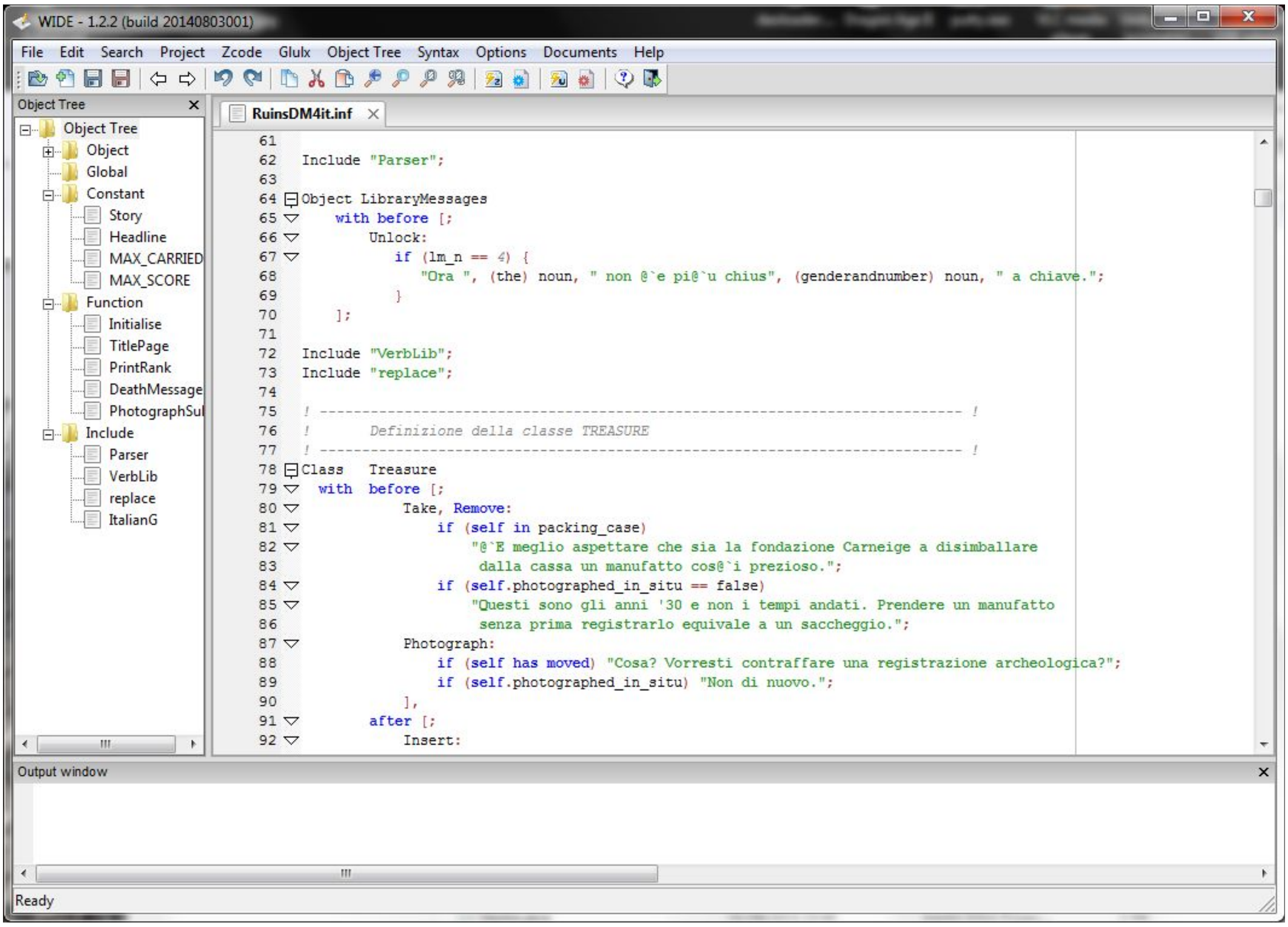
WIDE – WxWidgets IDE for Inform 6

https://github.com/silverslade/wide

WIDE is an Integrated Development Environment (IDE) for the Inform 6 Language.
Wide is a win32 Editor for Inform6: is an Integrated Development Environment (IDE), written entirely in C++/WxWidgets, for the creation of text adventures based on Graham Nelson's Inform6 Language. With Wide it's possible to edit, compile and run a text adventure in the z-code and glulx formats.



```

Include "Parser";

Object LibraryMessages
    with before [;
        Unlock:
           if (lm_n == 4) {
              "Ora ", (the) noun, " non @`e pi@`u chius", (genderandnumber) noun, " a chiave.";
           }
    ];

Include "VerbLib";
Include "replace";

! ---------------------------------------------------------------------------- !
!       Definizione della classe TREASURE
! ---------------------------------------------------------------------------- !
Class   Treasure
  with  before [;
            Take, Remove:
                if (self in packing_case)
                    "@`E meglio aspettare che sia la fondazione Carneige a disimballare
                     dalla cassa un manufatto cos@`i prezioso.";
                if (self.photographed_in_situ == false)
                    "Questi sono gli anni '30 e non i tempi andati. Prendere un manufatto
                     senza prima registrarlo equivale a un saccheggio.";
            Photograph:
                if (self has moved) "Cosa? Vorresti contraffare una registrazione archeologica?";
                if (self.photographed_in_situ) "Non di nuovo.";
            ],
        after [;
            Insert:
```

Main Features:

- Written in C++ (using the wxWidgets Library)
- Wide runs on Win32 / Linux systems
- Wide is localized in spanish, english, italian, german and french (simply customizing its wide.ini file…)
- Auto-completion window, in edit mode

- Wide is just an exe file: no DLL, no virtual machines. Wide is a stand alone application.
- Wrap lines, show number of lines, code folding
- Inform/Glulx support: wide supports zcode (5 and 8 version), ulx, blb, bres
- Objects tree and Syntax highlight is customizable
- Wide is portable: it can be started from an USB key/pen

Authors:

- Alessandro Schillaci
- Paolo Lucchesi

License:

Wide is an open source project (GNU GPL License). This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

Thanks to:

Urbatain, Christof Menear and Eric Forgeot for the wide translation in spanish, german and french.
Josè Manuel, for hints, suggestions and testing

Install on Linux:

Archlinux

If you’re a ArchLinux user, wide is on the AUR repository. Simply use this command:

```
yaourt -S wide
```

Other Linux distros install from source

1) install runtime wxgtk3.0 and git

Archlinux example:

```
sudo pacman -S wxgtk git
sudo pacman -S extra/wxgtk
sudo pacman -S gtk2hs-buildtools
```

Debian/Ubuntu/Mint example:

Install the following packages with Software Manager, Synaptic or from terminal with the ‘sudo apt-get install ’ command:

```
- git
- libwxbase3.0-0
- libwxbase-dev
- libwxgtk3.0-0
- libwxgtk3.0-dev
- wx-common
- wx3.0-headers
```

2) clone the wide repository

```
git clone https://github.com/silverslade/wide
```

You can use a git client (http://git-cola.github.io/)

3) compile wide

```
cd wide
cd src
make
```

Note:

- in case of errors uncomment line starting by “WX_LIBS” (see the makefile)
  you shouldn’t use “—as-needed” flag, use “—no-as-needed” instead

4) compiling wide under kdevelop (optional)

If you want to use kdevelop (with code completion) to compile wide, install it with

```
sudo pacman -S kdevelop
```

Start Kdevelop and open wide/Makefile as project

Install on Windows:

1) Install mingw and wxwidgets 3.0.1

- download and install mingw from here http://sourceforge.net/projects/mingw/files/latest/download
- download wxwidgets 3.0.1 or above from https://www.wxwidgets.org/downloads/
- unzip wxwidgets 3.0 (under c:/wxWidgets-3.0.1)
- compile it

```
cd c:/wxWidgets-3.0.1
cd build
cd msw
mingw32-make -f makefile.gcc SHARED=0 RUNTIME_LIBS=static MONOLITHIC=1 BUILD=release
UNICODE=1
```

(in case of errors rename C:\MinGW\msys in C:\MinGW\OLDmsys and recompile for a msys uncompatibilty)
after compiling wxwidgets, unrename C:\MinGW\OLDmsys to C:\MinGW\msys

In case of this error:

```
../../src/msw/treectrl.cpp: In member function 'virtual bool wxTreeCtrl::MSWOnNotify(int,
WXLPARAM, WXLPARAM*)':
../../src/msw/treectrl.cpp:3280:17: error: 'NMTVDISPINFOWW' was not declared in this
scope
```

edit line 2217 of /mingw/include/commctrl.h and change

```
<pre>#define TV_DISPINFO NMTVDISPINFO
```

instead of

```
<pre>#define TV_DISPINFO __AW(NMTVDISPINFO)
```

2) install GIT

install GIT as command line from http://git-scm.com/download/win
(or install github as client windows from https://windows.github.com/)

3) clone the wide repository

```
git clone https://github.com/silverslade/wide
```

4) edit the wide/makefile.gcc

edit the wide/makefile.gcc to fit your directories before compile it

5) Compile Wide under Windows

then you can compile wide under windows using this makefile and MinGW

```
cd wide
cd src
mingw32-make -f makefile.gcc
```

6) Install CodeLite (optional)

if you want to use a C++ IDE, you can install CodeLite (http://codelite.org/) and edit wide.project using your wxWidgets path and then open wide workspace with CodeLite.

Example:

```
<Compiler Options="--fno-exceptions -fno-pcc-struct-return -fstrict-aliasing -Wall
-D__WXMSW__ -D__GNUWIN32__ -D__WIN95__ -DUNICODE=1" C_Options="-g;-O0;-Wall" Assembler=""
Required="yes" PreCompiledHeader="" PCHInCommandLine="no" UseDifferentPCHFlags="no"
PCHFlags="">
        <IncludePath Value="."/>
        <IncludePath Value="C:/wxWidgets-3.0.1/contrib/include"/>
        <IncludePath Value="C:/wxWidgets-3.0.1/include"/>
        <IncludePath Value="C:/wxWidgets-3.0.1/lib/gcc_lib/mswu"/>
        <IncludePath Value="C:/wxWidgets-3.0.1/"/>
      </Compiler>
```

and

```
<SearchPaths>C:\wxWidgets-3.0.1\include\</SearchPaths>
```

CodeLite will build wide using the Makefile.gcc and it will let you use the code completion

In case of multiple installation of MinGW (i.e. MinGW and MinGW-4.8.1), you can force CodeLite to use a custom build system in this way:

- Open CodeLite
- settings → environment variables
- add this variable: PATH=C:\MinGW\bin;$(PATH)
- where C:\MinGW is the correct path to use in compiling

Do you want to Help?

We need help for:

- documentation
- manual
- test and documentation about installing WIDE on Linux/MacOsX Platforms
- C++ programming