

CA9-98-019

18

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

- 1 1. A method for debugging multiple related processes in one instance of a debugger comprising the
2 steps of:
3 initiating debugging of a plurality of related processes in said debugger;
4 checking during a timeout period for a debug event that has occurred within said related processes;
5 if a debug event has occurred, processing said debug event; and
6 after said steps of checking or processing, continuing debugging of all related processes.
- 1 2. The method of claim 1, wherein the step of processing said debug event comprises, if said debug
2 event relates to spawning of a related process, creating a view in a graphical user interface of said debugger
3 with respect to said spawned related process.
- 1 3. The method of claim 1 or 2, further comprising the step of selectively designating a process as
2 stopped.
- 1 4. The method of any one of claims 1 to 3, further comprising the step of:
2 determining if any of said related processes are stopped, waiting or running; and
3 if there are no processes stopped, waiting or running, terminating the execution of the debugger.
- 1 5. The method of any one of claims 1 to 4, wherein the step of processing said debug event comprises
2 stopping a related process based upon a type of said debug event.
- 1 6. The method of any one of claims 1 to 5, wherein the step of processing said debug event comprises.
2 if said debug event relates to termination or completion of a related process, deleting a view in a graphical
3 user interface of said debugger with respect to said terminated or completed related process.

CA9-98-019

19

1 7. The method of any one of claims 1 to 6 performed in the Windows NT operating system.

1 8. The method of claim 7, wherein the step of checking comprises executing the
2 WaitForDebugEvent() command for said timeout period and the steps of initiating and continuing
3 comprise executing the ContinueDebugEvent() command for said related processes.

1 9. A method for debugging a program comprising the steps of:
2 designating a plurality of related processes of the program for debugging in a single instance of a
3 debugger;
4 determining during a timeout period whether a debug event has occurred among said plurality of
5 related processes;
6 if a debug event has occurred, processing said debug event; and
7 performing said steps of designating, determining or processing repeatedly until all said related
8 processes are terminated.

1 10. An article of manufacture comprising a computer usable medium having computer readable
2 program code means therein for debugging multiple related processes in one instance of a debugger,
3 the computer readable program code means in said computer program product comprising:
4 computer readable code means for initiating debugging of a plurality of related processes in
5 said debugger;
6 computer readable code means for checking during a timeout period for a debug event that has
7 occurred within said related processes;
8 computer readable code means for processing said debug event if said debug event has occurred;
9 and
10 computer readable code means for continuing debugging of all related processes after said checking
11 or processing.

CA9-98-019

20

1 11. An article of manufacture of claim 10, wherein the computer readable code means of processing
2 said debug event comprises computer readable code means for, if said debug event relates to spawning of
3 a related process, creating a view in a graphical user interface of said debugger with respect to said
4 spawned related process.

1 12. An article of manufacture of claim 10 or 11, further comprising computer readable code means for
2 selectively designating a process as stopped.

1 13. An article of manufacture of any one of claims 10 to 12, further comprising:
2 computer readable code means for determining if any of said related processes are stopped, waiting
3 or running; and
4 computer readable code means for, if there are no processes stopped, waiting or running,
5 terminating the execution of the debugger.

1 14. An article of manufacture of any one of claims 10 to 13 wherein said computer readable code
2 means for processing said debug event comprises computer readable code means for stopping a related
3 process based upon a type of said debug event.

1 15. An article of manufacture of any one of claims 10 to 14 wherein the computer readable code means
2 of processing said debug event comprises computer readable code means for, if said debug event relates
3 to termination or completion of a related process, deleting a view in a graphical user interface of said
4 debugger with respect to said terminated or completed related process.

1 16. An article of manufacture of any one of claims 10 to 15 wherein said computer readable code
2 means is capable of execution in the Windows NT operating system.

1 17. An article of manufacture of claim 16, wherein said computer readable code means for checking

CA9-98-019

21

2 comprises computer readable code means for executing the WaitForDebugEvent() command for said
3 timeout period and said computer readable code means for initiating and continuing comprise computer
4 readable code means for executing the ContinueDebugEvent() command for said related processes.

1 18. An article of manufacture comprising a computer usable medium having computer readable
2 program code means therein for debugging a program, the computer readable program code means in said
3 computer program product comprising:

4 computer readable code means for designating a plurality of related processes of the program for
5 debugging in a single instance of a debugger;

6 computer readable code means for determining during a timeout period whether a debug event has
7 occurred among said plurality of related processes;

8 computer readable code means for, if a debug event has occurred, processing said debug event; and

9 computer readable code means for performing said steps of designating, determining or processing
10 repeatedly until all said related processes are terminated.

1 19. A computer system for debugging multiple related processes in one instance of a debugger
2 comprising:

3 means for initiating debugging of a plurality of related processes in said debugger;

4 means for checking during a timeout period for a debug event that has occurred within said related
5 processes;

6 means for processing said debug event if a debug event has occurred; and

7 means for continuing debugging of all related processes after said checking or processing.

1 20. A computer system for debugging a program comprising:

2 means for designating a plurality of related processes of the program for debugging in a single
3 instance of a debugger;

4 means for determining during a timeout period whether a debug event has occurred among said

CA9-98-019

22

5 plurality of related processes;

6 means for, if a debug event has occurred, processing said debug event; and

7 means for performing said steps of designating, determining or processing repeatedly until all said

8 related processes are terminated.