

TestDrive Script Language

2016.10.30

Questions : clonextop@gmail.com

차 례

1).시스템 관리	4
2).메모리 관리	5
3).파일 관리	6
4).프로파일 관리	7
5).빌트인 스크립트	8
6).프로파일 트리 스크립트	8
7).도큐먼트 레벨 스크립트	11
8).도큐먼트 레벨 : screen 스크립트	12
9).도큐먼트 레벨 : chart 스크립트	13
10).도큐먼트 레벨 : report 스크립트	17
11).도큐먼트 레벨 : button 스크립트	18

이 스크립트 언어는 TestDrive에 적용 가능한 .profile 파일이나 .sp 파일을 작성하기 위한 것으로, 이 문서에서는 TSL(TestDrive Script Language)의 문법과 작성 예를 보이며 설명한다.

1). 시스템 관리

Prefix : 'system.'

Script	Description
title ↳ \$title_name	title_name으로 메인 타이틀을 지정한다. 윈도우 타이틀에 표시된다. ex) system. title "Meitner"
subtitle ↳ \$title_name	title_name으로 서브 타이틀을 지정한다. 윈도우 타이틀에 표시된다. ex) system. subtitle "Tile Rendering"
clear	메시지 창을 클리어 한다. ex) system. clear
call ↳ \$file_name {, \$argument { , \$path { , \$err_token ... }}}	file_name의 파일을 path 위치에서 argument 를 주어 실행한다. 실행 도중 err_token이 검출되면 예외로 간주하며 실행을 마칠 때, system.error 와 같은 작업을 수행한다. 실행 파일이 있는 폴더가 실행 위치로 지정된다. path 가 지정되지 않을 경우 현재 실행중인 프로파일의 위치와 동일하게 실행되며, 실행 중 출력되는 메시지는 메시지 창에서 볼 수 있다. err_token 은 여러 개를 나열 할 수 있다. ex) system. call "compile.bat", "a.txt", "..\..\\" , "error:" system. call "run.exe"
document.add ↳ \$doc_name, @doc_script	doc_name의 이름을 가지는 도큐먼트를 생성하고, 도큐먼트 뷰에 이를 출력한다. doc_script에 해당하는 별도의 문법을 가진다. 이미 도큐먼트가 존재할 경우이면서 Lock 상태일 경우는 해당 도큐먼트를 Overwrite 하지 않는다. doc_script의 상세 구현은 빌트인 스크립트-도큐먼트 레벨 스크립트 단락을 참고한다. ex) system. document.add "Display" { USE_EDIT_MODE // Title report("Title", 0, 0, 962, 50){ Open("title.rtf"); //EnableEdit(); //ShowOutline(); } position(0, 50){ screen("ORIGINAL", 0, 0, 962, 542){ Create(1920, 1080, ABGR_8888,

	<pre> 0, 0, "default.jpg"); ShowOutline(); } } SetProgram("../Program\\Control.dll"); } </pre>
document.close ↳{\$doc_name}	<p>doc_name의 이름을 가지는 도큐먼트를 제거한다. 이름을 지정하지 않을 경우 모든 도큐먼트를 제거한다. 이 때 해당 document가 Lock 상태일 경우 닫기를 수행하지 않고, 닫기 예약을 하여, Unlock 상태가 될 때 자동으로 도큐먼트가 제거된다.</p> <p>ex) system.document.close "Document#1"</p>
document.call ↳{\$doc_name,} #command, {\$wParam {\$iParam}}	<p>doc_name의 이름을 가지는 도큐먼트(doc_name이 지정되지 않을 경우 눈에 보이는 활성화된 도큐먼트)의 DLL 구현(구현이 존재하지 않을 경우 종료)의 OnCommand 함수에 command 상수와 wParam, lParam의 문자열(wParam또는 lParam이 지정되지 않을 경우 각각 첫문자가 NULL로 시작되는 문자열을 반환한다.)을 보낸다.</p> <p>ex) system.document.call "Document#2", 5 system.document.call "Document", 2, "ABC" system.document.call "Document", 6, "System", "RUN"</p>
error ↳\$err_msg	<p>시스템 메시지 창에 err_msg 에러 문장을 출력하고 프로파일 실행을 중지시킨다. 프로젝트 시작 파일의 경우 프로젝트 열림을 취소시킨다.</p> <p>ex) system.error "에러가 발생하였습니다."</p>
msg ↳\$log_msg	<p>시스템 메시지 창에 log_msg 로그 문장을 출력한다.</p> <p>ex) system.msg "출력 완료.\n"</p>
sflash ↳\$file_name {, show_time {, fade_step }}	<p>스플래쉬 스크린을 표시한다.</p> <p>file_name 이미지 또는 동영상의 읽어 show_time (ms 단위, default = 2000)시간만큼 표시한 뒤 사라진다. 이때 fade_step(1~255, default = 15) 만큼 fade in/out 이 시도되며, 이미지 파일은 색상에 RGB 정보 이외에 반드시 알파 정보를 포함시켜야 정상적으로 표시된다.</p>

Table 시스템 관리 스크립트

2). 메모리 관리

Prefix : 'memory.'

Script	Description
title ↳\$title_name	title_name으로 메인 타이틀을 지정한다. 윈도우 타이틀에 표시된다. ex) system. title "Meitner"
create ↳#size { \$name}	size(byte 단위) 크기 만큼의 시스템 메모리를 생성시킨다. 이 메모리는 TestDrive/도큐먼트/외부 실행파일에서 모두 접근할 수 있는 메모리이며, 실제 물리 메모리에 관계없이 최대 4GB까지 생성 가능하다. 이 메모리는 TestDrive 이외 다른 프로세서에서 접근 가능한 페이지드 메모리를 name 으로 생성한다. name 이 명시되지 않을 경우 "TESTDRIVE_MEMORY_MAPPED" 로 간주한다. 이 이름이 같은 다른 프로젝트는 동시에 열 수 없으므로, 동시에 여러 프로젝트를 열람 하려면 각 프로젝트는 name에 유일한 이름을 정해야 한다. ex) memory. create 0x10000000 256MB의 시스템 메모리를 생성한다.
load.sim ↳\$file_name, #address	시스템 메모리의 address 위치에 시뮬레이션 파일 file_name을 읽어 저장한다. 구분자는 Space/Tab/, 를 혼용하여 사용할 수 있다. ex) memory. load.sim "test.txt", 1024 : 1Kbyte 위치에 test.txt를 읽어 시스템 메모리에 저장한다. "Test.txt" 0478 054B FFAA AB49 50FE 0073 0313 0542 0321 F3A3
load.bin ↳\$file_name, #address	시스템 메모리의 address 위치에 바이너리 file_name 파일을 읽어 파일 크기만큼 저장한다. ex) memory. load.bin "A.bin", 0 : 0byte 위치에 a.bin을 읽어 시스템 메모리에 저장한다.
load.dword ↳\$file_name, #address	시스템 메모리의 address 위치에 DWORD 형식으로 작성된 파일 file_name을 읽어 저장한다. 읽기 형식은 little-endian을 준수한다. ex) memory. load.dword "dword.txt", 0x10000 : 65KB 위치에 dword.txt 파일을 시스템 메모리에 저장한다. "dword.txt" 3 1024 -4 2 0 : 20 바이트로 03000000 00040000 FCFFFFFF 02000000 00000000 가 메모리에 저장된다.
load.word ↳\$file_name, #address	시스템 메모리의 address 위치에 WORD 형식으로 작성된 파일 file_name을 읽어 저장한다. 읽기 형식은 little-endian을 준수한다. ex) memory. load.word "word.txt", 0x40000 : 262144 Byte 위치에 word.txt 파일을 시스템 메모리에 저장한다. "word.txt"

	<p>3 1024 -4 2 0 : 10 바이트로 0300 0004 FCFF 0200 0000 가 메모리에 저장된다.</p>																				
load.byte ↳\$file_name, #address	<p>시스템 메모리의 address 위치에 BYTE 형식으로 작성된 파일 file_name을 읽어 저장한다. ex) memory.load.byte “byte.txt”, 40 : 40 Byte 위치에 byte.txt 파일을 시스템 메모리에 저장한다. “byte.txt” 3 1024 -4 0x2 0 : 5 바이트로 03 00 FC 02 00 가 메모리에 저장된다.</p>																				
load.float ↳\$file_name, #address	<p>시스템 메모리의 address 위치에 float 형식으로 작성된 파일 file_name을 읽어 저장한다. 읽기 형식은 little-endian을 준수한다. ex) memory.load.float “float.txt”, 0x40000 : 262144 Byte 위치에 float.txt 파일을 시스템 메모리에 저장한다. “float.txt” 3 1024 -4 2 0.4 : 바이트로 00004040 00008044 000080C0 00000040 CDCCCC3E 가 메모리에 저장된다.</p>																				
load.text ↳\$file_name, #address	<p>시스템 메모리의 address 위치에 정수 또는 실수를 지정하고 타입 크기를 가지는 형식으로 작성된 파일 file_name을 읽어 저장한다. 읽기 형식은 little-endian을 준수한다. ex) memory.load.text “text.txt”, 10 : 10 Byte 위치에 text.txt 파일을 시스템 메모리에 저장한다. “text.txt” F32(1.45) I16(0x10) S(10) I8(15) I32(100) S(20) I32(1023) F64(12.3) F24(-0.12) : 시스템 메모리 10byte 위치에 9A99B93F 1000 가 저장되며, 시스템 메모리 (10+10)byte 위치에 0F00 64000000 가 저장되며, 시스템 메모리 (10+20)byte 위치에 FF030000 9A99999999992840 86EBBB 가 저장된다.</p>																				
load.image ↳\$file_name, #address, %format { #byte_stride}	<p>이미지 파일 file_name(png/bmp/jpg/tif/gif)을 address 위치에 format 형식으로 byte_stride의 가로 크기를 가지는 이미지로 저장한다. 색상 format 형식</p> <table> <tr> <td>ALPHA</td><td>: 8bit alpha</td></tr> <tr> <td>LUMINANCE</td><td>: 8bit luminance</td></tr> <tr> <td>LUMINANCE_ALPHA</td><td>: 8/8bit luminance and alpha</td></tr> <tr> <td>RGB_565</td><td>: 5/6/5bit RGB</td></tr> <tr> <td>RGBA_4444</td><td>: 4/4/4/4bit RGBA</td></tr> <tr> <td>RGBA_5551</td><td>: 5/5/5/1bit RGBA</td></tr> <tr> <td>RGB_888</td><td>: 8/8/8bit RGB</td></tr> <tr> <td>ABGR_8888</td><td>: 8/8/8/8bit ABGR</td></tr> <tr> <td>RGBA_8888</td><td>: 8/8/8/8bit RGBA</td></tr> <tr> <td>ARGB_8888</td><td>: 8/8/8/8bit ARGB</td></tr> </table>	ALPHA	: 8bit alpha	LUMINANCE	: 8bit luminance	LUMINANCE_ALPHA	: 8/8bit luminance and alpha	RGB_565	: 5/6/5bit RGB	RGBA_4444	: 4/4/4/4bit RGBA	RGBA_5551	: 5/5/5/1bit RGBA	RGB_888	: 8/8/8bit RGB	ABGR_8888	: 8/8/8/8bit ABGR	RGBA_8888	: 8/8/8/8bit RGBA	ARGB_8888	: 8/8/8/8bit ARGB
ALPHA	: 8bit alpha																				
LUMINANCE	: 8bit luminance																				
LUMINANCE_ALPHA	: 8/8bit luminance and alpha																				
RGB_565	: 5/6/5bit RGB																				
RGBA_4444	: 4/4/4/4bit RGBA																				
RGBA_5551	: 5/5/5/1bit RGBA																				
RGB_888	: 8/8/8bit RGB																				
ABGR_8888	: 8/8/8/8bit ABGR																				
RGBA_8888	: 8/8/8/8bit RGBA																				
ARGB_8888	: 8/8/8/8bit ARGB																				

	<p>주의 1) byte_stride가 지정되지 않거나 실제 width * format 크기보다 작을 경우 자동으로 width * format 크기로 확장된다.</p> <p>주의 2) alpha 채널이 없는 이미지를 alpha를 포함한 format으로 이미지로 읽을 경우 모든 alpha는 0으로 설정된다.</p> <p>ex) memory.load.image “background.jpg”, 0, RGBA_8888 : 8/8/8/8bit RGBA 형식으로 background.jpg 이미지를 메모리 주소 0 위치에 이미지의 width * height * 4bytes 크기만큼 읽어들인다.</p>
store.sim ↳\$file_name, #address, #byte_size, #byte_stride	<p>시스템 메모리 address 위치에서 byte_size 크기만큼 시뮬레이션 형식으로 file_name 파일로 저장한다. 이때 byte_stride 바이트 개수마다 line feed를 삽입하여 줄을 나누어 저장한다.</p> <p>ex) memory.store.sim “a.txt”, 0, 10, 4 “a.txt” 예 0A9C3021 45ED3A00 0478 : 0 byte 위치의 10byte 크기의 메모리를 4byte 단위로 끊어서 위의 “a.txt” 형식과 같이 저장한다.</p>
store.bin ↳\$file_name, #address, #byte_size	<p>시스템 메모리 address 위치에서 byte_size 크기만큼 바이너리 형식으로 file_name 파일로 저장한다.</p> <p>ex) memory.store.bin “binary.txt”, 0x1000, 10 : 시스템 메모리 4096 byte 위치의 10 byte 만큼 binary.txt 파일로 저장한다.</p>
store.image ↳\$file_name, #address, #width, #height, %format { #use_alpha { #byte_stride}}	<p>시스템 메모리 address 위치에서 색상형식 format으로 가로/세로 width, height 크기와 byte_stride의 가로간격으로 저장된 이미지를 파일 file_name으로 저장한다. 이 때 use_alpha 가 0 일 경우 이미지에 alpha 채널을 저장하지 않는다. byte_stride가 생략되거나 width * height * format size보다 작은 경우는 자동으로 width * height * format size로 지정된다. use_alpha가 생략될 경우 0 으로 간주한다. 저장될 파일 타입은 file_name의 확장자 형식으로 자동 분류되어 저장된다. 확장자 형식을 알 수 없을 경우엔 PNG 형식으로 저장된다.</p> <p>format은 memory.load.image 의 format 형식과 동일하다.</p> <p>ex) memory.store.image “a.jpg”, 0x100, 640, 480, RGBA_8888 : 시스템 메모리 0x100 위치에서 VGA 타입의 RGBA_8888 형식으로 저장된 이미지를 alpha 채널을 제외하고 a.jpg에 JPEG 이미지 형식으로 저장한다.</p>

Table 메모리 관리 스크립트

3). 파일 관리

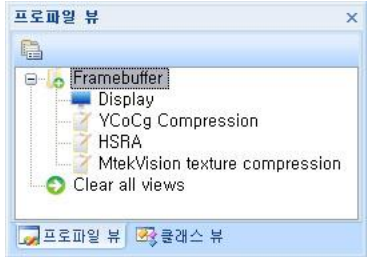

Prefix : 'if.'

Script	Description
file ↳\$file_name	이 문법은 반드시 if.else/if.end와 짝이 이루어 사용되어야 한다. file_name이 존재할 경우 참의 조건으로 다음 if.else/if.end 사이의 내용이 실행되며 그렇지 않을 경우 실행하지 않는다. ex) if. file "A.gpp" if.else system.call "compile a.gpp" if.end : A.gpp 파일이 없을 경우 systemcall을 호출하여 컴파일 한다.
else	if.file의 조건이 거짓일 경우 if.else~if.end 까지의 내용이 실행된다.
end	if.file/if.else를 참조한다.

Table 파일 관리 스크립트

4). 프로파일 관리

Prefix : 'profile.'

Script	Description
tree ↳{\$pf_branch} @pf_script	<p>프로파일 트리를 지정한다. pf_branch가 지정될 경우 프로파일 트리에서 pf_branch에 해당하는 하위 프로파일 트리를 지우고 pf_script에 나열된 내용으로 대체하며, pf_branch가 생략될 경우 모든 프로파일 트리 내용을 지우고 pf_script을 통해 대체한다. tree 내의 상세한 구현 내용은 빌트인 스크립트-프로파일 트리 스크립트 단락을 참고한다.</p> <p>ex) profile.tree</p> <pre> { branch("Framebuffer") { screen("Display", "Framebuffer\\Output\\main.sp"); report("YCoCg Compression", "YCoCg\\main.sp"); report("HSRA", "HSRA\\main.sp"); report("MtekVision texture compression", "MVTC\\main.sp"); expand(); } profile("Clear all views", "profiles\\clear_views.sp"); } </pre> 
clear	<p>프로파일 트리를 모두 지운다.</p> <p>ex) profile.clear</p>
call ↳\$file_name, {,%MSG}	<p>프로파일 문법으로 작성된 파일 file_name을 실행한다. 이 과정에서 출력되는 메시지는 MSG 메시지 창에 통해 출력된다. MSG가 생략되어 있을 경우 시스템 메시지 창으로 간주한다.</p> <p>MSG 형식</p> <p>sys : 시스템 메시지 창</p> <p>app : 어플리케이션 메시지 창</p> <p>ex) profile.call "program\\main.profile"</p>
set.initialize ↳\$file_name	<p>프로파일 문법으로 작성된 파일 file_name을 초기화 프로파일로 지정한다. 메뉴의 "프로파일 초기화" 버튼 또는 F10 키로 실행된다.</p> <p>ex) set.initialize "initialize.sp"</p> 



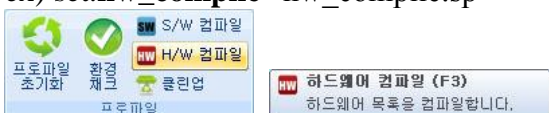

set.check ↳\$file_name	<p>프로파일 문법으로 작성된 파일 file_name을 실행 환경 체크 프로파일로 지정한다. 메뉴의 “환경 체크” 버튼 또는 F4 키로 실행된다.</p> <p>ex) set.check “check.sp”</p> 
set.sw_compile ↳\$file_name	<p>프로파일 문법으로 작성된 파일 file_name을 S/W 컴파일 프로파일로 지정한다. 메뉴의 “S/W 컴파일” 버튼 또는 F2 키로 실행된다.</p> <p>ex) set.sw_compile “sw_compile.sp”</p> 
set.hw_compile ↳\$file_name	<p>프로파일 문법으로 작성된 파일 file_name을 H/W 컴파일 프로파일로 지정한다. 메뉴의 “H/W 컴파일” 버튼 또는 F3 키로 실행된다.</p> <p>ex) set.hw_compile “hw_compile.sp”</p> 
set.cleanup ↳\$file_name	<p>프로파일 문법으로 작성된 파일 file_name을 실행 과정에서 발생하는 파일을 지우는 Clean-Up 프로파일로 지정한다. 메뉴의 “클린업” 버튼 또는 Shift+ESC키 조합으로 실행된다.</p> <p>ex) set.cleanup “cleanup.sp”</p> 
path ↳\$path	<p>프로파일 경로를 환경 변수로 등록합니다. 이 경로는 TESTDRIVE_PROFILE 로 등록됩니다.</p>


Table 프로파일 관리 스크립트

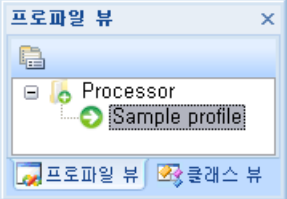
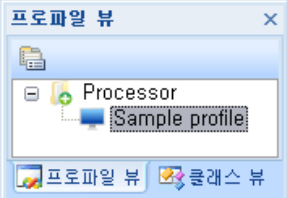
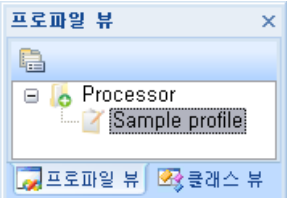
5). 빌트인 스크립트

스크립트 명령어 중 몇몇은 상세 설정을 위해 개별의 빌트인 스크립트 문법을 가진다. 다음 단락에서는 상위에 설명되지 않은 상세 빌트인 스크립트를 설명한다.

6). 프로파일 트리 스크립트

“profile.tree {Spf_branch} @pf_script” 문법의 fp_script에 대한 상세 설명이다. fp_script는 {...} 로 표현 영역을 제한하며, 이 표현 영역에 아래와 같은 명령어를 나열 할 수 있다.

Script	Description
branch ↳(\$name) {...}	Name 이름으로 트리의 새로운 branch를 가지며, branch 영역 {...}으로 다시 구분한다. 이 branch에 다시 새로운 branch를 가질 수 있으며, 프로파일들을 구분하기 위해 사용된다. ex) <pre>profile.tree{ branch("Processor") { branch("Core") { branch("ALU") { } } branch("Memory") { } } }</pre> 
profile ↳(\$name, \$file_name);	name 이름의 노드를 가지고 profile 형식을 나타내는 아이콘을 가지는 프로파일을 현재 위치에 추가한다. 더블 클릭시 해당 프로파일이 실행된다. ex) <pre>profile.tree{ branch("Processor") { profile("Sample profile", "test.sp"); } }</pre>

	
screen ↳(\$name, \$file_name);	name 이름의 노드를 가지고 screen 형식을 나타내는 아이콘을 가지는 프로파일을 현재 위치에 추가한다. 더블 클릭시 해당 프로파일이 실행된다. ex) <pre>profile.tree{ branch("Processor") { screen("Sample profile", "test.sp"); } }</pre> 
report ↳(\$name, \$file_name);	name 이름의 노드를 가지고 report 형식을 나타내는 아이콘을 가지는 프로파일을 현재 위치에 추가한다. 더블 클릭시 해당 프로파일이 실행된다. ex) <pre>profile.tree{ branch("Processor") { report("Sample profile", "test.sp"); } }</pre> 
chart ↳(\$name, \$file_name);	name 이름의 노드를 가지고 chart 형식을 나타내는 아이콘을 가지는 프로파일을 현재 위치에 추가한다. 더블 클릭시 해당 프로파일이 실행된다. ex) <pre>profile.tree{ branch("Processor") { chart("Sample profile", "test.sp"); } }</pre>

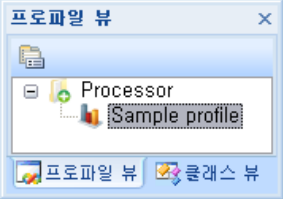
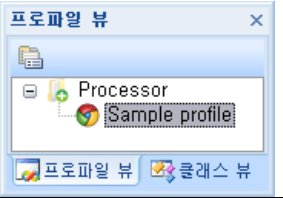


		
network ↳(\$name, \$file_name);	name 이름의 노드를 가지고 network 형식을 나타내는 아이콘을 가지는 프로파일을 현재 위치에 추가한다. 더블 클릭시 해당 프로파일이 실행된다. ex) <pre>profile.tree{ branch("Processor") { network("Sample profile", "test.sp"); } }</pre>	
expand ↳();	해당 branch의 모든 서브 노드들을 expand 된 상태로 만든다. 각 branch의 초기 상태는 각 branch가 닫힌 상태이다. ex) <pre>profile.tree{ branch("Processor") { chart("Sample profile", "test.sp"); network("Sample profile", "test.sp"); expand(); } }</pre>	<div>   </div> <div> expand(O), expand(X) </div>

Table 프로파일 트리 스크립트

예제에 포함된 Test.profile에 대해 설명한다..

```

system.clear           // 메시지 창을 클리어한다.
system.title           "Meitner"           // 메인 타이틀을 지정한다.
system.subtitle        "Tile Rendering"    // 서브 타이틀을 지정한다.
memory.create          0x4000000          // 시스템 메모리를 64MB 할당한다.
profile.tree           // 프로파일을 초기화 한다.
{
  branch("Sample (test)") {                // Sample (test) 아이템 추가
    branch("Core") {                      // Sample (test)/Core 아이템 추가

```

TestDrive Script Language

```
        profile("Bezier parameter", "Bezier\\main.sp"); // 서브 프로파일 추가
    }
    chart("Performance", "cleanup.sp"); // 서브 프로파일 추가
(chart 아이콘)
    chart("Utilization : Core", "cleanup.sp");
    chart("All utilization", "cleanup.sp");
    branch("Report : IP") { // Sample (test)/Report : IP 아이콘
추가
        report("Vertex loader", "cleanup.sp"); // 서브 프로파일 추가 (report
아이콘)
        report("Geometry", "cleanup.sp");
        report("Clipping", "cleanup.sp");
        report("Rasterizer", "cleanup.sp");
        report("Total", "cleanup.sp");
        expand(); // Sample (test)/Report 탭을 확장시켜 목록들을 보인다.
    }
    expand(); // Sample (test) 탭을 확장시켜 목록들을 보인다.
}
branch("Framebuffer") { // Framebuffer
    screen("Display", "Framebuffer\\Output\\main.sp");
    report("YCoCg Compression", "YCoCg\\main.sp");
    report("HSRA", "HSRA\\main.sp");
    report("MtekVision texture compression", "MVTC\\main.sp");
    expand();
}
branch("Secondary compute") {
    screen("Quaternion Julia Set", "Julia4D\\main.sp");
    expand();
}
}
profile("Clear all views", "profiles\\clear_views.sp"); // 모든 도큐먼트 제거
프로파일 추가
}
system.application "Program\\Control.dll" // 시스템 레벨 구현 등록
```

7). 도큐먼트 레벨 스크립트

“system.add.document \$doc_name, @doc_script”명령어의 doc_script에 대한 상세 설명이다.

doc_script는 {...} 로 표현 영역을 제한하며, 이 표현 영역에 아래와 같은 명령어를 나열 할 수 있다.

Script	Description
USE_EDIT_MODE	현재 도큐먼트 GUI 작성하는 프로파일 파일 내용이 외부에서 변경이 있을 경우, 자동으로 그 프로파일을 다시 실행하여 현재의 GUI를 갱신하도록 한다. 단 다른 도큐먼트나 시스템 레벨에서 참조에 의한 Lock 상태일 경우 USE_EDIT_MODE를 선언하더라도 외부에 의한 변경은 무시된다. ex) system.add.document “Test”{ USE_EDIT_MODE }
SetDescription ↳(\$desc);	현재의 도큐먼트의 간단한 설명을 넣는다. ex) system.add.document “Codec” { SetDescription (“Codec을 구현합니다.”); }
position ↳(#x,#y){...}	{...} 안에 해당하는 GUI 객체들을 x,y 만큼 위치를 이동시킨다. 이 명령을 중첩 사용할 있다. ex) system.add.document “Sample” { report(“”, 10, 10, 50,50); // (10,10) 위치에 50x50 크기 리포트 position (100,100){ position (20,30){ // (130,140) 위치에 50x50 크기 리포트 report(“”, 10, 10, 50,50); } // (110,110) 위치에 50x50 크기 리포트 report(“”, 10, 10, 50,50); } }
SetProgram ↳(\$file_name);	file_name에 해당하는 DLL을 읽어 도큐먼트 구현 인터페이스를 얻는다. ex) system.add.document “ATS”{ // lib.dll 파일에서 “ATS” 도큐먼트 구현 인터페이스를 얻는다. SetProgram (“.\Program\lib.dll”); }
screen ↳(\$name, #x, #y, #width, #height)	name 이름을 가지는 스크린을 x, y 위치에 width, height 크기의 도큐먼트 영역에 표시한다. 여기서의 width, height는 실제 screen의 스크린 영역을 의미하지 않는다. scr_script는 {...} 영역으로 구분되며, 상세 문법은 빌트인 스크립트-도큐먼트

@scr_script	레벨 : screen 스크립트 을 참조한다. ex) system.add.document “Test” { screen (“main_screen”, 10, 10, 600, 400){ // 스크린 스크립트 영역 } }
chart ↳(\$name, #x, #y, #width, #height) @cht_script	name 이름을 가지는 차트를 x, y 위치에 width, height 크기의 도큐먼트 영역에 표시한다. cht_script은 {...} 영역으로 구분되며, 상세 문법은 빌트인 스크립트-도큐먼트 레벨 : chart 스크립트 를 참조한다 ex) system.add.document “Test” { chart (“main_chart”, 10, 10, 600, 400){ // 차트 스크립트 영역 } }
report ↳(\$name, #x, #y, #width, #height) @rpt_desc	name 이름을 가지는 리포트를 x, y 위치에 width, height 크기의 도큐먼트 영역에 표시한다. rpt_desc은 {...} 영역으로 구분되며, 상세 문법은 1.5.6.3 을 참조한다 ex) system.add.document “Test” { report (“main_report”, 10, 10, 600, 400){ // 리포트 스크립트 영역 } }
button ↳(\$name, #x, #y, #width, #height) @btn_desc	name 이름을 가지는 버튼을 x, y 위치에 width, height 크기의 도큐먼트 영역에 표시한다. btn_desc은 {...} 영역으로 구분되며, 상세 문법은 1.5.6.4 를 참조한다 ex) system.add.document “Test” { button (“main_report”, 10, 10, 600, 400){ // 리포트 스크립트 영역 } }

Table 도큐먼트 레벨 스크립트

이 doc_script 에 다시 여러 GUI 객체를 선언하여 GUI 환경을 구성할 수 있는데, 다음 하위 단락에서 표현하는 TSL 문법에 따라 작성할 수 있다. 두개 이상의 GUI 객체의 영역이 겹칠 경우 나중에 선언한 GUI 객체 순으로 앞에 보이게 된다.

8). 도큐먼트 레벨 : screen 스크립트

Screen GUI 객체의 표현은 system.add.document의 doc_script 영역의 screen 명령의 scr_script에 한정한다. 스크립트 영역은 {...} 으로 표현한다. 사용자가 지정한 DIB형식의 2D 화면을 출력하는 역할을 하며, TestDrive에서는 기본적으로 축소/확대, 파일로 저장/로딩과 안티알리아싱을 지원한다.

Script	Description
Create ↳(#width, #height, %format {, #base_addr , #stride {, \$file_name}});	format형식의 색상 포맷과 width x height 해상도를 가지는 스크린 버퍼를 생성한다. format은 아래 중 하나를 사용한다. base_addr 과 stride가 존재하면 시스템 메모리의 레퍼런스 주소와 한 라인의 byte stride 길이를 지정할 수 있다. stride가 0 일 경우 이 값은 최소크기로 자동 계산된다. file_name이 존재할 경우 파일에서 읽어 현재 버퍼의 기본 이미지로 설정한다. 색상 포맷 형식 ALPHA : 8bit alpha LUMINANCE : 8bit luminance LUMINANCE_ALPHA : 8/8bit luminance and alpha RGB_565 : 5/6/5bit RGB RGBA_4444 : 4/4/4/4bit RGBA RGBA_5551 : 5/5/5/1bit RGBA RGB_888 : 8/8/8bit RGB ABGR_8888 : 8/8/8/8bit ABGR RGBA_8888 : 8/8/8/8bit RGBA ARGB_8888 : 8/8/8/8bit ARGB ex) Create (1920, 1080, RGBA_8888); Create (1920, 1080, RGBA_8888, 0x0, 4096); Create (1920, 1080, RGBA_8888, 0, 0, "default.jpg");
ShowOutline ↳();	스크린 객체의 외각선(회색)을 표시한다. 디폴트는 외각선 없음. ex) ShowOutline ();
UseAlpha ↳();	알파 사용을 활성화 시킨다. 디폴트는 알파 사용 안함. 활성화 할 경우 파일 저장시 알파값이 같이 저장된다. 활성화하지 않을 경우 RGB 정보만 저장된다. ex) UseAlpha ();
EnableAntialiasing ↳();	안티알리아싱을 활성화 한다. 이 항목은 GUI 환경의 메뉴에서 사용자가 변경 가능하며, GUI 환경에서 스크린을 32x 배율 이상 확대할 경우 자동 해제된다. ex) EnableAntialiasing ();
screen ↳빌트인 스크립트-	연결된 스크린 GUI 객체를 생성한다. 각 스크린에 사용자가 축소/확대/이동의 컨트롤이 연결된 모든 스크린에 적용된다. 다른 조건의 동일한 스크린 결과물을 유관으로 비교하기

<p>도큐먼트 레벨 : screen 스크립트 참조</p>	<p>수월하게 한다. GUI 환경에서 메뉴를 통해 임시로 연결 링크를 해제하거나 다시 연결될 수 있다.</p> <p>ex)</p> <pre> screen("Step#1", 0, 0, 100,100){ Create(320,240, RGBA_8888); screen("Step#2", 100, 0, 100,100){ Create(320,240, RGBA_8888); screen("Step#3", 200, 0, 100,100){ Create(320,240, RGBA_8888); } } } </pre>
---	--

Table 도큐먼트 레벨 : screen 스크립트

9). 도큐먼트 레벨 : **chart** 스크립트

Chart GUI 객체의 표현은 system.add.document의 doc_desc 영역의 chart 명령의 cht_script에 한정한다. 스크립트 영역은 {...} 으로 표현한다.

구현 참조 : <http://www.codeproject.com/KB/miscctrl/High-speedCharting.aspx>

Script	Description
ShowOutline ↳();	스크린 객체의 외각선(회색)을 표시한다. 디폴트는 외각선 없음. ex) ShowOutline ();
SetBackColor ↳(#r, #g, #b);	r,g,b 색상의 배경 색을 지정한다. 디폴트는 흰색. ex) SetBackColor (255,0,0); // 배경을 붉은색으로 지정한다.
SetBoarderColor ↳(#r, #g, #b);	Boarder 색상을 r, g, b로 지정한다. 디폴트 회색. ex) SetBoarderColor (0,0,0); // boarder 색상을 검은색으로 지정한다.
EnablePan ↳(#enable);	enable이 0 일 경우 GUI환경에서 panning을 허가하지 않는다. 디폴트 허가. ex) EnablePan (0); // panning을 금지한다.
EnableZoom ↳(#enable);	enable이 0 일 경우 GUI환경에서 zoomming을 허가하지 않는다. 디폴트 허가. ex) EnableZoom (0); // zoomming을 금지한다.
Title ↳@title_script	타이틀을 지정한다. title_script 는 {...} 의 표현영역을 가지며, title_script(Table 참조) 에 의해 그 문법이 정의된다. ex) Title { AddString("제목"); // 제목 string 지정 SetFont("Time New Roman", 140, 1, 1); // 폰트 지정 SetColor(0, 0, 128); // 제목 색상 지정 }
Axis ↳(%type, %position) @axis_script	position 위치에 type 형식으로 축을 만든다. axis_script 는 {...} 의 표현영역을 가지며, title_script(Table 참조) 에 의해 그 문법이 정의된다. type 형식 Standard : 표준 형식 Logarithmic : log 형식으로 축 간격을 가지는 형식 DateTime : 날짜 형식 Position 형식 Left : 왼쪽 위치 Right : 오른쪽 위치 Top : 위쪽 위치 Bottom : 아래쪽 위치 ex) Axis (Standard, Left){ SetMinMax(0, 100); }
Serie	type 형식의 데이터열을 만든다.

<code>↳(%type {#secondary_ h, {#secondary_ v} }) @serie_script</code>	<p>secondary_h 가 존재하고 0 이 아닐 경우 오른쪽 축을 기준으로 사용한다. 디폴트 왼쪽 축 기준.</p> <p>secondary_v 가 존재하고 0 이 아닐 경우 위쪽 축을 기준으로 사용한다.</p> <p>디폴트 아래쪽 축 기준.</p> <p>serie_script 는 {...} 의 표현영역을 가지며, serie_script(Table 참조) 에 의해 그 문법이 정의된다.</p> <p>type 형식</p> <p>Point : 점 형식</p> <p>Line : 라인 형식</p> <p>Surface : 면 형식</p> <p>Bar : 바 형식</p> <p>CandleStick : 캔드스틱 형식</p> <p>Gantt : 간트 형식</p> <p>ex) Serie(Line){ SetName("Data A"); // 데이터열 이름 지정 SetWidth(1); // 라인 너비 지정 SetSmooth(1); // 라인을 spline 형식으로 지정 }</p>
<p>Legend</p> <code>↳@legend_scri pt</code>	<p>레전드를 표시한다.</p> <p>legend_script 는 {...} 의 표현영역을 가지며, legend_script(Table 참조) 에 의해 그 문법이 정의된다.</p> <p>ex) Legend{ SetVisible(1); }</p>

Table 도큐먼트 레벨 : chart 스크립트

Title 스크립트

Script	Description
<p>AddString</p> <code>↳(\$str);</code>	<p>제목에 str을 추가한다.</p>
<p>SetFont</p> <code>↳(\$font_name {#point_size {#bBold {#bItalic {#bUnderline {#bVertical}} }}});</code>	<p>font_name 이름의 폰트를 지정한다.</p> <p>point_size : point 크기를 나타내며 point_size/10 의 실제 크기를 가진다.</p> <p>bBold : 1 일 경우 굵은 글씨를 지정한다.</p> <p>bItalic : 1 일 경우 이탤릭 글씨체를 지정한다.</p> <p>bUnderline : 1 일 경우 밑줄을 지정한다.</p> <p>bVertical : 1 일 경우 세로줄로 표시한다.</p>
<p>SetColor</p> <code>↳(#r, #g, #b);</code>	<p>제목 색상을 r,g,b로 지정한다.</p>

Table 도큐먼트 레벨 : chart - Title 스크립트

Axis 스크립트

Script	Description
SetMinMax ↳(#min, #max);	축의 표시 범위를 min~max로 지정한다. ex) SetMinMax (0, 100);
SetZoomLimit ↳(#limit);	줌을 사용할 때 최대 확대 범위를 지정한다. (초기값 0.001) ex) SetZoomLimit (0.000001);
SetAutomatic ↳();	축의 표시 범위를 데이터에 따라 자동 변경되도록 한다. ex) SetAutomatic ();
SetAxisColor ↳(#r, #g, #b);	축 색상을 r,g,b로 지정한다. ex) SetAxisColor (0,0,100);
SetTextColor ↳(#r, #g, #b);	축의 값 표시 색상을 r,g,b로 지정한다. ex) SetTextColor (128, 128, 128);
SetInverted ↳(#bInvert);	bInvert가 1 일 경우 대칭인 그래프를 그린다. ex) SetInverted (1);
Label.SetText ↳ (\$str);	축 이름을 지정한다. ex) Label.SetFont ("Label Title");
Label.SetFont ↳ Table 의 SetFont 참조	축의 폰트를 지정한다. Table 의 SetFont 참조 ex) Label.SetFont ("Time New Roman", 120);
Lable.SetColor ↳(#r, #g, #b);	축의 색상을 r,g,b로 지정한다. ex) Label.SetColor (48,48,48);
Grid.SetVisible ↳(#bVisible);	bVisible이 0 이면 축과 직각인 그리드를 감춘다. 디폴트 보임. ex) Grid.SetVisible (0);
Grid.SetColor ↳(#r, #g, #b);	그리드 색상을 지정한다. 디폴트 회색. ex) Grid.SetColor (0,0,0);
SetTickIncrement ↳(#bAuto #,increment);	Standard 타입의 축에만 사용된다. 보이는 Tick 증가 값을 설정한다. bAuto가 1 일 경우 자동으로 보이는 tick 간격을 지정한다. bAuto가 0 일 경우 수동으로 increment 값에 따라 보이는 tick 간격을 지정한다. ex) SetTickIncrement (0, 10); // 10 간격으로 숫자가 보인다.
SetAxisToScrollStep ↳(#PrevStep, #CurStep, #bScrollInverted);	Logarithm 타입의 축에만 사용된다. 스크롤 스텝을 지정한다. PrevStep : 스크롤 스텝의 시작을 지정한다. CurStep : 스크롤 스텝의 끝을 지정한다. bScrollInverted : 1 일 경우 스크롤이 반대로 된다. ex) SetAxisToScrollStep (0, 100, 0);
SetTickIncrement ↳(#bAuto, %Interval);	DateTime 타입의 축에만 사용된다. Tick 증가 간격을 지정한다. bAuto : 1 일 경우 Tick 간격이 자동으로 조절된다. Interval 타입 Second : 초 간격으로 Tick 표시

	<p>Minute : 분 간격으로 Tick 표시 Hour : 시간 간격으로 Tick 표시 Day : 날짜 간격으로 Tick 표시 Month : 달 간격으로 Tick 표시 Year : 연도 간격으로 Tick 표시 ex) SetTickIncrement(0, Second);</p>
--	--

Table 도큐먼트 레벨 : **chart - Axis** 스크립트

Serie 스크립트

Script	Description
SetName ↳(\$name);	데이터열 이름을 지정한다. ex) SetName ("income");
SetColor ↳(#r, #g, #b);	데이터열 색상을 지정한다. 기본적으로 자동으로 지정된다. ex) SetColor (255,0,0);
SetShadowColor ↳(#r, #g, #b);	그림자 색을 지정한다. ex) SetShadowColor (0,0,0);
EnableShadow ↳(#bEnable);	그림자를 활성화한다. 디폴트 그림자 없음. bEnable이 0 일 경우 그림자 없음. ex) EnableShadow (1);
SetShadowDepth ↳(#Depth);	Depth만큼 그림자 깊이를 지정한다. ex) SetShadowDepth (3);
SetPointSize ↳(#width, #height);	Point 타입의 Serie에만 사용된다. 가로 width 크기, 새로 height 크기를 가지는 포인트를 지정한다. ex) SetPointSize (5,5);
SetPointType ↳(%type);	Point 타입의 Serie에만 사용된다. Type 형식의 포인트를 지정한다. Type 형식 <div style="margin-left: 40px;"> Ellipse : 타원형 (default) Rectangle : 사각형 Triangle : 삼각형 </div> ex) SetPointType (Ellipse);
SetBorderColor ↳(#r, #g, #b);	Point 타입의 Serie에만 사용된다. 포인트 경계색상을 r,g,b로 지정한다. ex) SetBorderColor (0,0,0);
SetWidth ↳(#width);	Line 타입의 Serie에만 사용된다. 라인의 너비를 width로 지정한다. ex) SetWidth (4);
SetSmooth ↳(#bSmooth);	Line 타입의 Serie에만 사용된다. bSmooth(default=0)가 0 이 아닐 경우 라인을 스플라인 형식으로 부드럽게 표시한다. ex) SetSmooth (1);
SetPenStyle ↳(%Style);	Line 타입의 Serie에만 사용된다.

	<p>라인 펜 스타일을 지정한다.</p> <p>Style 형식</p> <p>Solid : Dash : Dot : DashDot : DashDotDot :</p> <p>ex) SetPenStyle(DashDot);</p>
<p>SetFillStyle ↳(%Style);</p>	<p>Surface 타입의 Serie에만 사용된다.</p> <p>지정된 Style 형식으로 채움을 지정한다.</p> <p>Style 형식</p> <p>Solid : 단색 HatchDownDiag : 오른쪽 아래 사선 HatchUpDiag : 오른쪽 위 사선 HatchCross : 십자선 HatchDiagCross: 십자 사선 HatchHorizontal: 수평선 HatchVerical : 수직선</p> <p>ex) SetFillStyle(HatchCross);</p>
<p>SetHorizontal ↳(#bHorizon);</p>	<p>Surface 타입의 Serie에만 사용된다.</p> <p>bHorizon(default=0)이 0 이 아니면 면의 시작이 수직 축으로부터 수평하게 그린다.</p> <p>ex) SetHorizontal(1);</p>
<p>SetHorizontal ↳(#bHorizon);</p>	<p>Bar 타입의 Serie에만 사용된다.</p> <p>bHorizon(default=0)이 0 이 아니면 수평 바로 표시한다.</p> <p>ex) SetHorizontal(1);</p>
<p>SetBorderColor ↳(#r, #g, #b);</p>	<p>Bar/Gantt 타입의 Serie에만 사용된다.</p> <p>경계선 색상을 r,g,b로 지정한다.</p> <p>ex) SetBorderColor(0,0,0);</p>
<p>SetBorderWidth ↳(#Width);</p>	<p>Bar/Gantt 타입의 Serie에만 사용된다.</p> <p>Width(default=1) 크기를 가지는 경계선을 지정한다.</p> <p>ex) SetBorderWidth(4);</p>
<p>SetBarWidth ↳(#Width);</p>	<p>Bar/Gantt 타입의 Serie에만 사용된다.</p> <p>Width(default=20) 크기를 가지는 바 너비를 지정한다.</p> <p>ex) SetBarWidth(10);</p>
<p>ShowGradient ↳(#bShow);</p>	<p>Bar/Gantt 타입의 Serie에만 사용된다.</p> <p>bShow(default=0)가 0 이 아닐 경우 Gradient 타입으로 바를 표시한다.</p> <p>Ex) ShowGradient(1);</p>
<p>SetGradient ↳(#r,#g,#b, %Type);</p>	<p>Bar/Gantt 타입의 Serie에만 사용된다.</p> <p>바 색상과 Gradient 색상 r,g,b와 Type 표시 형식을 지정한다.</p> <p>Type 형식</p> <p>Horizontal : 수평 그라데이션 Vertical : 수직 그라데이션</p>

	<p>HorizontalDouble : 수평 원통형 그라데이션 VerticalDouble : 수직 원통형 그라데이션 ex) SetGradient(255,255,0,HorizontalDouble);</p>
<p>SetStacked ↳(#bStack);</p>	<p>Bar 타입의 Serie에만 사용된다. bStack(default=0)가 0 이 아닐 경우 바가 다른 Bar Serie위에 쌓여 표시되도록 한다. ex) SetStacked(1);</p>
<p>SetWidth ↳(#Width);</p>	<p>CandleStick 타입의 Serie에만 사용된다. Width 너비를 가지는 CandleStick을 지정한다. ex) SetWidth(10);</p>

Table 도큐먼트 레벨 : chart – Serie 스크립트

Legend 스크립트

Script	Description
<p>SetVisible ↳(#bVisible);</p>	<p>bVisible(default=0)이 0 이 아니면 레전드를 보인다. ex) SetVisible(1);</p>
<p>SetBackColor ↳(#r, #g, #b);</p>	<p>배경 색을 r,g,b로 지정한다. ex) SetBackColor(255,255,255);</p>
<p>SetShadowColor ↳(#r, #g, #b);</p>	<p>그림자 색을 r,g,b로 지정한다. ex) SetShadowColor(0,0,0);</p>
<p>EnableShadow ↳(#bEnable);</p>	<p>bEnable(default=1)이 0 이 아니면 그림자를 보인다. ex) EnableShadow(0);</p>
<p>SetShadowDepth ↳(#Depth);</p>	<p>그림자 깊이를 Depth로 지정한다. ex) SetShadowDepth(5);</p>
<p>SetTransparent ↳(#bTrans);</p>	<p>bTrans(default=0)가 0 이 아니면 배경을 투명하게 한다. ex) SetTransparent(1);</p>
<p>SetHorizontalMode ↳(#bHorizon);</p>	<p>bHorizon(default=0)이 0 이 아니면 레전드 목록들을 수평으로 길게 나열한다. ex) SetHorizontal(1);</p>
<p>DockLegend ↳(%Side);</p>	<p>레전드가 차트를 기준의 붙는 위치를 Side(default=Right) 형식으로 지정한다. Side 형식 Left : 차트 왼쪽 Right : 차트 오른쪽 Top : 차트 위쪽 Bottom : 차트 아래쪽 ex) DockLegend(Bottom);</p>
<p>UndockLegend ↳(#x, #y);</p>	<p>레전드 위치를 임의로 x,y 에 지정한다. ex) UndockLegend(300,30);</p>

Table 도큐먼트 레벨 : chart – Legend 스크립트

10).도큐먼트 레벨 : report 스크립트

Script	Description
Open ↳(\$file_name, #Auto_update);	워드패드에서 RTF 형식으로 만들어지는 문서 파일 file_name을 읽어 리포트 영역에 표시한다. Auto_update가 0 이 아니면 해당 문서가 변경되었을 때 다시 읽어 표시한다. ex) Open ("title.rtf", 1);
EnableEdit ↳();	사용자가 리포트 영역을 수정할 수 있도록 한다. ex) EnableEdit ();
ShowOutline ↳();	리포트 영역의 외각선을 표시한다. ex) ShowOutline ();

Table 도큐먼트 레벨 report 스크립트

11).도큐먼트 레벨 : button 스크립트

Script	Description
SetText ↳(\$msg);	버튼 위에 표시할 메시지를 지정한다. 기본으로 GUI 객체 이름으로 초기화 되어있다. ex) SetText ("Press to continue...");
SetEnable ↳(#bEnable);	bEnable이 0 일 경우 버튼을 비활성화 한다. ex) SetEnable (0);
SetBitmap ↳(\$file_name);	버튼 이미지 파일 file_name을 읽어 버튼 위에 표시한다. ex) SetBitmap ("image.png");

Table 도큐먼트 레벨 button 스크립트