

TAU4 - Tools And Utilities For the public

Devel's Manual

Autor: Dr. Franz Geiger

Adresse: DATEC Datentechnik GmbH, Schmiedgasse 7, A-6890 Lustenau

E-Mail: f.geiger@datec.at

Organisation: DATEC Datentechnik GmbH

Druckdatum: October 1, 2011

Copyright (C) 1998 - 2011, DATEC Datentechnik GmbH

Dieses Dokument ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdrucks und Vervielfältigung durch Kopieren oder Scannen sowie der Speicherung in Retrieval-Systemen des gesamten Dokumentes oder Teilen daraus, sind *DATEC Datentechnik GmbH* vorbehalten.

Kein Teil des Dokumentes darf ohne schriftliche Genehmigung von *DATEC Datentechnik GmbH* in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung, reproduziert oder unter Verwendung elektronischer Systeme gespeichert, verarbeitet, vervielfältigt oder verbreitet werden.

Die Weitergabe an Dritte ist nur mit ausdrücklicher Erlaubnis von *DATEC Datentechnik GmbH* gestattet.

Alle Marken und Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Titelhälter.

Contents

1	Document Revision History	7
2	License	8
3	What tau4 Is, And How To Install It	9
4	Packages	10
4.1	tau4automation	10
4.1.1	plc.py	10
4.2	tau4concurrency	13
4.3	tau4hw	14
4.3.1	tau4hw.baal	14
4.3.1.1	Names and Abbreviations	14
4.3.1.2	Synopsis	15
4.3.1.3	Example Showing How Backend-Variables Are Accessed .	15
4.3.1.4	Example Showing How Backend-Variables Are Def'ed . .	16
4.3.1.5	Modules	16
4.4	tau4misc	17
4.4.1	class Singleton	17
4.4.2	class Id	17
4.5	tau4patterns	18
4.6	tau4simulations	18
5	baal-Example Application	19
5.1	Goals	19
5.2	Main Program	20
5.3	Change Some Boilerplate Code	21
5.4	Define The baal-Variable	21
5.5	Add A Button For Sending Values And A Number Widget For Showing The Value	22
5.6	The Final Result	23
6	FAQ	31
6.1	FAQ tau4hw.baal	31
6.1.1	Program Changes Often Cause An <i>Er 50</i> On My SIGMATEK- CPU, Why?	31

Contents

7	Appendix	32
7.1	wxPython-Template For Thick Clients	32
7.2	License of tau4	38

List of Figures

5.1	Block <i>TestBlock1</i> on backend	19
5.2	Window as seen, if the template used for the <i>app4baal</i> main program ist run w/o any changes	20
5.3	<i>app4baal</i> 's main window	30

List of Tables

1.1	Revision History Versionsgeschichte	7
-----	--	---

1 Document Revision History

Table 1.1: Revision History || Versionsgeschichte

Date Datum	Changes	Änderungen
2001-10-02	Translation to English.	Alle Kapitel ins Englische übersetzt.
2001-10-01	1 st public ed.	Erstausgabe.

2 License

Lizenz

tau4 is released under the GPL3 license, see Chapter License of tau4.

tau4 ist unter der GPL Ver. 3 freigegeben, s. Kap. License of tau4.

3 What tau4 Is, And How To Install It

Was *tau4* ist und wie's installiert wird

tau4 is a collection of packages, which may shorten the development time of Python programs. See Chapter Modules for a list of packages currently available.

There's no installer for **tau4**, yet. So, simply unzip it and drop it into your app's root directory or copy it to **site-packages** (don't forget about a PTH file).

tau4 ist eine Sammlung von Packages, die die Entwicklungszeit von Python-Programmen verkürzen. Für eine Liste der derzeit verfügbaren Packages sei auf Kap. Packages verwiesen.

Es gibt noch keinen Installer für **tau4**. Solange können Sie es einfach unzippen und ins Root-Directory Ihrer Applikation kopieren oder nach **site-packages** (entsprechendes PTH-File nicht vergessen).

4 Packages

4.1 tau4automation

4.1.1 plc.py

This module contains the class *TDFSM* - a Finite State Machine, which is config'ed by a table.

Dieses Modul enthält nur die Klasse *TDFSM* - eine FSM, die über eine Tabelle definiert wird.

Example Application: Te following code

Anwendungsbeispiel: Folgender Code

Listing 4.1: Python Code

```
1
2 class MyFSM(TDFSM):
3
4     def IDLE_ea( self):
5         """Entry action for state IDLE. """
6         self.__time = time.time()
7         print ThisName( self) + "(): self.__time = %f; " % self.__time
8
9     def IDLE_sa( self):
10        """State action for state IDLE. """
11        print ThisName( self)
12
13    def IDLE_xa( self):
14        """Exit action for state IDLE. """
15        print ThisName( self) + "(): time.time() = %f; " % time.time()
16        print ThisName( self) + "(): Elapsed time = %f s; " % (time.time() - self.__time)
17
18    def IDLE_to_RUN_condition( self):
19        if time.time() - self.__time > 0.5:
20            return True
21        return False
22
23    def RUN_ea( self):
24        print ThisName( self)
25
26    def RUN_sa( self):
27        print ThisName( self)
28
29    def RUN_xa( self):
30        print ThisName( self)
```

4 Packages

```
31
32     rows = ((1, IDLE_ea, IDLE_sa, IDLE_xa, ((IDLE_to_RUN_condition, 2))),
33             (2, RUN_ea, RUN_sa, RUN_xa, None),
34             )
35
36 fsm = MyFSM( MyFSM.rows)
37 q = fsm.execute()
38 while not q:
39     q = fsm.execute()
40     time.sleep( 0.1)
41
42 return
```

leads to this output on a console:

führt zu folgender Ausgabe:

```
MyFSM::IDLE_ea(): self.__time = 1317403345.164493;
Execution time = 0.077963 ms;
MyFSM::IDLE_sa
Execution time = 0.057220 ms;
MyFSM::IDLE_sa
Execution time = 0.122070 ms;
MyFSM::IDLE_sa
Execution time = 0.127077 ms;
MyFSM::IDLE_sa
Execution time = 0.115156 ms;
MyFSM::IDLE_sa
Execution time = 0.108004 ms;
MyFSM::IDLE_sa
MyFSM::IDLE_xa(): self.__time = 1317403345.164493;
MyFSM::IDLE_xa(): Elapsed time = 0.502288 s;
MyFSM::RUN_ea
Execution time = 0.226021 ms;
MyFSM::RUN_sa
MyFSM::RUN_xa
Execution time = 0.128031 ms;
```

4 Packages

The definition of the FSM is rather simple Die Definition der FSM erfolgt also in ganz
this way: einfacher Weise:

```
#      State number
#      |  Entry action
#      | |          State action
#      | |          |          Exit action
#      | |          |          |          Exits
#      | |          |          |          |Condition for
#      | |          |          |          |exit to state 2
#      | |          |          |          ||
#      | |          |          |          ||          Exit state
#      | |          |          |          ||          |
rows = ((1, IDLE_ea, IDLE_sa, IDLE_xa, ((IDLE_to_RUN_condition, 2),)),
        (2, RUN_ea, RUN_sa, RUN_xa, None),
        # |
        # No exit, i.e. stop FSM-execution.
)
)
```

4.2 tau4concurrency

2DO: Needs to be
doc'ed!

4.3 tau4hw

This package provides modules for accessing hardware components.

Dieses Package ermöglicht den Zugriff auf Hardware.

4.3.1 tau4hw.baal

4.3.1.1 Names and Abbreviations

Namen und Abkürzungen

baal: Backend Access Abstraction Library.

baal: Backend Access Abstraction Library.

b_name - Backend Name: Name of the system, e.g. “main-system”, “default-system”, “mechanical-system”, etc.

b_name - Backend Name: Name des Systems wie bsw. “main-system”, “default-system”, “mechanical-system” usw.

b_type - Backend Type: At the time of writing these types are supported:

b_type - Backend Type: Zur Zeit werden folgende Typen unterstützt:

- sigmatek
- tcp-server
- none

- sigmatek
- tcp-server
- none

4.3.1.2 Synopsis

Übersicht

With *baal* it is possible to access variables living on a backend w/o knowing anything about the backend:

- PLC of *SIGMATEK GmbH & Co KG*, e.g. a *CCP 531*.
- PLC of *Beckhoff Automation GmbH*
- TCP-Server which may simulate a backend.

In *one* program arbitrarily many backends may be def'ed and accessed.

Mit *baal* ist es möglich, auf Variable eines Backends zuzugreifen, ohne wissen zu müssen, was für ein Backend es ist:

- Steuerung von *SIGMATEK GmbH & Co KG*, z.B. eine *CCP 531*.
- Steuerung von *Beckhoff Automation GmbH*
- TCP-Server, der bsw. eine Simulation realisiert.

Es sind beliebig viele verschiedene Backends in *einem* Programm ansprechbar.

4.3.1.3 Example Showing How Backend-Variables Are Accessed

Beispiel für Variablenzugriff

To read a variable *ClassSvr* of a block *Locker* on the backend *main-system*, we would write

```
value = baal.V[ "main-system.locker.class-svr" ].readR()
```

Um die Variable `ClassSvr` im Block `Locker` des Backends `main-system` zu lesen, schreibt man

To write on it, we would write the statement

```
baal.V[ "main-system.locker.class-svr" ].writeR( 12345)
```

Um auf die Variable zu schreiben, schreibt man

Please notice that an app using code this way doesn't need to know anything about the backend at hand!

Das Backend ist also bei diesen Aufrufen völlig transparent.

4.3.1.4 Example Showing How Backend-Variables Are Def'ed

Beispiel für die nötigen Definition, um auf Variable zugreifen zu können

To write to the variable *ClassSvr* of a block *Locker* on a backends *main-system* as shown in the last Chapter, these definitions are necessary:

Um die Variable `ClassSvr` im Block `Locker` des Backends `main-system` beschreiben zu können, wie es im letzten Kapitel gezeigt worden ist, sind folgende Definitionen nötig:

Listing 4.2: Python Code

```

1
2 ## Define the accessor (has to be done only once)
3 #
4 baal.X.Acc4Sigmatek( "main-system", ("TCP:150.10.10.1", None))
5
6 ## Define the variable
7 #
8 baal.V.Int32(\
9     "main-system.locker.class-svr",
10    "main-system/Locker/ClassSvr/svr"
11    0,
12    None,
13    None,
14    None,
15    "Class server of the main system's locker"
16    bal.V._NO_UPDATE
17    )

```

4.3.1.5 Modules

2DO: Needs to be doc'ed!

4.4 tau4misc

4.4.1 class Singleton

Threadsafe Singleton. May be moved into `tau4.tau4patterns` someday.

Application Example:

Threadsafe Singleton. Wird zukünftig vielleicht nach `tau4.tau4patterns` verlegt werden.

Anwendungsbeispiel:

Listing 4.3: Python Code

```

1
2 from tau4 import tau4misc
3
4 class MySingleton:
5
6     __metaclass__ = tau4misc.Singleton
7
8     def __init__( self):
9         """This ctor is called once and only once."""
10
11         # Your inits go here.
12         return
13
14     # All your methods go here.
```

4.4.2 class Id

Define names and ensure they are unique.

Application Example:

Stellt sicher, dass Namen, die eindeutig sein sollen, auch eindeutig sind.

Anwendungsbeispiel:

Listing 4.4: Python Code

```

1
2 from tau4 import tau4misc
3
4 my_unique_variable_name = tau4misc.Id( "My unique variable")
5 print my_unique_variable_name.as_str()
6
7 # The following statement raises eIdNotUniqueError, i.e. the
8 # print statement will not be exec'ed!
9 my_unique_variable_name = tau4misc.Id( "My unique variable")
10 print my_unique_variable_name.as_str()
```

4.5 tau4patterns

2DO: Needs to be doc'ed!

4.6 tau4simulations

2DO: Needs to be doc'ed!

5 baal-Example Application

5.1 Goals

We write a wxPython-app to show how to use *baal*. The app writes a setpoint value to a *SIGMATEK-CCP 531* and reads a current value back from it. We name this app *app4baal*.

app4baal needs a SIGMATEK-PLC as a backend to run properly. On this PLC a block *TestBlock1* has to be present as shown in Fig. 5.1. The app shall achieve the following:

Es soll nun eine wxPython-App geschrieben werden, die *baal* nützt, um Sollwerte an eine SIGMATEK-CCP 531 zu senden und Istwerte von dort zu lesen. Wir nennen diese App *app4baal*.

app4baal setzt als Backend eine SIGMATEK-Steuerung voraus, auf der ein Block *TestBlock1* läuft - s. Bild 5.1. Die App soll Folgendes leisten:

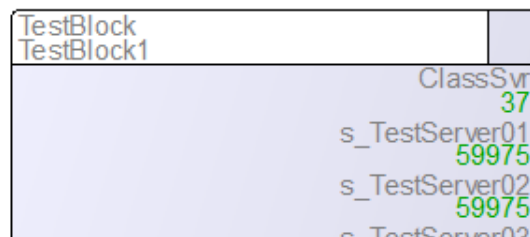


Figure 5.1: Block *TestBlock1* on backend

- If a button is pressed the value of *ClassSvr* of the block *TestBlock1* is read, incremented and written back to *ClassSvr*. The background task in turn increments the value again.
- Periodic read of the value of *ClassSvr* and display in a wxWindow.
- Auf Knopfdruck Lesen des Wertes von *ClassSvr* des Blockes und Schreiben des um 1 erhöhten Wertes auf *ClassSvr* des Blockes, der dann seinerseits den Wert um 1 erhöht.
- Periodisches Lesen des Wertes von *ClassSvr* und Anzeige in einem wxWindow.

5.2 Main Program

We will use the template shown in Chapter wxPython-Template For Thick Clients in the appendix. If the template is run w/o any modifications a gui is shown (Fig. 5.2).

Wir verwenden das Template aus dem Anhang wxPython-Template For Thick Clients als Vorlage für *app4baal.py*. Wenn man dieses Template startet, sieht man das GUI wie in Bild 5.2 gezeigt.

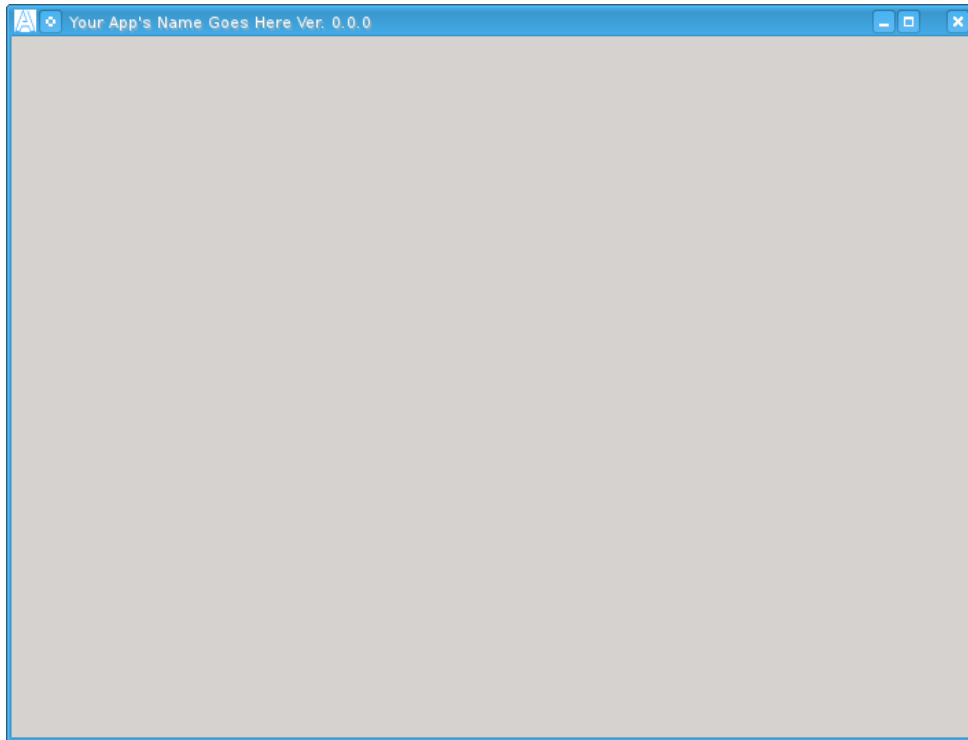


Figure 5.2: Window as seen, if the template used for the *app4baal* main program ist run w/o any changes

The two files *icon.ico* and *splash.png* need to be in the same directory as *app4baal.py*.

Die beiden Files *icon.ico* und *splash.png* müssen sich im gleichen Directory befinden wie *app4baal.py*.

Now, a couple of steps have to be exec'ed.

Nun haben wir eine Reihe von Schritten zu erledigen

5.3 Change Some Boilerplate Code

We substitute the app's name in the main program: Am Beginn des Hauptprogramms setzen wir den Namen unserer App ein:

Listing 5.1: Python Code

```
1
2 _APP_NAM = "app4baal"
```

5.4 Define The baal-Variable

Listing 5.2: Python Code

```
1
2 class MainFrame(wx.Frame):
3
4     def __init__( self, parent, id, caption):
5
6         this_name = u"MainFrame.__init__"
7                                     # The name of this method, just in case
8                                     # we needed it for logging.
9
10        ## Init the base class
11        wx.Frame.__init__( self, parent, id, caption, wx.DefaultPosition
12                           , size=(700, 500)
13                           #, style=wx.MAXIMIZE
14                           )
15
16        ## Define the baal-accessor and the baal-variable
17        x = baal.X.Acc4Sigmatek( "ccp531", ( "TCP:10.10.150.1", None))
18        if not x.is_connected():
19            x.connect( baal.ConnectorReconnectingConnector())
20
21        baal.V.Int32(\
22            "app4baal-class-svr",
23            "ccp531/App4BAAL/ClassSvr/svr"
24            , 0, None, None,
25            "", "",
26            baal.V._MID_SPEED_UPDATE
27        )
28
29        ## Build this MainFrame
30        self._build_()
31
32        self.Maximize()
33
34        self.Layout()
35
36        ## Activate all events we will need
```

5 baal-Example Application

```
37 wx.EVT_IDLE( self, self._on_EVT_IDLE_)
38 # ^ ^ ^
39 # | | |
40 # | | Event target's event handler
41 # | Event target
42 # Event
43
44 # Activate IDLE events. These are called,
45 # if the GUI is - well - IDLE. You may
46 # then do things you wouldn't want to do
47 # if the GUI is processing.
48 wx.EVT_CLOSE( self, self._on_close_window_)
49 # Activate CLOSE event. You may want to
50 # do some cleanup in case we are
51 # closing down.
52
53 ## Activate the IDLE-timer
54 self._idle_timer = wx.Timer( self)
55 self.Bind( wx.EVT_TIMER, self._on_idle_timer_)
56 self._idle_timer.Start( 100)
57
58 ## Some attributes
59 self._is_closing = False
60
61 return
```

5.5 Add A Button For Sending Values And A Number Widget For Showing The Value

We define a button and a number widget and add it to the layout.

Wir definieren den Button und das Number-Widget und fügen die beiden gleich zum Layout hinzu.

Listing 5.3: Python Code

```
1
2 def _build_( self):
3     """Builds the MainFrame, i.e. creates all widgets and puts them into sizers."""
4
5     ## Put the App's icon in the MainFrame
6     if os.path.exists( "icon.ico"):
7         icon = wx.EmptyIcon()
8         bmp = wx.Image( "icon.ico").ConvertToBitmap()
9         icon.CopyFromBitmap( bmp)
10        self.SetIcon( icon)
11
12    ## Build the panel and all its widgets
13    self._build_main_panel_()
14
15    return
```

5 baal-Example Application

```
16
17 def _build_main_panel_( self):
18     """Builds the main panel, i.e. creates all widgets and puts them into sizers."""
19
20     ## Create the panel
21     panel = wx.Panel( self, -1)
22
23     ## Create the button and bind the handler
24     self.__w_sender_button = wx.Button( panel, -1, "Send current value to backend", size=(
25     self.__w_sender_button.Bind( wx.EVT_BUTTON, self._on_EVT_BUTTON__SEND_)
26
27     ## Create the number widget and connect it with the baal-variable
28     self.__w_value = tau4wx.widgets.View4Number( panel, -1, (100,-1))
29     self.__mvc_value = baal.adapter4gui.MVC4BV( baal.V[ "app4baal-class-svr"])
30     self.__mvc_value.view_connect( self.__w_value)
31
32     ## Layout all the widgets
33     s0 = wx.BoxSizer( wx.HORIZONTAL)
34     s = s0
35
36     s.AddStretchSpacer( 1)
37     s.Add( self.__w_sender_button, 1, wx.ALIGN_CENTER|wx.ALL, _PADDING)
38     s.AddStretchSpacer( 1)
39     s.Add( self.__w_value, 1, wx.ALIGN_CENTER|wx.ALL, _PADDING)
40     s.AddStretchSpacer( 1)
41
42     panel.SetSizer( s0)
43     s0.Fit( panel)
44     return panel
```

The button's event handler is rather short: Der Event-Handler für den Button ist nicht sehr lang:

Listing 5.4: Python Code

```
1
2 def _on_EVT_BUTTON__SEND_( self, wxE):
3     baal.V["app4baal-class-svr"].writeR( baal.V["app4baal-class-svr"].readL())
```

5.6 The Final Result

Fig. 5.3 shows the final result. The program that does all this is listed below.

Das Ergebnis ist dann das folgende Listing. Wie sich das Programm auf dem Bildschirm präsentiert, zeigt Bild 5.3.

```
1 # -*- coding: utf8 -*-
2 #
3 #
4 # Copyright (C) by DATEC Datentechnik GmbH, A-6890 LUSTENAU, 1998 - 2011
```

5 baal-Example Application

```
5 #
6 # This file is part of tau4.
7 #
8 # tau4 is free software: you can redistribute it and/or modify
9 # it under the terms of the GNU General Public License as published by
10 # the Free Software Foundation, either version 3 of the License, or
11 # (at your option) any later version.
12 #
13 # tau4 is distributed in the hope that it will be useful,
14 # but WITHOUT ANY WARRANTY; without even the implied warranty of
15 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
16 # GNU General Public License for more details.
17 #
18 # You should have received a copy of the GNU General Public License
19 # along with tau4. If not, see <http://www.gnu.org/licenses/>.
20
21
22 import logging; from logging import handlers
23
24 import os
25 import string
26 import sys
27 from tau4.tau4hw import baal
28 from tau4 import tau4wx
29 import time
30 import wx
31
32
33 _PADDING = 5
34
35 _APP_NAME = u"app4baal"
36 _APP_VERSION_NUMBERS = (0, 1, 0)
37 _APP_VERSION = u"%d.%d.%d" % _APP_VERSION_NUMBERS
38 _APP_CAPTION = u"%s Ver. %s" % (_APP_NAME, _APP_VERSION)
39
40 _USE_CUSTOM_EXCEPT_HOOK = 0
41
42                                     # Be aware, that using your own excpet hook is
43                                     # not convenient during development. Activate
44                                     # in releases only.
45
46 _LOGGING_LEVEL = (\
47     logging.DEBUG,
48     logging.INFO,
49     logging.WARNING,
50     logging.ERROR,
51     logging.CRITICAL,
52 ) [0]
53
54 if _USE_CUSTOM_EXCEPT_HOOK:
55     import traceback
56
57     def Excepthook( typ, val, tbk):
58         tbk_formatted = ""
```

5 baal-Example Application

```
59     for file, line, module, info in traceback.extract_tb( tbk):
60         tbk_formatted += "\tFile:\t%s\n\tLine:\t%d\n\tModule:\t%s\n\tInfo:\t%s\n\n" %
61
62     if tbk_formatted:
63         tbk_formatted = tbk_formatted[:-1]
64
65     additional_infos = "\tCurrent working directory = %s\n\tsys.argv = %s" % (os.getcwd(),
66     logging.critical( "An uncaught exception has occurred:\ntype = '%s'\nvalue = '%s'\n"
67                       % (typ, val, tbk_formatted, additional_infos)
68                       )
69     sys.__excepthook__( typ, val, tbk)
70     return
71
72     sys.excepthook = Excepthook
73
74
75 def _InitLogging_():
76     logger = logging.getLogger()
77     hdlr = handlers.RotatingFileHandler( "./%s.log" % _APP_NAME, "a", 10000000, 10)
78     formatter = logging.Formatter( "%(asctime)s %(process)d %(thread)d %(levelname)10s %(message)s")
79     hdlr.setFormatter( formatter)
80     logger.addHandler( hdlr)
81     logger.setLevel( _LOGGING_LEVEL)
82     return
83
84 _Logger = logging.getLogger()
85
86
87 class MainFrame(wx.Frame):
88
89     def __init__( self, parent, id, caption):
90
91         this_name = u"MainFrame.__init__"
92                                     # The name of this method, just in case
93                                     # we needed it for logging.
94
95         ## Init the base class
96         wx.Frame.__init__( self, parent, id, caption, wx.DefaultPosition
97                             , size=(700, 500)
98                             #, style=wx.MAXIMIZE
99                             )
100
101         ## Define the baal-accessor and the baal-variable
102         x = baal.X.Acc4Sigmatek( "ccp531", ( "TCP:10.10.150.1", None))
103         if not x.is_connected():
104             x.connect( baal.ConnectorReconnectingConnector())
105
106         baal.V.Int32(\
107             "app4baal-class-svr",
108             "ccp531/TestBlock1/ClassSvr/svr"
109             , 0, None, None,
110             "", "",
111             baal.V._MID_SPEED_UPDATE
112         )
```

5 baal-Example Application

```
113     ## Build this MainFrame
114     self._build_()
115                                     # Build the MainFrame
116     self.Maximize()
117                                     # Let the MainFrame fill the whole screen
118     self.Layout()
119                                     # 2D0: Is this still needed?
120
121     ## Activate all events we will need
122     wx.EVT_IDLE( self, self._on_EVT_IDLE_)
123     # ^           ^           ^
124     # |           |           |
125     # |           |           Event target's event handler
126     # |           Event target
127     # Event
128                                     # Activate IDLE events. These are called,
129                                     # if the GUI is - well - IDLE. You may
130                                     # then do things you wouldn't want to do
131                                     # if the GUI is processing.
132     wx.EVT_CLOSE( self, self._on_close_window_)
133                                     # Activate CLOSE event. You may want to
134                                     # do some cleanup in case we are
135                                     # closing down.
136
137     ## Activate the IDLE-timer
138     self._idle_timer = wx.Timer( self)
139     self.Bind( wx.EVT_TIMER, self._on_idle_timer_)
140     self._idle_timer.Start( 100)
141
142     ## Some attributes
143     self._is_closing = False
144
145     return
146
147 def _build_( self):
148     """Builds the MainFrame, i.e. creates all widgets and puts them into sizers."""
149
150     ## Put the App's icon in the MainFrame
151     if os.path.exists( "icon.ico"):
152         icon = wx.EmptyIcon()
153         bmp = wx.Image( "icon.ico").ConvertToBitmap()
154         icon.CopyFromBitmap( bmp)
155         self.SetIcon( icon)
156
157     ## Build the panel and all its widgets
158     self._build_main_panel_()
159
160     return
161
162 def _build_main_panel_( self):
163     """Builds the main panel, i.e. creates all widgets and puts them into sizers."""
164
165     ## Create the panel
166     panel = wx.Panel( self, -1)
```

5 baal-Example Application

```
167
168     ## Create the button and bind the handler
169     self.__w_sender_button = wx.Button( panel, -1, "Send current value to backend", si
170     self.__w_sender_button.Bind( wx.EVT_BUTTON, self._on_EVT_BUTTON__SEND_)
171
172     ## Create the number widget and connect it with the baal-variable
173     self.__w_value = tau4wx.widgets.View4Number( panel, -1, (100,-1))
174     self.__mvc_value = baal.adapter4gui.MVC4BV( baal.V[ "app4baal-class-svr" ])
175     self.__mvc_value.view_connect( self.__w_value)
176
177     ## Layout all the widgets
178     s0 = wx.BoxSizer( wx.HORIZONTAL)
179     s = s0
180
181     s.AddStretchSpacer( 1)
182     s.Add( self.__w_sender_button, 1, wx.ALIGN_CENTER|wx.ALL, _PADDING)
183     s.AddStretchSpacer( 1)
184     s.Add( self.__w_value, 1, wx.ALIGN_CENTER|wx.ALL, _PADDING)
185     s.AddStretchSpacer( 1)
186
187     panel.SetSizer( s0)
188     s0.Fit( panel)
189     return panel
190
191 def _on_close_window_( self, event):
192     self._is_closing = True
193     busy = wx.BusyInfo( "Just a moment, please...")
194     wx.SafeYield( None, True)
195     time.sleep( 1.0)
196
197     self.Destroy()
198
199 def _on_EVT_BUTTON__SEND_( self, wxE):
200     print baal.V["app4baal-class-svr"].readL()
201     baal.V["app4baal-class-svr"].writeR( baal.V["app4baal-class-svr"].readL() + 1)
202
203 def _on_EVT_IDLE_( self, event):
204     this_name = "MainFrame::_on_EVT_IDLE_"
205
206     event.Skip()
207     return
208
209 def _on_idle_timer_( self, event):
210     """Ensures that idle events get handled.
211
212     NOTE:
213         To do it this way is more efficient than a call to event.RequestMore() in each
214         event handler.
215
216         Every idle event handler should call event.Skip() now.
217     """
218     if not self._is_closing:
219         wx.WakeUpIdle()
220
```

5 baal-Example Application

```
221         return
222
223     def _on_timer_( self, event):
224         if self._is_closing:
225             return
226
227         return
228
229
230
231 class _SplashScreen(wx.SplashScreen):
232
233     def __init__( self, main_frame):
234         if os.path.exists( "splash.png"):
235             bmp = wx.Image( "splash.png").ConvertToBitmap()
236         else:
237             bmp = wx.EmptyBitmap( 500, 350)
238
239         bmp = wx.Image( "splash.png").ConvertToBitmap()
240         wx.SplashScreen.__init__( self, bmp
241                                   , wx.SPLASH_CENTRE_ON_SCREEN | wx.SPLASH_TIMEOUT
242                                   , 5000
243                                   , None, -1
244                                   , size=(800,-1)
245                                   )
246         self._main_frame = main_frame
247         self.Bind( wx.EVT_CLOSE, self.OnClose)
248         self._fc = wx.FutureCall( 2000, self._show_main_)
249         return
250
251     def OnClose(self, evt):
252         """Override; see wxPython dox."""
253
254         ## Make sure the default handler runs too so this window gets destroyed
255         evt.Skip()
256         self.Hide()
257
258         ## If the timer is still running then go ahead and show the MainFrame now
259         if self._fc.IsRunning():
260             self._fc.Stop()
261             self._show_main_()
262
263         return
264
265     def _show_main_(self):
266         self._main_frame.Centre( wx.BOTH)
267         self.Show( False)
268         self._main_frame.Show( True)
269         return
270
271
272 class App(wx.App):
273
274     def OnInit(self):
```

5 baal-Example Application

```
275     """Override; see wxPython dox."""
276
277     main_frame = MainFrame( None, -1, _APP_CAPTION)
278     splash = _SplashScreen( main_frame)
279     splash.Show()
280     self.SetTopWindow( main_frame)
281     return True
282
283
284
285 def main():
286
287     ## First we change the path to where the executable resides
288     if sys.platform.startswith( "win"):
289         try:
290             dirname = os.path.split(sys.argv[0])[0]
291             os.chdir( dirname)
292         except ValueError:
293             pass
294
295     ## Now we start logging
296     _InitLogging_()
297     _Logger.critical( u"***** %s has started. *****" % _APP_NAME)
298     _Logger.info( "Program has started with these args: '%s'." % sys.argv)
299
300     ## And finally we start the app
301     redirect_stderr_and_stdout = 0
302     app = App( redirect_stderr_and_stdout)
303     app.MainLoop()
304
305
306 if __name__ == '__main__':
307     main()
```

5 *baal-Example Application*

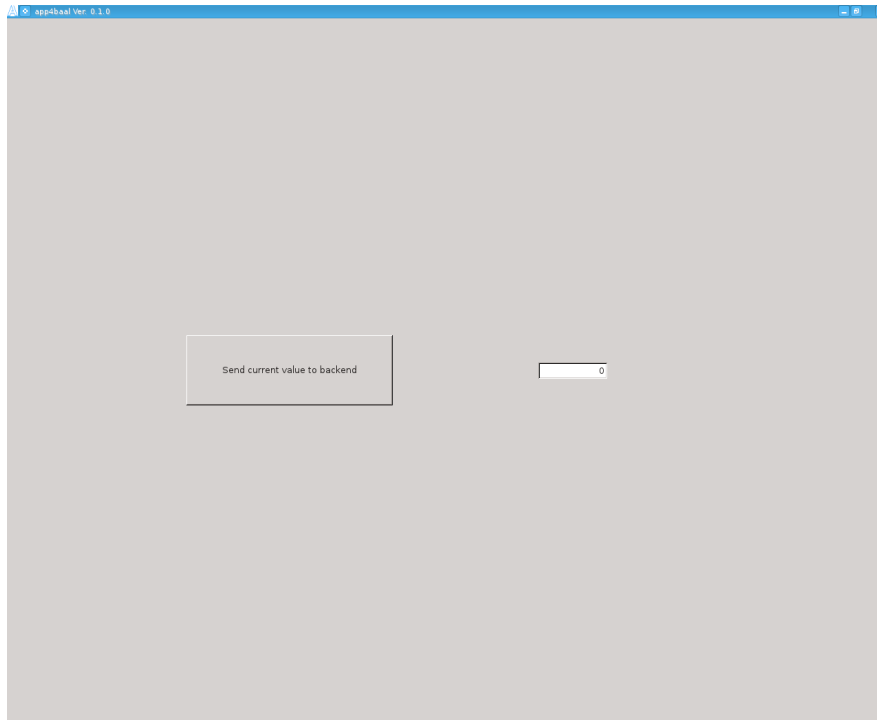


Figure 5.3: app4baal's main window

6 FAQ

6.1 FAQ tau4hw.baal

6.1.1 Program Changes Often Cause An Er 50 On My SIGMATEK-CPU, Why?

This error shows up if you change your backend program while the gui is running. The gui buffers the variables' object addresses for much faster access to them. If the backend changes, the gui will not notice this and continue to access objects now lying somewhere else in the memory.

Woher kommen die *Er 50* auf der SIGMATEK-CPU nach Programmänderungen?

Dieser Fehler kann auftreten, wenn Sie Programmänderungen am Backend durchführen während die Visu noch läuft. Die Objektadressen auf der Visu werden gepuffert (das beschleunigt die Zugriffe um den Faktor 5), ändern sich aber bei Änderungen auf dem Backend. Die Visu greift dann noch auf die alten Adressen zu, was zu **Er 50** führen kann.

7 Appendix

7.1 wxPython-Template For Thick Clients

```
1 # -*- coding: utf8 -*-
2
3 #
4 #   Copyright (C) by DATEC Datentechnik GmbH, A-6890 LUSTENAU, 1998 - 2011
5 #
6 #   This file is part of tau4.
7 #
8 #   tau4 is free software: you can redistribute it and/or modify
9 #   it under the terms of the GNU General Public License as published by
10 #   the Free Software Foundation, either version 3 of the License, or
11 #   (at your option) any later version.
12 #
13 #   tau4 is distributed in the hope that it will be useful,
14 #   but WITHOUT ANY WARRANTY; without even the implied warranty of
15 #   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
16 #   GNU General Public License for more details.
17 #
18 #   You should have received a copy of the GNU General Public License
19 #   along with tau4. If not, see <http://www.gnu.org/licenses/>.
20
21 import logging; from logging import handlers
22
23 import os
24 import string
25 import sys
26
27 from tau import tauD, tauWX
28 import traceback
29 import time
30 import wx
31
32
33 _PADDING = 5
34
35
36 _APP_NAME = u"Your App's Name Goes Here"
37 _APP_VERSION_NUMBERS = (0, 0, 0)
38 _APP_VERSION = u"%d.%d.%d" % _APP_VERSION_NUMBERS
39 _APP_CAPTION = u"%s Ver. %s" % (_APP_NAME, _APP_VERSION)
40
41
42
```

7 Appendix

```
43 _USE_CUSTOM_EXCEPT_HOOK = 0
44                                     # Be aware, that using your own excpet hook is
45                                     #   not convenient during development. Activate
46                                     #   in releases only.
47
48 _LOGGING_LEVEL = (\
49     logging.DEBUG,
50     logging.INFO,
51     logging.WARNING,
52     logging.ERROR,
53     logging.CRITICAL,
54 ) [0]
55
56
57
58 if _USE_CUSTOM_EXCEPT_HOOK:
59     import traceback
60
61     def Excepthook( typ, val, tbk):
62         tbk_formatted = ""
63         for file, line, module, info in traceback.extract_tb( tbk):
64             tbk_formatted += "\tFile:\t%s\n\tLine:\t%d\n\tModule:\t%s\n\tInfo:\t%s\n\n" %
65
66         if tbk_formatted:
67             tbk_formatted = tbk_formatted[:-1]
68
69         additional_infos = "\tCurrent working directory = %s\n\tsys.argv = %s" % (os.getcwd
70         logging.critical( "An uncaught exception has occurred:\ntype = '%s'\nvalue = '%s'\n"
71                             % (typ, val, tbk_formatted, additional_infos)
72                             )
73         sys.__excepthook__( typ, val, tbk)
74         return
75
76     sys.excepthook = Excepthook
77
78
79
80 def _InitLogging_():
81     logger = logging.getLogger()
82     hdlr = handlers.RotatingFileHandler( "./%s.log" % _APP_NAME, "a", 1000000, 10)
83     formatter = logging.Formatter( "%(asctime)s %(process)d %(thread)d %(levelname)10s %(m
84     hdlr.setFormatter( formatter)
85     logger.addHandler( hdlr)
86     logger.setLevel( _LOGGING_LEVEL)
87     return
88
89 _Logger = logging.getLogger()
90
91
92 class MainFrame(wx.Frame):
93
94     def __init__( self, parent, id, caption):
95
96         this_name = u"MainFrame.__init__"
```

7 Appendix

```
97         # The name of this method, just in case
98         # we needed it for logging.
99     ## Init the base class
100    wx.Frame.__init__( self, parent, id, caption, wx.DefaultPosition
101                      , size=(700, 500)
102                      #, style=wx.MAXIMIZE
103                      )
104
105    ## Build this MainFrame
106    self._build_()
107
108    self.Maximize()
109
110    self.Layout()
111
112    ## Activate all events we will need
113    wx.EVT_IDLE( self, self._on_idle_ )
114    # ^ ^
115    # | |
116    # | | Event target's event handler
117    # | | Event target
118    # Event
119
120    # Activate IDLE events. These are called,
121    # if the GUI is - well - IDLE. You may
122    # then do things you wouldn't want to do
123    # if the GUI is processing.
124    wx.EVT_CLOSE( self, self._on_close_window_ )
125    # Activate CLOSE event. You may want to
126    # do some cleanup in case we are
127    # closing down.
128
129    ## Activate the IDLE-timer
130    self._idle_timer = wx.Timer( self )
131    self.Bind( wx.EVT_TIMER, self._on_idle_timer_ )
132    self._idle_timer.Start( 100 )
133
134    ## Some attributes
135    self._is_closing = False
136
137    return
138
139    def _build_( self ):
140        """Builds the MainFrame, i.e. creates all widgets and puts them into sizers."""
141
142        ## Put the App's icon in the MainFrame
143        if os.path.exists( "icon.ico" ):
144            icon = wx.EmptyIcon()
145            bmp = wx.Image( "icon.ico" ).ConvertToBitmap()
146            icon.CopyFromBitmap( bmp )
147            self.SetIcon( icon )
148
149        ## Create all the other widgets
150        pass
```

7 Appendix

```
151
152     return
153
154     def _on_close_window_( self, event):
155
156         self._is_closing = True
157         busy = wx.BusyInfo( "Just a moment, please...")
158         wx.SafeYield( None, True)
159         time.sleep( 1.0)
160
161         self.Destroy()
162
163     def _on_idle_( self, event):
164         this_name = "MainFrame::_on_idle_"
165
166         event.Skip()
167         return
168
169     def _on_idle_timer_( self, event):
170         """Ensures that idle events get handled.
171
172         .. note::
173             To do it this way is more efficient than a call to event.RequestMore() in each
174             event handler.
175
176             Every idle event handler should call event.Skip() now.
177         """
178         if not self._is_closing:
179             wx.WakeUpIdle()
180
181         return
182
183     def _on_timer_( self, event):
184         if self._is_closing:
185             return
186
187         return
188
189
190
191     class _SplashScreen(wx.SplashScreen):
192
193         def __init__( self, main_frame):
194             if os.path.exists( "splash.png"):
195                 bmp = wx.Image( "splash.png").ConvertToBitmap()
196             else:
197                 bmp = wx.EmptyBitmap( 500, 350)
198
199             bmp = wx.Image( "splash.png").ConvertToBitmap()
200             wx.SplashScreen.__init__( self, bmp
201                                     , wx.SPLASH_CENTRE_ON_SCREEN | wx.SPLASH_TIMEOUT
202                                     , 5000
203                                     , None, -1
204                                     , size=(800, -1)
```

7 Appendix

```
205         )
206         self._main_frame = main_frame
207         self.Bind( wx.EVT_CLOSE, self.OnClose)
208         self._fc = wx.FutureCall( 2000, self._show_main_)
209         return
210
211     def OnClose(self, evt):
212
213         ## Make sure the default handler runs too so this window gets destroyed
214         evt.Skip()
215         self.Hide()
216
217         ## If the timer is still running then go ahead and show the MainFrame now
218         if self._fc.IsRunning():
219             self._fc.Stop()
220             self._show_main_()
221
222         return
223
224     def _show_main_(self):
225         self._main_frame.Centre( wx.BOTH)
226         self.Show( False)
227         self._main_frame.Show( True)
228         #if self._fc.IsRunning():
229         #    self.Raise()
230         # ### Guess, this isn't needed, is it?
231
232         return
233
234
235     class App(wx.App):
236
237         def OnInit(self):
238             main_frame = MainFrame( None, -1, _APP_CAPTION)
239             splash = _SplashScreen( main_frame)
240             splash.Show()
241             self.SetTopWindow( main_frame)
242             return True
243
244
245
246     def main():
247
248         ## First we change the path to where the executable resides
249         if sys.platform.startswith( "win"):
250             try:
251                 dirname = os.path.split(sys.argv[0])[0]
252                 os.chdir( dirname)
253             except ValueError:
254                 pass
255
256         ## Now we start logging
257         _InitLogging_()
258         _Logger.critical( u"***** %s has started. *****" % _APP_NAME)
```

7 Appendix

```
259     _Logger.info( "Program has started with these args: '%s'. " % sys.argv)
260
261     ## And finally we start the app
262     redirect_stderr_and_stdout = 0
263     app = App( redirect_stderr_and_stdout)
264     app.MainLoop()
265
266
267 if __name__ == '__main__':
268     main()
```

7.2 License of tau4

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights. Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received. You must make sure that they, too, receive
or can get the source code. And you must show them these terms so they
know their rights.

Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software. For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

Some devices are designed to deny users access to install or run
modified versions of the software inside them, although the manufacturer
can do so. This is fundamentally incompatible with the aim of
protecting users' freedom to change the software. The systematic
pattern of such abuse occurs in the area of products for individuals to
use, which is precisely where it is most unacceptable. Therefore, we
have designed this version of the GPL to prohibit the practice for those
products. If such problems arise substantially in other domains, we
stand ready to extend this provision to those domains in future versions
of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents.
States should not allow patents to restrict development and use of
software on general-purpose computers, but in those that do, we wish to
avoid the special danger that patents applied to a free program could
make it effectively proprietary. To prevent this, the GPL assures that
patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and
modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of
works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this
License. Each licensee is addressed as "you". "Licensees" and

7 Appendix

"recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains

7 Appendix

in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License,

7 Appendix

in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and

7 Appendix

adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and

7 Appendix

finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a

7 Appendix

publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any

7 Appendix

author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
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 3 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.
```

```
You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program

7 Appendix

into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.