Debug Environment Maintenance

Since a debug environment mainly consists of debug entries, it is established by setting breakpoints and watchpoints as described in the relevant maintenance sections.

Once established, a debug environment can be stored for subsequent usage. The file where debug environments are stored can be specified with the debugger command `PROFILE` (see *Navigation and Information Commands*). You can also delete a debug environment or reset its counters to their initial values.

Note:
See also the usage restrictions described in *Operational Requirements*.

The following items are also part of a debug environment and are therefore saved or loaded every time you save or load a debug environment:

- the test mode setting (`ON` or `OFF`);
- all options that can be set with the debugger command `PROFILE` (except the file for loading or saving debug environments);
- the settings of the **Statement execution statistics maintenance** function (`ON`, `OFF` or `COUNT`).

To invoke the debug environment maintenance function

- In the **Debug Main Menu**, enter function code `E`.

  Or:
  Enter the following direct command:

  ```
  EM
  ```

  The **Debug Environment Maintenance** menu appears.

This section describes the functions provided in the **Debug Environment Maintenance** menu and provides instructions for performing maintenance functions in different libraries.

With each function selected, you must enter the name of the debug environment to be maintained.

- Set Test Mode ON/OFF
- Load Debug Environment
- Save Debug Environment
- Reset Debug Environment
- Delete Debug Environment

- Maintain Debug Environments in Different Libraries

Set Test Mode ON/OFF

See the section *Switch Test Mode On and Off*.

Load Debug Environment

To load a debug environment from your user system file (FUSER)

- In the **Debug Environment Maintenance** menu, enter function code `L` and the name of an environment.

  Or:
  Enter the following direct command:

  ```
  LOAD ENVIRONMENT name
  ```

  The specified debug environment is loaded.

If you do not specify a name, the default environment with the name Noname is loaded.

Enter an asterisk (*) to obtain a list all available debug environments. On the list, you can mark the desired environment with the line command `LO` to load it into the debug buffer, or with the line command `DE` to delete it.

Save Debug Environment

To save a debug environment

- In the **Debug Environment Maintenance** menu, enter function code `S` and the name of an environment.

  Or:
  Enter the following direct command:

  ```
  SAVE ENVIRONMENT name
  ```

  The specified environment is reset (see below) and saved to the file location specified with the debugger command `PROFILE` (see the section *Navigation and Information Commands*).

If you do not specify a name, the environment is saved with the name Noname.

If a debug environment with the specified name already exists, you are prompted for confirmation to overwrite the old environment.

Reset Debug Environment

The debug environment should be reset before each test run. Resetting the environment leads to the following results:

- The current states of all debug entries are set to their initial states;
- All event counts are set to zero;
- The call statistics in the debug buffer are cleared as described in the section *Call Statistics Maintenance*.

To reset a debug environment

- In the **Debug Environment Maintenance** menu, enter function code `R` and the name of an environment.

  Or:
  Enter the following direct command:

  ```
  RESET ENVIRONMENT name
  ```

  The specified debug environment is reset.

If you do not specify an environment name, the current debug environment is reset.

Delete Debug Environment

To delete a debug environment

1. In the **Debug Environment Maintenance** menu, enter function code `D` and the name of the environment.

   Or:
   Enter the following direct command:

   ```
   DELETE ENVIRONMENT name
   ```

   The confirmation window appears.

2. In the confirmation window, enter `Y` (Yes) to confirm the deletion.

   The debug specified environment is deleted.

If you do not specify an environment name, the current debug environment is deleted.

Maintain Debug Environments in Different Libraries

The SYSMAIN utility provides the functions to copy or move debug environments between different libraries and/or system files and to delete, list or rename a debug environment.

When a debug environment has been moved or copied from one library to another, the breakpoints and watchpoints still refer to the old (source) library. You adapt the debug environment to the new (target) library by modifying the corresponding breakpoints (see also *Modify Breakpoint* in *Breakpoint Maintenance*) or watchpoints (see also *Modify Watchpoint* in *Watchpoint Maintenance*). When you perform the modify function, you do not have to change any of the existing definitions; upon executing the save command (PF5), the library reference automatically changes to the new library as can be seen in the **Library** field entry on the **Modify Breakpoint** or **Modify Watchpoint** screen.

Related Topic:

- *Processing Debug Environments* - *SYSMAIN Utility*, *Utilities* documentation