

ogMake



Make your Notarized builds a play

User's guide

Component's user data sheet

Release	Version	Date	Rédacteur	Commentaires
Initial		10 mars 2023	OG	First version
r1		13 avril 2023	OG	Some corrections



from v19 R6, for project mode

Protée
IT, app design

Protée sarl
31, rue Sainte-Marthe
31000 TOULOUSE

Tél : 0970 46 56 46
Mob : 06 3718 5941
www.protee.org / info@protee.org



Overview	3
Introduction	3
Preparation	4
Description	4
Configuration	4
ogMake forms	5
Methods	7
Calls	11
Register & Configure	11
Build by methods	11
Make by 4DPop / ogPop	12
Thanks	12



Overview

Introduction

The **ogMake** component helps the developer in its daily routine with they 4D products, as the goal was to generate the build chain in two clicks: one to invoke it, one on the ok button! It makes and builds your component or database in a snap. It works after a full analyse of the host "build file" (build folder name, options for CS, mono, component...). It memorise for all your databases the options you choose in the UI too.

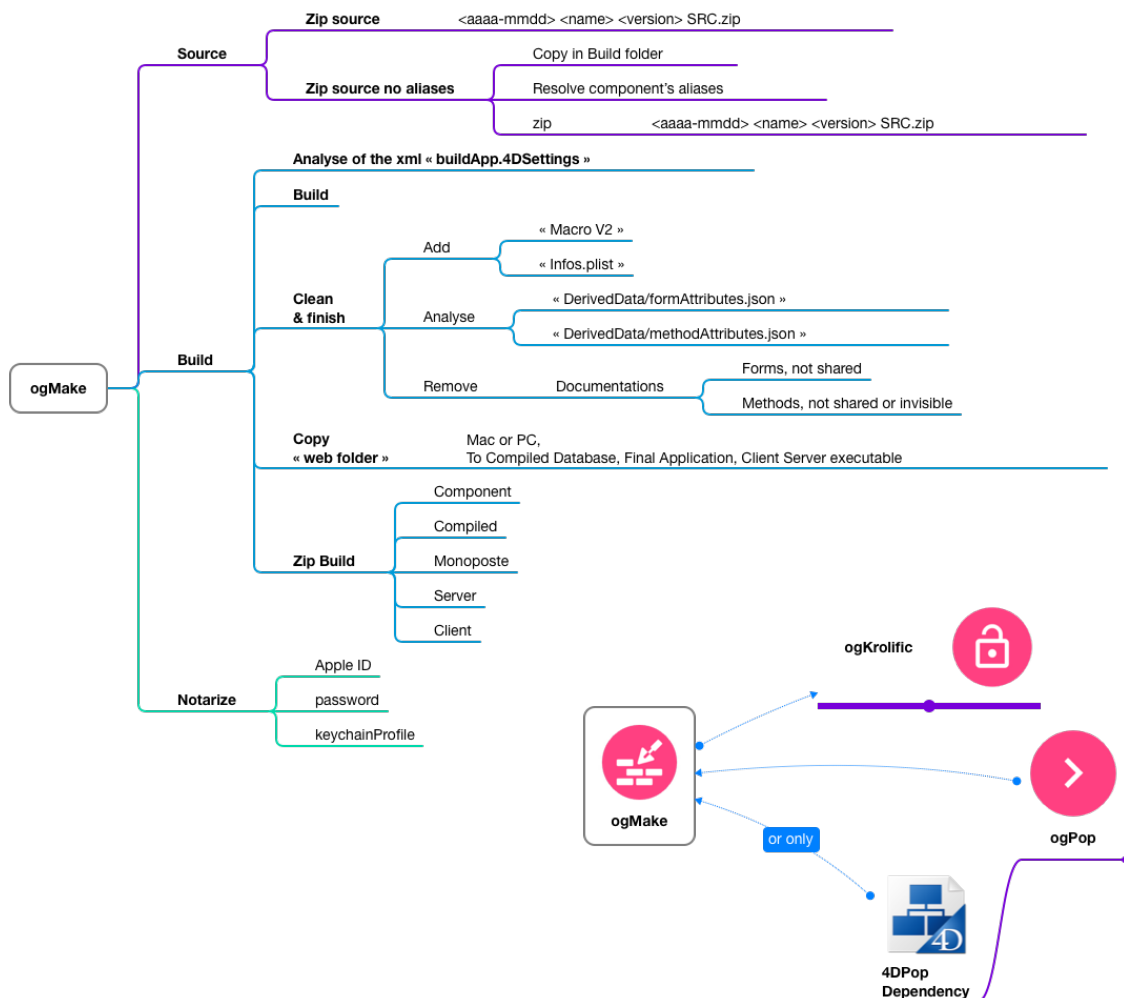
Very easy to use, especially for components, where it is a must! For databases, it works too for sure, and you can use Oviyam that does the same stuff.

Dependencies

To get **ogMake** works, you need to have it in your component's folder, and **ogKrolific** (Protée's license manager) installed too (aliases work). Don't forget to exclude them from your build's settings if you don't need them in the compiled target. Use it with 4DPop or best with **ogPop** (but you need **4DPop** installed for both).

What it does

All is made with the analyse of the xml « buildApp.4DSettings ». Depending on it, ogMake knows the build folder and its name, and the options you can select or not.





Preparation

To get **ogMake** works for notarizing, you must prepare your platform following what is explained in Oviyam documentation « Oviyam Notarizing.pdf » for [altool](#) and [notarytool](#) in your keychain.

Description

- SOURCE
 - zip your source project database folder, adding date and version to the name, and "SRC". ie ogColours will give
 - **2023-0112 ogColours 2.2.00 SRC.zip** => easy for version archive
 - Copy your source project database folder to the build folder, then resolve for the components all the aliases found, then zip it adding date and version to the name, and "SRC". ie ogColours will give
 - **2023-0112 ogColours 2.2.00 SRC.zip** => easy for version archive
- BUILD
 - Eventual copy from SettingsMac and SettingsWindows to Settings if « buildApp.4DSettings » found in it
 - Analyse of the xml « buildApp.4DSettings »
 - Make the build
 - Clean & Finish
 - Add into the build:
 - info.plist and macro v2 if exist
 - eventually the web folder if addressed, into the build file
 - Remove in the build:
 - After the analyse of « DerivedData/formAttributes.json »
 - all documentation's files for non shared forms
 - After the analyse of « DerivedData/methodAttributes.json »
 - all documentation's files for invisible methods, or not shared.
 - launch a method in your host if given, to let you do additional stuff
 - Copy the « web folder » if defined
 - Into all the possible targets: Compiled, Final Application, Client Server Executable/Server
 - ZIP
 - zip the build's file(s), adding date to the name. ie ogColours will give « **2023-0112 ogColours 2.2.00.zip** »
- NOTARIZE
 - notarize for all the build's file(s), with signing, ditto (dmg), notarytool, and staple.
- Produce a sound when finished !

Configuration

The target is always your host database, but for the Quick Notary.

What needs **ogMake** to properly name generated files, is a product name and a version string, optionally a settings_file to indicate the build.settings file.

There is two ways to give them

- by configuring of **ogMake** on startup
- giving to the build command an object with « t_name » and « t_version » properties.

If you attend to use it with **ogPop**, you have no other way than configure **ogMake**, to make it knows the values.



ogMake forms

Four interfaces and methods/prefs files exist:

- Database: for full options (component, monoposte, Server, Client).
- Component: for component (reduced options)
- How Do I: to get a copy of your sources, then manage the aliases of the components of this host (delete or resolve), and notarize
- Quick Notary: to do the notarize on a folder you drop on the interface

All the options are enabled/disabled regarding the options found in the xml build file.

- To change the values for Sign/dmg/notarize/staple, just double-click on the desired field.
- The icon displayed is taken from resources folder/pictures/logo_product.png if any. Otherwise, the standard ogMake icon is used instead.

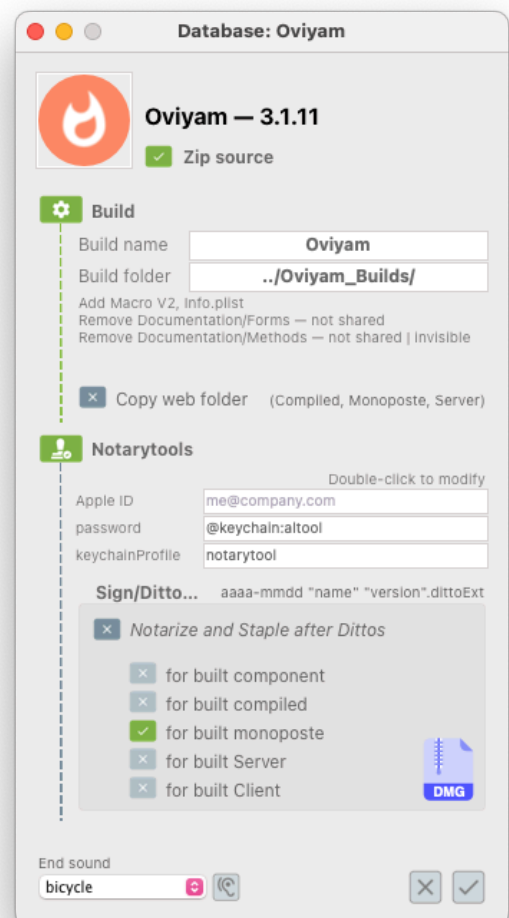
All the settings are saved in a file for your product's name.

- Documents/ogToolsSuite/ogMake/<productName>/, then
 - build_make.json, component_make.json, hdi_make.json, quickNotary.json

Database - Make

On this interface, you can enable/disable some build options.

- Zip source. The name is « <aaaa-mmdd> <baseName> <version> SRC.zip
- Build
 - Displayed, the build name and the build folder taken from the xml build file.
 - Do some stuff, as add Macro v2 folder, info.plist, remove Form's documentation if not shared, and Method's documentation if not shared or invisible.
 - Copy web folder to relevant enabled targets (not for client for security reason)
- Notarytools
 - Apple id, password, keychainProfile
 - Sign/Ditto
 - Choose if you want to Notarize and Staple after Dittos
 - Choose ditto kind - Zip, Dmg, Pkg
 - Enable Notarytools
 - for built component
 - for built compiled
 - for built monoposte
 - for built server
 - for built client
- End sound
 - Choose the end build process sound



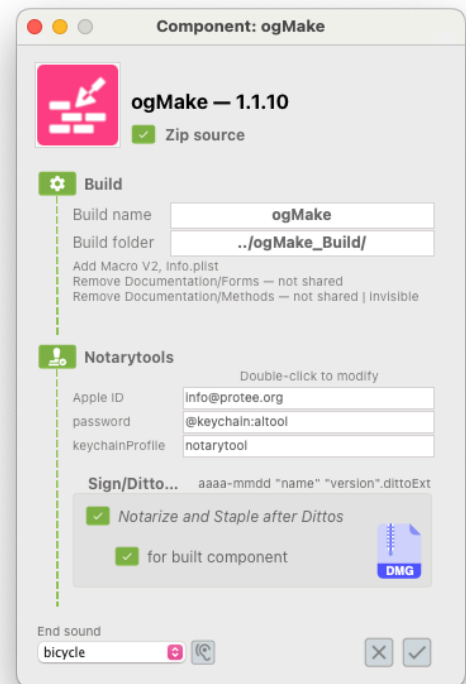


Component - Make

This Form is a reduced set of the general « Build » Form. It proposes a lighter interface, but what you can do with it is the same as the Build form, and for component options only.

Explanations

- Zip source. The name is « <aaaa-mmdd> <baseName> <version> SRC.zip »
- Build
 - Displayed, the build name and the build folder taken from the xml build file.
 - Do some stuff, as add Macro v2 folder, info.plist, remove Form's documentation if not shared, and Method's documentation if not shared or invisible.
- Notarytools
 - Apple id, password, keychainProfile
 - Sign/Ditto
 - Choose if you want to Notarize and Staple after Sign/Dittos
 - Choose ditto kind - Zip, Dmg, Pkg
 - Enable Notarytools
 - for built component
- End sound
 - Choose the end build process sound



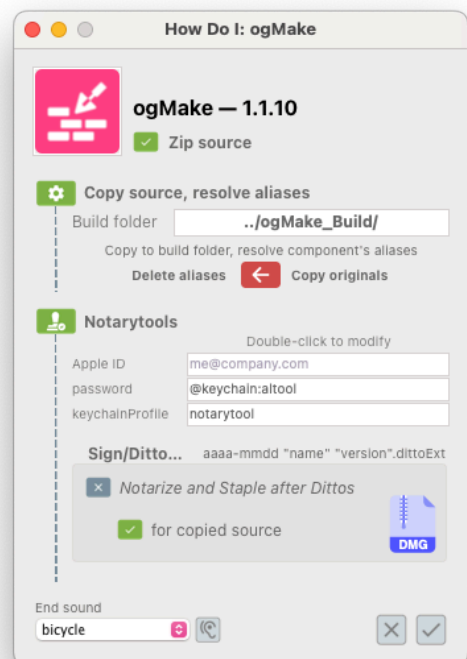
How Do I - Make

This Form is a specific Form. It proposes an interface for those needs:

- Generate a copy with component's aliases resolved with real file if you work with aliases, ie have a copy for your PC.
- Generate a copy with component's aliases removed for source distribution like 4DPop if you work with aliases.
- Generate a copy with component's aliases resolved for source distribution like 4D HDI if you work with aliases.

Explanations

- Zip source. The name is « <aaaa-mmdd> <baseName> <version> SRC.zip »
- Copy source, resolve aliases
 - Displayed, the build folder taken from the xml build file.
 - Choose how to resolve aliases, deleted or original copied.
- Notarytools
 - Apple id, password, keychainProfile
 - Sign/Ditto
 - Choose if you want to Notarize and Staple after Sign/Dittos
 - Choose ditto kind - Zip, Dmg, Pkg
 - Enable Notarytools
 - for copied source
- End sound
 - Choose the end build process sound

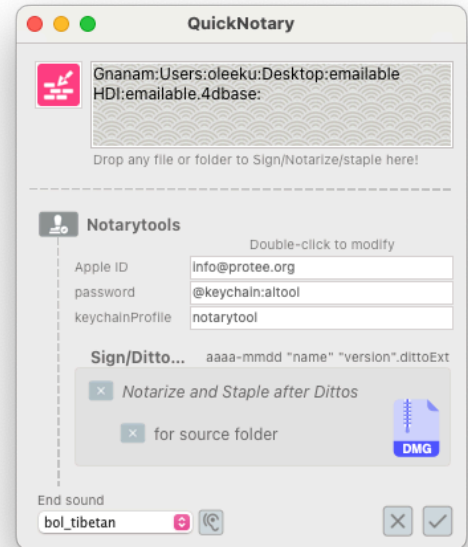




Quick Notary - Make

On this interface, you can apply the Notarytools quickly to a folder you drag&Drop on the zone.

- Zip source. The name is « <aaaa-mmdd> <baseName> <version> SRC.zip
- Notarytools
 - Apple id, password, keychainProfile
 - Sign/Ditto
 - Choose if you want to Notarize and Staple after Sign/Dittos
 - Choose ditto kind - Zip, Dmg, Pkg
 - Enable Notarytools
 - for source folder
- End sound
 - Choose the end build process sound



Methods

wom__storage_prefs

Description

This main object is accessible with this method, and stores all the global preferences.

#DECLARE->\$vJ_prefs : Object			
Parameter	Type		Description
\$vJ_prefs	object	↩	Gives back the main prefs object of ogMake

\$vJ_prefs	Type	Default	Contents
t_name	text	ogMake	Component name
t_version	text	x.y.zz	Current version of svgWidgets
t_lip_app	text	wom	License app name
l_make	number	1	Build kind
l_top	number	140	pixels from top as non usable area for Form windows
t_sound	text	"bell_ayuth a_low"	Default sound for new prefs files.
j_host	object		host options, see below
j_barbers	object		Barber options, see below



j_host

Property	Type	Default	Contents
t_name	text		Host name
t_version	text		Host version
l_make	number	1	Select the right kind of make you want. 0 Database ; 1 Component ; 2 How Do I ; 3 Quick Notary
j_settings_file	object	Null	Host setting file to use if non Null. If null, uses the default « Settings/buildApp.4DSettings ». Before that, look into « SettingsMac » (mac OS) or « SettingsWindows », and if that file is present it is copied into « Settings ». This allows multiple buildApp from multiple platform.
t_callback	text		callback method name in the host, must be shared. Called when all is ready, before zip built and notarizing. Method called: // #DECLARE(\$vJ_dbf : Object; \$vJ_build_options : Object; \$vT_barber : Text)->\$isOk : Boolean

j_barbers

Property	Type	Default	Contents
l_w	num		Internal « _barber" form width (read only)
l_h	num		Internal « _barber" form height (read only)
l_x	num	0	Initial x for barbers
l_y	num	prefs.l_top	Initial y for barbers
l_count_hv	num	5	Horizontally barbers count
l_gap	num	4	Gap between barbers
l_count	num	0	Count of opened barbers (read only)

wom_initRegister

Description

This method initialise the full component. You have to call this method before using any of the other methods. You must provide a correct fixed serial, otherwise you are in demo mode, where you have 30 minutes to do only one build after restart.

#DECLARE(\$vT_host_name : Text; \$vT_serial : Text)->\$is_serial_ok : Boolean			
Parameter	Type		Description
{ \$vT_host_name }	text	→	Host name associated to the serial.
{ \$vT_serial }	text	→	Serial of your component.
\$isOk	boolean	↩	True if register successful.

wom_4dPop

This method is the one invoked on a click to the ogMake button in the ogPop palette.



Following make methods

They use the same description for \$vJ_host_option

Property	Type	Default	Contents
t_name	text		Host name
t_version	text		Host version
{j_settings_file}	object	Null	Host setting file to use if non Null
{t_callback}	text		callback method name in the host, must be shared. Called when all is ready, before zip built and notarizing. Method called: // #DECLARE(\$vJ_dbf : Object; \$vJ_build_options : Object; \$vT_barber : Text)->\$isOk : Boolean

wom_build_make

Description

Open the interface for a complete make.

#DECLARE(\$vJ_host_options : Object)->\$isOk : Boolean			
Parameter	Type		Description
\$vJ_host_options	object	→	Optionnel host options. If not given, uses prefs.j_host
\$isOk	boolean	↩	True if Make successful.

wom_component_make

Description

Open the interface for a reduced component make.

#DECLARE(\$vJ_host_options : Object)->\$isOk : Boolean			
Parameter	Type		Description
\$vJ_host_options	object	→	Optionnel host options. If not given, uses prefs.j_host
\$isOk	boolean	↩	True if Make successful.

wom_hdi_make

Description

Open the interface for the « How Do I » make.

#DECLARE(\$vJ_host_options : Object)->\$isOk : Boolean			
Parameter	Type		Description
\$vJ_host_options	object	→	Optionnel host options. If not given, uses prefs.j_host
\$isOk	boolean	↩	True if Make successful.



wom_quickNotary_make

Description

Open the interface for the « QuickNotary » make.

#DECLARE(\$vT_path : Text)->\$isOk : Boolean			
Parameter	Type		Description
\$vT_path	Text	→	Optionnel path to what to notary
\$isOk	boolean	↔	True if Make successful.

wom_configure

Description

Initialise the prefs.j_host with given values.

#DECLARE(\$vT_name : Text; \$vT_version : Text; {\$vV_build : Variant}; {\$vJ_settings_file : Object})			
Parameter	Type		Description
\$vT_name	text	→	Host name
\$vT_version	text	→	Host version
\$vV_build	variant	→	Boolean: is_component ; Num: 0-3 for l_make. \$vV_build: -1 ask with popup ; 0 generic build ; 1 component ; 2 HDI, 3 quick notarize. You can also put a boolean for 0/1 with false/true.
\$vJ_settings_file	object	→	Host build file if given or default Build file.

wom_configure_vJ

Description

Initialise the prefs.j_host with given properties in \$vJ_storage_prefs, with an overload of given properties to wom__storage_prefs.j_host.

#DECLARE(\$vJ_storage_prefs : Object) // Recover t_name, t_version, j_settings_file			
Parameter	Type		Description
\$vJ_storage_prefs	object	→	Host options, for the following if exist properties: t_name, t_version, l_make, j_settings_file

wox_app_set_t_host_name

You can then add this following to your init code.

```
EXECUTE METHOD("wox_app_set_t_host_name"; *, "myHostName")
```

#DECLARE(\$vT_host_name: Text)			
Parameter	Type		Description
\$vT_host_name	text	→	In ogKrolific: if not empty, all the preferences are taken in a subfolder of that name. This isolates prefs files form projects to projects.



Calls

Register & Configure

Register & configure with object

To initialise and/or register ogMake, you must execute this code on startup, here with ogPop. If you have no serial, just use `wom_initRegister` without parameters or without serial.

All call are with EXECUTE METHOD to free up the host build from those components, like ogPop, ogMake, etc.

```

ARRAY TEXT($aT_Components; 0)
COMPONENT LIST($aT_Components)
If (Find in array($aT_Components; "ogPop")>0)
    EXECUTE METHOD("ogPop_palette")
End if
If (Find in array($aT_Components; "ogMake")>0)
    // ***** ogMake
    // *
    EXECUTE METHOD("wom_initRegister"; $isOk; "myHostName"; "mySerial")
    EXECUTE METHOD("wom_configure_vJ"; *; woxx__storage_prefs)
    // *
    // *****
End if

```

`woxx__storage_prefs` is supposed to give back an object with properties `t_name`, `t_version`, `l_make`, and an optional `j_settings_file` that indicates a build file.

Register & configure with variables

```

ARRAY TEXT($aT_Components; 0)
COMPONENT LIST($aT_Components)
If (Find in array($aT_Components; "ogPop")>0)
    EXECUTE METHOD("ogPop_palette")
End if
If (Find in array($aT_Components; "ogMake")>0)
    // ***** ogMake, for 4D product "xxxx67982"
    // *
    EXECUTE METHOD("wom_initRegister"; $isOk; ""; "mySerial")
    EXECUTE METHOD("wom_configure"; *; "myHostName"; "1.0.00";1)
    // *
    // *****
End if

```

Configure direct through prefs object

```

var $vJ_prefs; $vJ_host_options : Object
EXECUTE METHOD("wom__storage_prefs"; $vJ_prefs) // get prefs object
$vJ_host_options:=$vJ_prefs.j_host
$vJ_host_options.t_name:="myHostName"
$vJ_host_options.t_version:="1.0.00"

```

Demo mode

In demo mode, you get a 30 minutes to work and you will be able to launch one build after each restart.

Build by methods



Build

```
// With no configuration made before
var $vJ_host_options : Object
$vJ_host_options:=New object
$vJ_host_options.t_name:="myHostName"
$vJ_host_options.t_version:="1.0.00"
$vJ_host_options.l_make:=1 // it's a component
EXECUTE METHOD("wom_build_make"; *; $vJ_host_options)

// With configuration made before, following is enough
EXECUTE METHOD("wom_build_make")
```

Component

```
// With no configuration made before
var $vJ_host_options : Object
$vJ_host_options:=New object
$vJ_host_options.t_name:="myHostName"
$vJ_host_options.t_version:="1.0.00"
$vJ_host_options.l_make:=1 // it's a component
EXECUTE METHOD("wom_component_make"; *; $vJ_host_options)

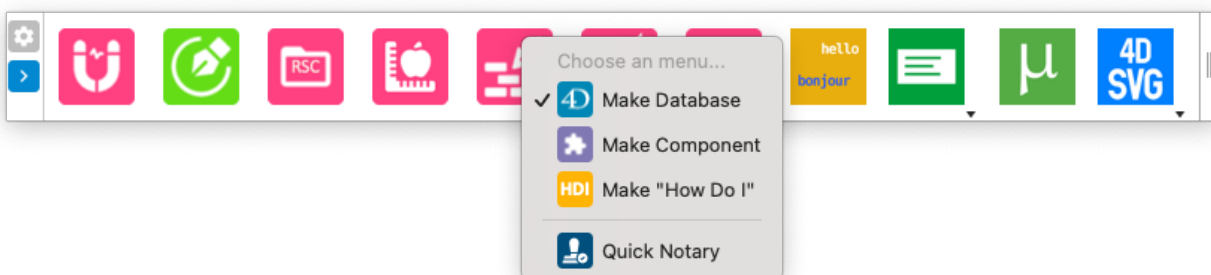
// With configuration made before, the following is enough
EXECUTE METHOD("wom_component_make")
```

Make by 4DPop / ogPop

First, don't forget it is mandatory to configure ogMake before, in order to let it knows the name and version, and eventually its kind of make (Database, Component, How Do I, Quick Notary).

Just click on the icon and it opens the right interface based to j_host.l_make.

With Right click or Macintosh command | Windows Ctrl down, it opens the menu where you might choose witch « make » interface you want to use:



Thanks

Thanks to Patrick Emanuel for our long-term collaboration and help during the tests.

oooOOOooo