

Debugging With Server Express 4.0

This article discusses various ways of debugging in Micro Focus Server Express.

See also Knowledge Base article [23062](#).

INTRODUCTION

Animator enables you to debug Micro Focus Cobol applications. You can debug all types of application: single-threaded, multi-threaded, with Language C, created in intermediate code file, created in generated code file, callable shared object or system executable code file.

The Debugging Guide, part of the product's documentation, describes the Animator and debugging.

PREPARING FOR ANIMATION

To animate the application, you must compile and generate your application with the directives and the options required. The directive of the compilation is ANIM, and the option of the generation is g. By default, the directive ANIM is setting. When a program is compiled or generated to be animated, the file with name of the program as basename is created with the extension .idy. This file contains additional information required by Animator. It depends of the type of the executable file.

compilation : source file -> .int
generation: source or .int -> .gnt, object, or executable

Examples:

cob program.cbl: program.int program.idy (intermediate code)
cob -u program.cbl -g : program.gnt program.idy (generated code)
cob -z program.cbl -g : program.so program.idy (shared object)
cob -x program.cbl -g : program program.idy (executable)

The tool cobfflst gives the information about the creation of the file .idy

Example: cobfflst program (the file .idy must be in the current directory)

If program is created to be animated, Animator must know the directory of the file idy. So if you would like that a part of your application must not be animated, you should create without the directive or without the option.

Animator searches also the all sources files. The manual User's Guide describes the environment variables that may be used with Animator.

STARTING ANIMATOR

Animator displays information in window of a terminal. Before you invoke Animator, set the environment variable TERM to the appropriate value of your terminal.

To animate a program, you can use one or two terminals.

With a character terminal, you must switch from animator window to user's window (A function key is used to).

With an Xwindows terminal, Animator creates a new window. So two windows are used: one for user, one for Animator.

You can also use two terminals, one for you application, and one for Animator. See the section Cross-session debugging.

By default, if a Micro Focus application which runs on an Xwindows terminal, must be animated, Animator try to create a new window on the Xwindows terminal. Animator reads the environment variable TERM to know the type of the terminal. So it is necessary to verify the variable TERM and also the variable DISPLAY. This way should be modified with the cross-session debugging environment.

To invoke Animator:

See Debugging Handbook for all options:

- anim program : use the tool anim. By default, not be used, if mixed-language.
- Environment variable COBSW set to +A : Animator is called. This way can be used all times.
- CBL_DEBUGBREAK : function that should be called by a program anywhere. This function invokes Animator
- Runtime tunable: debug_on_error : If an error is detected, Animator is invoked. See User Guide.

CROSS-SESSION DEBUGGING

If your application must run in background, or if the sharing of the terminal is not possible, the cross-session debugging may be used.

Only the user which runs the application, can debug it. So sometimes, it is necessary to modify the environment of the application to use Animator (example: to animate a Web application)

The application tries to activate Animator when it is necessary. So, in cross-session debugging, the animator process must run when the application must be animated. In practice, you must run animator tool (cobanimsv) before the application. With Server Express 4.0, the function CBL_DEBUG_START can be used to wait the animator tool.

You can use cross-session debugging in two ways:

- With environment variable COBANIMSRV: set to a unique value. The application that must be animated, must run with this variable with the same value.
- Without COBANIMSRV : cobanimsrv will animate the first animated application that runs in the same directory.

With the cross-session debugging, you can stop the animation (to detach) and the execution of the application or only the animation. But you must try to detach before stopping the application.

CBL_DEBUG_START & CBL_DEBUG_STOP

These functions can use anywhere in a Micro Focus application.

CBL_DEBUG_START :

The values of the parameters enable you:

- to anim without cross-session debugging
- to wait for an animator tool
- to not animate if no animator tool runs
- to begin Animator at the first verb of the next called program (the module in which CBL_DEBUG_START is called, is not animated, the next module is animated)
- to choose the animator tool with or without COBANIMSRV

CBL_DEBUG_STOP:

The values of the parameters enable you:

- to stop animator and execution
- to stop to be animated (Animator will be invoked when the functions CBL_DEBUG_START or CBL_DEBUGBREAK is called)
- to stop the animator tool.
- to detach the animator tool

DEBUGGING A SERVICE USING CBL_DEBUGBREAK

This method should be used only if the server can be started by using an x-term .

a/ The user ID of the Enterprise System Administrator has the System Administrator permissions (as user root) and starts the server by using the Web Interface Enterprise Server:

make sure DISPLAY is not set
start the Directory Server (mfd command)
start the server by using the Web Interface Enterprise Server

b/ The user ID of the Enterprise System Administrator has the System Administrator permissions (as user root) and the server is started by using the Web Interface Enterprise Server :

make sure DISPLAY points to the local machine to the user

start the Directory Server (mfd command)

start the server by using the Web Interface Enterprise Server

c/ Other case , the cassi -a command must be used to start the server .by the user ID

The Directory Server was started

```
cassi -a -rserver
```

Insert the following line at the beginning of the Procedure Division of your program:

```
call "CBL_DEBUGBREAK"
```

Compile your program or generate it with a command such as :

```
cob myprog.cbl  
cob -gu myprog.cbl
```

Deploy the service to server, or if you have already deployed it, redeploy it (this involves deleting the deployed service). Ensure that you also deploy the program's .cbl and .cpy source files and the .idy file, which is needed for debugging.

On the message saying a call to CBL_DEBUGBREAK has been detected. The Server Express Animator starts, displaying the source of your program ready for debugging. You can now step through the deployed service in the usual manner.

Remember to remove the line that calls CBL_DEBUGBREAK when you have finished debugging the service.

As an alternative to inserting the call statement, you can compile the program with the Compiler directive INITCALL"CBL_DEBUGBREAK", before you deploy it.

DEBUGGING A SERVICE USING THE CASSI COMMAND

To debug a service using the cassi command using the default enterprise server ESDEMO, follow these steps.

Start the server ESDEMO. You can do this either using the Web interface Enterprise Server Administration, or using the casstart command.

Start a debug service execution process on the server ESDEMO. Enter:
cassi -a -rESDEMO

Compile your program with a command such as:

```
cob myprog.cbl
```

Deploy the service to ESDEMO, or if you have already deployed it, redeploy it (this involves deleting the deployed service). Ensure that you also deploy the program's .idy file, which is needed for debugging.

Create a client program for your service.

Run the client program and enter some input.

As soon as the service is entered, the Server Express Animator starts, displaying the source of your program ready for debugging. You can now step through the deployed service in the usual manner.

You must also either deploy the .cbl and .cpy files or set the COBCPY environment variable in ESDEMO's configuration to point to the location of the files.

Any service that runs in the debug service execution process is debugged, that is, the Server Express Animator starts up as soon as the service is entered.