Subject: Re: umk got changed....
Posted by nixnixnix on Tue, 13 Dec 2011 01:07:29 GMT
View Forum Message <> Reply to Message

Hi Honza,

If you could replace the ":" with ";" to make umk windows compatible the only problem then the fact that umk.exe does not recognise the output path.

Two reasonably small problems?

Nick

```
CONSOLE_APP_MAIN
{
 Ide ide;
 TheIde(&ide);
 ide.console.SetSlots(CPU_Cores());
 ide.console.console = true;
 bool clset = false;
 const Vector<String>& arg = CommandLine();
 if(arg.GetCount() >= 3) {
  for(int i = 3; i < arg.GetCount(); i++)
   if(arg[i][0] == '-') {
    String x = arg[i];
    for(int i = 1; i < x.GetCount(); i++) {
     if(x[i] == 'l')
      SilentMode = true;
     if(x[i] == 'v')
      ide.console.verbosebuild = true;
    }
   }
  String v = GetUmkFile(arg[0] + ".var");
  if(!FileExists(v)) {
   Vector<String> h = Split(arg[0], ';'); // this change makes it windows compatible ?
   for(int i = 0; i < h.GetCount(); i++)
    h[i] = GetFullPath(TrimBoth(h[i]));
   String x = Join(h, ";");
   SetVar("UPP", x, false);
   PutVerbose("Inline assembly: " + x);
   String outdir = ConfigFile("_out"); // this line makes the output path not work?
   RealizeDirectory(outdir);
   SetVar("OUTPUT", outdir, false);
  }
  else {
   if(!LoadVars(v)) {
    Puts("Invalid assembly\n");
```

```
  SetExitCode(2);
  return;
 }
 PutVerbose("Assembly file: " + v);
 PutVerbose("Assembly: " + GetVar("UPP"));
}
PutVerbose("Output directory: " + GetVar("OUTPUT"));
v = SourcePath(arg[1], GetFileTitle(arg[1]) + ".upp");
PutVerbose("Main package: " + v);
if(!FileExists(v)) {
 Puts("Package does not exist\n");
 SetExitCode(2);
 return;
}
ide.main = arg[1];
ide.wspc.Scan(ide.main);
clset = true;
bool stoponerror = false;
const Workspace& wspc = ide.IdeWorkspace();
if(!wspc.GetCount()) {
 Puts("Empty assembly\n");
 SetExitCode(4);
 return;
}
const Array<Package::Config>& f = wspc.GetPackage(0).config;
if(f.GetCount())
 ide.mainconfigparam = f[0].param;
String m = arg[2];
String bp = GetBuildMethodPath(m);
PutVerbose("Build method: " + bp);
if(bp.GetCount() == 0) {
 SilentMode = false;
 Puts("Invalid build method\n");
 SetExitCode(3);
 return;
}
ide.method <<= m;
ide.debug.def.blitz = ide.release.def.blitz = 0;
ide.debug.def.debug = 2;
ide.release.def.debug = 0;
ide.debug.package.Clear();
ide.release.package.Clear();
ide.debug.linkmode = ide.release.linkmode = 0;
ide.release.createmap = ide.debug.createmap = false;
ide.targetmode = 0;
bool clean = false;
bool makefile = false;
int  exporting = 0;
```

```
String mkf;
for(int i = 3; i < arg.GetCount(); i++)
 if(arg[i][0] == '+' || arg[i][0] == '>')
  ide.mainconfigparam = Filter(~arg[i] + 1, CommaSpace);
 else
 if(arg[i][0] == '-') {
  String x = arg[i];
  for(int i = 1; i < x.GetCount(); i++)
   switch(x[i]) {
   case 'a':
    clean = true;
    break;
   case 'r':
    ide.targetmode = 1;
    break;
   case '1':
    ide.targetmode = 2;
    break;
   case '2':
    ide.targetmode = 3;
    break;
   case 'm':
    ide.release.createmap = ide.debug.createmap = true;
    break;
   case 'b':
    ide.release.def.blitz = ide.debug.def.blitz = 1;
    break;
   case 's':
    ide.debug.linkmode = ide.release.linkmode = 1;
    break;
   case 'd':
    ide.debug.def.debug = 0;
    break;
   case 'S':
    ide.debug.linkmode = ide.release.linkmode = 2;
    break;
   case 'e':
    stoponerror = true;
    break;
   case 'M':
    makefile = true;
    break;
   case 'v':
    ide.console.verbosebuild = true;
    break;
   case 'l':
    break;
   case 'x':
```

```
      exporting = 1;
      break;
     case 'X':
      exporting = 2;
      break;
     case 'H':
      if(i + 1 < x.GetCount() && x[i + 1] >= '1' && x[i + 1] <= '9')
       ide.console.SetSlots(x[++i] - '0');
      else
       ide.console.SetSlots(1);
      break;
     default:
      SilentMode = false;
      Puts("Invalid build option(s)");
      SetExitCode(3);
      return;
     }
   }
   else {
    ide.debug.target_override = ide.release.target_override = true;
    ide.debug.target = ide.release.target = mkf = arg[i];
   }
  if(clean)
   ide.Clean();
  if(exporting) {
   mkf = GetFullPath(mkf);
   Cout() << mkf << '\n';
   RealizeDirectory(mkf);
   if(makefile)
    ide.ExportMakefile(mkf);
   else
    ide.ExportProject(mkf, exporting == 2);
  }
  else
  if(makefile) {
   ide.SaveMakeFile(IsNull(mkf) ? "Makefile" : mkf, false);
   SetExitCode(0);
  }
  else
  if(ide.Build())
   SetExitCode(0);
  else
   SetExitCode(1);
 }
 else
  Puts("Usage: umk assembly main_package build_method -options [+flags] [output]\n"
       "Examples: umk examples Bombs GCC -ab +GUI,SHARED ~/bombs\n"
       "          umk examples:uppsrc Bombs ~/GCC.bm -rv +GUI,SHARED ~/bin\n"
```

```
        "See http://www.ultimatepp.org/app$ide$umk$en-us.html for details\n");
}
```