

# HIS Server

**HIS Server** is a feature that enables to divide both the implementation of ESL scripts of active picture (PICTURE) and the objects of **Event** type (EVENT), opened on the client side, in two parts. These parts are:

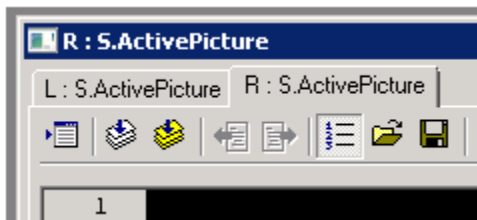
1. Local script - it is executed on the client side
2. Remote script - it is executed on the server side

The application programmer can divide the functionality into two units depending on the part of the system with which it is cooperating more intensive. We assume that the script on the client side will cooperate more intensive with the user interface through the fast communication channel and delimit the interaction with server to the minimum. And vice versa, the script on the server side will cooperate intensively with the application parts on the server side and limit the interaction with the user interface to the minimum.

**ESL Profiler** is used for debugging the system performance and the action response.

The parts are represented by two ESL scripts that belong to the configuration of particular object. From the user (application) point of view, this is, in a large extent, transparent and does not cause that the system could be excessively complicated.

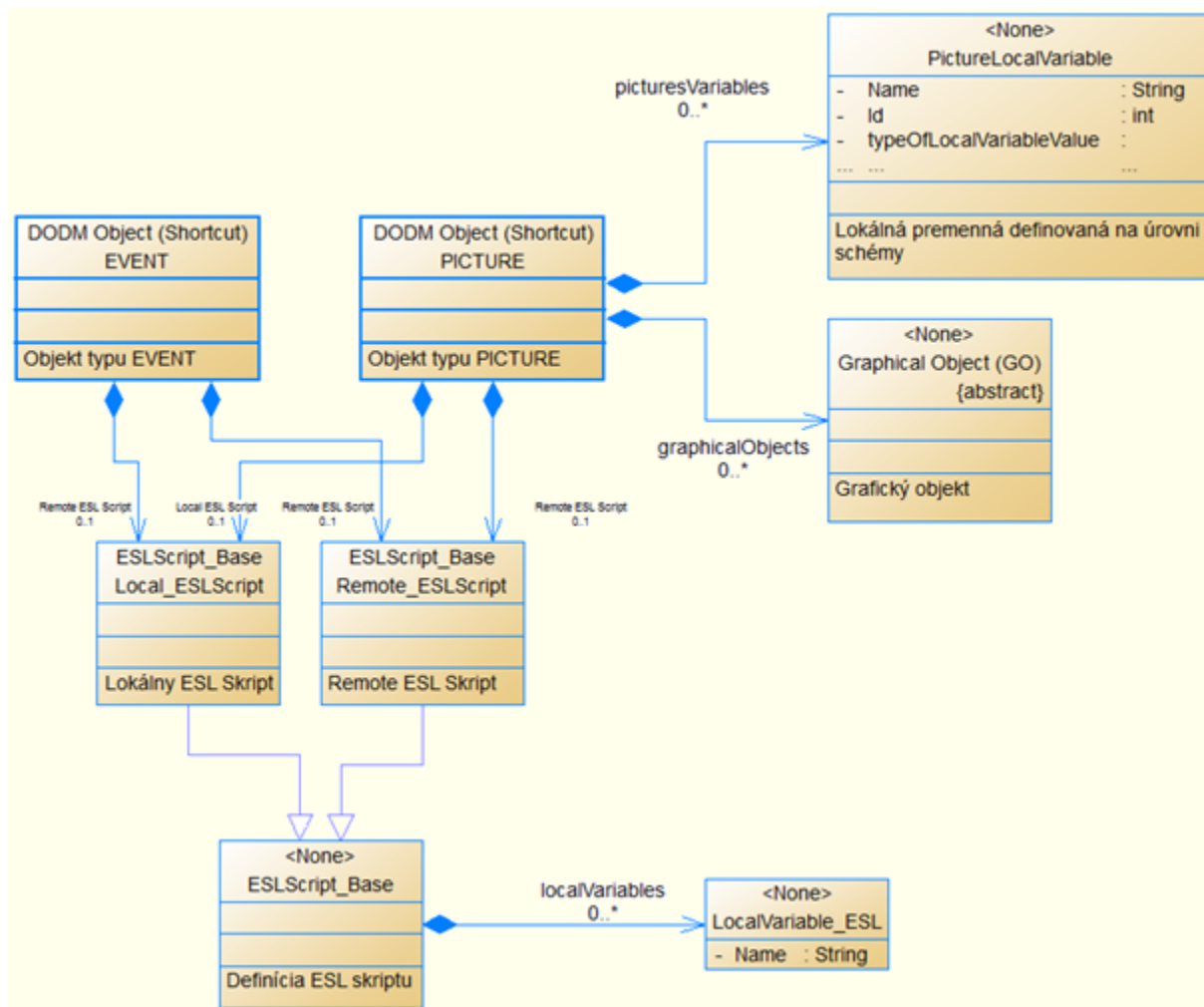
The division of ESL scripts in the script editor:



L: is the local script

R: is the remote script

The simplified data model of picture and event:



Explanations:

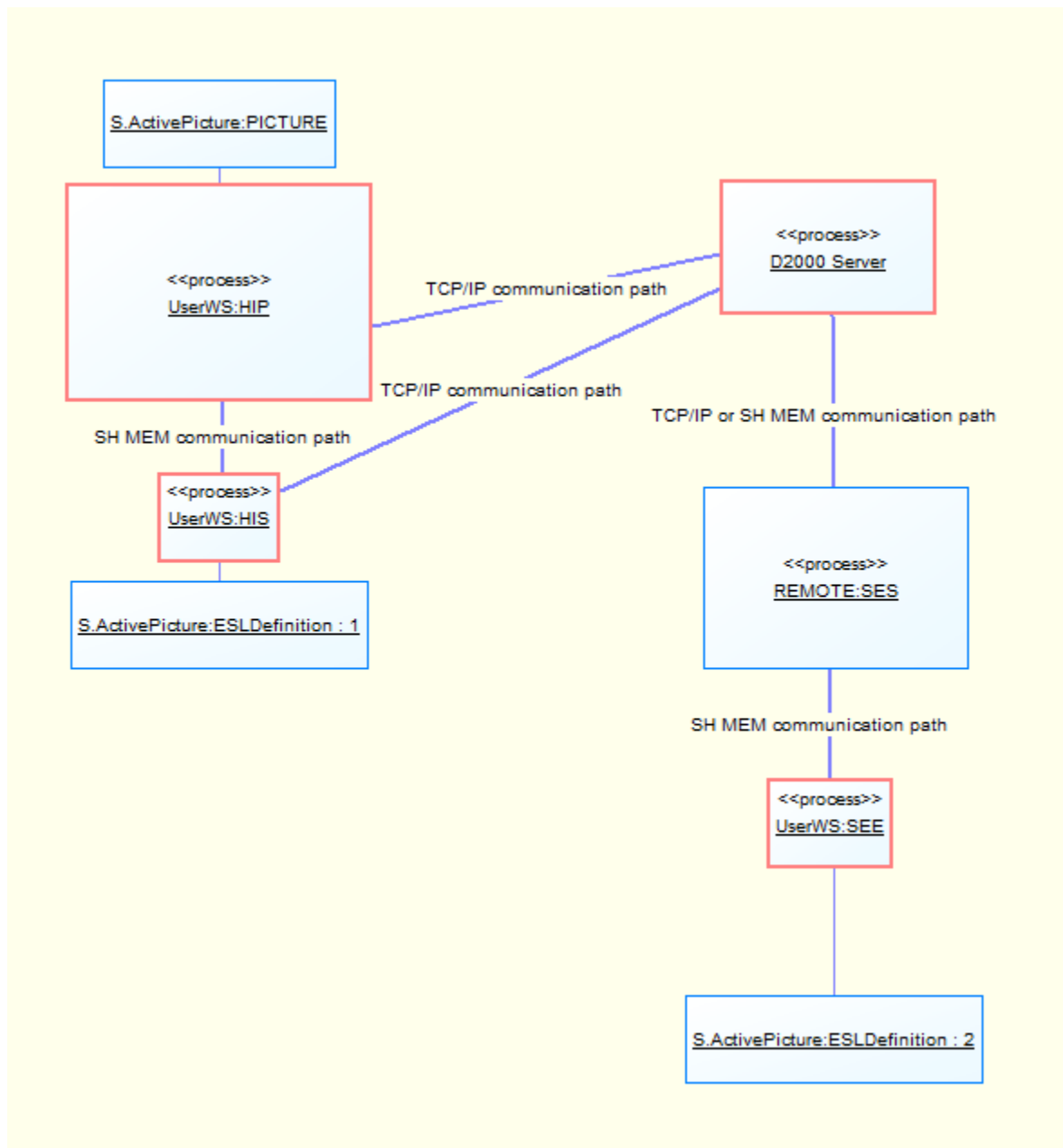
- EVENT – object of [Event](#) type
- PICTURE – object of [Picture](#) type
- Local\_ESLScript – ESL script that is executed on the client side
- Remote\_ESLScript – ESL script that is executed on the server side
- Graphical Object (GO) – [graphic object](#) in the picture
- PictureLocalVariable – [local variable](#) that is defined at the picture level
- LocalVariable\_ESL – [local variable](#) that is defined within ESL script

The functionality of **local ESL script** (on the client side) is ensured by **HIS** process (the name of physical process is usually derived from the name of particular HIP process), whose implementation is enclosed in the library *event.dll*.

The functionality of **remote ESL script** (on the server side) is ensured by virtual process **SEE** (**Script Execution Engine**). SEE processes are serviced by **SES** (**Script Execution Server**) process. His implementation is enclosed in the process *event.exe*.

The processes HIS and SEE execute interactively the requirements of client, which is, in this case, HIP process. For this purpose, there has been created a virtual communication channel between them, by means of which they communicate. From the client view, there is not important the physical placement of process (SEE). From the view of executing, both processes are independent and work in parallel.

This scheme shows the all involved components.



Process	Meaning
D2000 Server	Server process of D2000 System
UserWS.HIP	D2000 HI – user's console
UserWS_*.HIS	ESL Interpret on the Client side
UserWS_*.SEE	ESL Interpret on the Server side – Script Execution Engine
REMOTE.SES	Remote server that services *.SEE processes.

HIS and SEE processes are dynamic. Their names are derived from the particular HIP process and supplemented by randomly generated number, which ensures the uniqueness of name within the application. HIP process represents the user console on which there is open the picture **S.ActivePicture**. It uses the functionality of both the local ESL script and remote one, if the parameter *Server Script* is checked off in the dialog window *Picture settings* in **D2000 GrEditor**.

The above mentioned scheme also shows the position of the parts of picture configuration, which are depicted as objects.

Object	Meaning
S.ActivePicture:PICTURE	Definition of picture that is necessary for visualization.
S.ActivePicture:ESLDefinition:1	Local ESL script.
S.ActivePicture:ESLDefinition:2	Remote ESL script.

The distribution of the processes is usually following:

- D2000 Server and SES run on the application server,
- HIP and HIS run on the client side.

The processes are connected by communication channels. Depending on "distance" of the processes, there is used the communication via Shared Memory or TCP/IP. Shared Memory is fast communication channel but TCP/IP is slow one (this feature significantly becomes evident in TCP/IP connections in the large networks, in which the bound rate is usually sufficient but, vice versa, the latency is high, a few milliseconds). The channels are used for transmission of messages between processes, which ensure the functionality of picture as whole, as per its configuration.

The messages can be split into two categories:

#### 1. Messages for management of visualization

They are a result of using the ESL functions, mostly %HI\_Get\*, ...

Depending on the position of the commands, which initiate them, they are transmitted by these communication channels:

- HIS <-> HIP  
Fast communication channel.
- SEE <-> D2000 Server <-> HIP  
Latent communication channel.

#### 2. Messages for interaction with system

In this case, the system is represented by D2000 Server, which behaves as a "router" to handle the requirements of ESL script. The requirements are then processed by the processes according their type. They are the result of using the ESL actions, mostly DB\_READ, DBS\_READ, SQL\_\*, GETARCHDATA, GETARCHARR, ...

Depending on the position of the commands, which initiate them, they are transmitted by these communication channels:

- SEE <-> D2000 Server  
Fast communication channel.
- HIS <-> D2000 Server  
Latent communication channel.

In regard of the existence of synchronous actions or functions in ESL script, the latency is extremely unpleasant because it significantly prolongs the interactivity of application. You can avoid this situation, when you allocate properly the particular parts of ESL script on the client side (local ESL script) or server one (remote ESL script).

**ESL Profiler** can be used for debugging the system performance. It enables to analyze in detail the executing the actions of ESL script as regards the speed. It provides the information about a time of executing the action. If it is the synchronous action (the action must be confirmed to be finished by an implementor of action), it contains also the time indicating the passive waiting for confirmation.

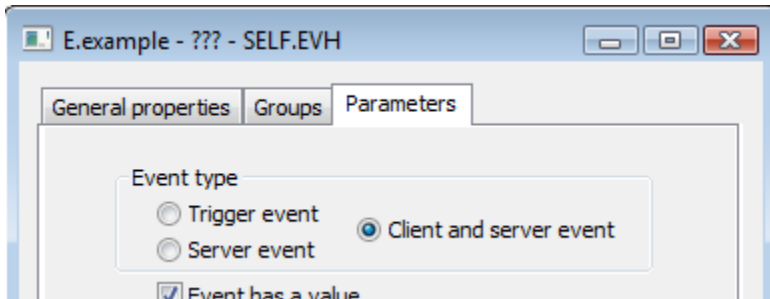
For example, the total time of reading from database by **DB\_READ** is characterized as sum of the times.

1. Evaluation of parameters and sending the requirement to read from database.
2. Waiting for the executing the requirement.

3. Receiving the reply and displaying the loaded data.

Point 2 represents (from the view of executing DB\_READ action) the passive waiting. Its duration is influenced by the latency of communication channel (as well as the time of reading from database).

The similar rules apply also for the object of Event type that is opened on the client side (by [OPENEVENT](#) action). ESL script can be divided on the local and remote one when setting the parameter "Event type" on "*Client and server event*".



RPC procedure calling between local and remote ESL script is ensured with the calling of [CALL](#) action. If duplicate declaration of procedures are in the scripts (local and remote one), the compiler shows an error ("Duplicate declaration of procedure") and disallows to save this configuration of ESL scripts (see [ESL extensions for HIS Server](#)).

The handling of all events (except for the defined exceptions) can be defined in both scripts. When the particular event occurs, the system ensures the calling of both handles. The exception is a synchronous event (the script waits for the return value). If these event handles are used in both scripts, the compiler evaluates this configuration with error ("Duplicate declaration of entry") and disallows to save it (see [ESL extensions for HIS Server](#)).

To keep the configuration of Remote ESL script when changing the event from "*Client and server event*" to other one, there is implemented the feature which copies the commented Remote script to Local ESL script. This ensure to maintain the configuration.

It also concerns the graphic pictures, if the parameter "Server Script" is checked off in the dialog window [Picture settings](#), opened from the menu **Settings** in [D2000 GrEditor](#).



#### Related pages:

[D2000 SES Process](#)  
[ESL extensions for HIS Server](#)  
[Script debugging](#)