



Q'n Apps #S4

Q: Native Mode or Compatibility Mode?

The Difference between Native and Compatibility Modes

ETS800 series testers can operate in one of two modes; Native Mode (default) or Compatibility Mode. The modes are selected from the lower portion of the Main Test Setup window:



Click this Native button, and the system and software switch to Compatibility Mode:



If this button is not present on your screen, you can activate it by adding the special script "AllowCompatibilityMode" into the file C:\Symphony\SpScript.txt. If the special script

includes the “C” directive (AllowCompatibilityMode C) then the software will always start up with Compatibility Mode as the default.

It is also important to note that when a SET file is saved in Symphony, it will contain the mode setting. So when a “Compatibility Mode” SET file is loaded, the mode will already be set properly. The same is true for “Native Mode” SET files. This is a required resource control for test Automation purposes.

How do I know which Mode to use?

Many of our users are owners of previous HiLevel tester models. Their new Griffin or TITAN has more power and capability than ever before. Older test systems do not have the speed, vector depth, etc. that their new testers offer, yet users still need to run old test programs on the newer systems.

HiLevel has always made backward compatibility a top priority. To that end, the Compatibility Mode was created to allow new ETS800 series testers behave like previous models. When developing new tests on your ETS800 system, operate in Native Mode to take full advantage of your tester’s capabilities. Particularly if you intend to run faster than a 50MHz test rate.

Please be aware that Native Mode operates using a “four vector boundary”. This means that Last Vector and Stop Vector must always be set to a value that is a multiple of 4 (which will always be an even number). In other words, you may set Last Vector to 4096 or 4100, but not to 4097, 4098, or 4099.

Special Consideration for Programmed Mode

The “four vector boundary” rule also applies to using the pattern generator in Programmed rather than Sequential mode. Creating loops or jumps in Native Mode requires more vectors than it does in Compatibility Mode. It may be advantageous in some instances to use Compatibility Mode where the test rate is less than 50MHz and Programmed mode is used for sequencing. See the following illustrations.

With the ETS800 series and Symphony, the Page size for the Programmed mode has been increased from 64K to 256K. This means that Program commands that cause vector flow to jump to an address that is on another page of vector memory *must* be preceded by the proper Page instruction.

Vectors: C:\Demo\MEMIC.TRN						
Vector Address (Dec)	Program					P
				MMM	NNNNNNN	A
				EEEE	EEEEEEE	R
				MMM	XXXXXXX	E
		BD	MD	PPPP	TTTTTTT	R
				3120	6543210	R
0	NOOP	00000000	75757575	0000	XXXXXXXX	X
1	NOOP	00000000	75757575	0000	XXXXXXXX	X
2	NOOP	00000000	75757575	0000	XXXXXXXX	X
3	NOOP	00000000	75757575	0000	XXXXXXXX	X
4	LOAD 00003	00000000	75757575	0000	XXXXXXXX	X
5	DEC	00000000	75757575	0000	XXXXXXXX	X
6	CJMP 00005	00000000	75757575	0000	XXXXXXXX	X
7	NOOP	00000000	75757575	0000	XXXXXXXX	X
8	NOOP	3F17FFFF	75757575	0000	XXXXXXXX	X
9	NOOP	3F17FFFF	75757575	0000	XXXXXXXX	X
10	NOOP	3F17FFFF	75757575	0000	XXXXXXXX	X
11	NOOP	00000000	75757575	0000	LLLHHHH	X
12	NOOP	00000000	75757575	0000	LLLHHHH	X
13	NOOP	00000000	75757575	0000	LLLHHHH	X
14	NOOP	00000000	75757575	0000	LLLHHHH	X
15	NOOP	00000000	75757575	0000	LLLHHHH	X
16	NOOP	0017FFFF	75757575	0000	LLLHHHH	X

Program example in Compatibility Mode

Vectors: C:\Demo\MEMIC.TRN						
'Native' Vector Address (Dec)	Program & TimeSwitch					P
				MMM	NNNNNNN	A
				EEEE	EEEEEEE	R
				MMM	XXXXXXX	E
		BD	MD	PPPP	TTTTTTT	R
				3120	6543210	R
0	NOOP	00000000	75757575	0000	XXXXXXXX	X
1	NOOP	00000000	75757575	0000	XXXXXXXX	X
2	NOOP	00000000	75757575	0000	XXXXXXXX	X
3	NOOP	00000000	75757575	0000	XXXXXXXX	X
4	LOAD 00003	00000000	75757575	0000	XXXXXXXX	X
5	LOAD 00003	00000000	75757575	0000	XXXXXXXX	X
6	LOAD 00003	00000000	75757575	0000	XXXXXXXX	X
7	LOAD 00003	00000000	75757575	0000	XXXXXXXX	X
8	DEC	3F17FFFF	75757575	0000	XXXXXXXX	X
9	DEC	3F17FFFF	75757575	0000	XXXXXXXX	X
10	DEC	3F17FFFF	75757575	0000	XXXXXXXX	X
11	NOOP	00000000	75757575	0000	LLLHHHH	X
12	CJMP 000008	00000000	75757575	0000	LLLHHHH	X
13	CJMP 000008	00000000	75757575	0000	LLLHHHH	X
14	CJMP 000008	00000000	75757575	0000	LLLHHHH	X
15	CJMP 000008	00000000	75757575	0000	LLLHHHH	X
16	NOOP	0017FFFF	75757575	0000	LLLHHHH	X
17	NOOP	0017FFFF	75757575	0000	LLLHHHH	X
18	NOOP	0017FFFF	75757575	0000	LLLHHHH	X

Program example in Native Mode

Also See:

Q'nApp S7: Patgen Looping

Q'nApp S8: Pattern Matching

Q'nApp S15: Special Script

Q'nApp S16: Symphony Files

Q'nApp S19: Little Red Question Mark

Q'nApp S21: Making Symphony Portable

Q'nApp S26: Funtional Test Preparation

Q'nApp S29: Forward Compatibility

Q'nApp S39: PRG File Attachment

Q'nApp S44: Vector Files without Path