initialization part (the action BEGIN). The following action CALL calls the procedure *Init_U_Int* which sets the value of 1 of the object U.Int. The procedure is without parameters. After returning from the procedure, the script will execute the action END and will be terminated. If some actions are placed after the action END, they will not be executed.