

Pensieve header: nb2tex for the OU project.

```
In[*]:= SetDirectory["C:\\drorbn\\AcademicPensieve\\Projects\\OU\\OU-221110-JKTR"];

Make["SomeComputations.tex", "SomeComputations.nb", Hold[
  nb2tex["SomeComputations", PDFFolder → "nb2tex_pdfs/SomeComputations", PDFWidth → 5.7]
]]

In[*]:= nb2tex["SomeComputations", PDFFolder → "nb2tex_pdfs/SomeComputations", PDFWidth → 5.7]
```

## Make

```
In[*]:= Make::usage =
  "Make[target, sources, Hold[action]] makes a target, or a list of targets, given
  sources, or a list of sources, in the style of the unix 'make' command.";
Make[target_String, sources_, action_Hold] :=
  Make[Evaluate@{target}, sources, action];
Make[target_, source_String, action__Hold] :=
  Make[target, Evaluate@{source}, action];
Make[target_List, sources_List, action_Hold] := Module[{},
  If[
    (And @@ ((FileType[#] != None) & /@ sources)) &&
    Or[
      Or @@ ((FileType[#] === None) & /@ targets),
      Min[AbsoluteTime[FileDate[#]] & /@ targets] <
      Max[AbsoluteTime[FileDate[#]] & /@ sources]
    ],
    Print["Making ", targets, " ..."];
    ReleaseHold[action]
  ]
];
```

## nb2tex

In as much as possible the program below should be identical to the one in pensieve://Projects/nb2tex/nb2tex.nb.

pdf

```
In[*]:= SetOptions[$FrontEndSession, PrintingStyleEnvironment → "Working"];
nb2tex[nb_String, opts__Rule] := nb2tex[nb, nb, opts];
```

pdf

In[ ]:=

```

nb2tex[nb_String, tex_String, opts___Rule] := Module[
  {notebook, PDFCounter = 0, lines, type, tag, pdfname, cell, c, cl,
    PDFFolder = PDFFolder /. {opts} /. PDFFolder → nb
  },
  nb2tex$PDFWidth = PDFWidth /. {opts} /. PDFWidth → 6.5;
  notebook = NotebookGet[NotebookOpen@FileNameJoin[{Directory[], nb <> ".nb"}]];
  If[FileType[PDFFolder] === None, CreateDirectory[PDFFolder]];
  DeleteFile /@ FileNames["*.pdf", PDFFolder];
  lines = Table[
    type = cell[[2]];
    tag = CellTags /. Cases[cell, _Rule] /. CellTags → "";
    Which[
      type == "Text" ^ tag == "tex", StringReplace[cell[[1]], {"'" → "'", "" → "\""}],
      StringMatchQ[tag, "pdf" ~~ ___], (
        pdfname = PDFFolder <> "/" <> ToString[++PDFCounter] <> ".pdf";
        Export[pdfname, Join[cell, Cell[PageWidth → 80 nb2tex$PDFWidth / 0.75]]];
        cl = "c:\\drorbn\\bin\\cpdf.exe -scale-page \"0.75 0.75\" " <>
          pdfname <> " -o " <> pdfname;
        Close@OpenRead["!" <> cl];
        StringReplace[
          "\\noindent\\nbpdfXXXType{pdfname}",
          {"XXX" → StringDrop[tag, 3], "Type" → type, "pdfname" → pdfname}
        ]
      ),
      type == "Text" ^ tag == "exec", ToExpression[cell[[1]]; "",
      True, ""
    ],
    {cell, Cases[notebook, c_Cell /; Length[c] ≥ 2 ^ FreeQ[c, _Cell, {1, ∞}], ∞]}
  ];
  lines = StringJoin @@ Riffle[DeleteCases[lines, ""], "\n\n"];
  WriteString[tex <> ".tex", lines]; Close[tex <> ".tex"]; ]

```