

Copyright (c) Prolog Development Center SPb

Author: Victor Yukhtenko

How to build and run WorkspaceManager

The WorkspaceManager application is designed to organize a working environment that includes many projects and related files of various types.

The application is written in the language of the Visual Prolog system (recent changes and builds were performed in version 9xxx).

The application is distributed under the GNU GPL v.3 license, that is, without restrictions on use.

Application source codes are part of the source code array SpbRsolutions.

Compiling and building a project requires a commercial Visual Prolog system.

To build executable applications, classes from the SpbVipTools toolkit of the same version are required.

Location of executable applications

The output codes allow you to build two types of applications running on MS Windows (the operation is tested in Windows 10 environment):

- Mono-application: **WorkspaceManager.exe**
- An application consisting of two executable parts - http-client and http-server:
 - **WSM_http_backend.exe**
 - **WSM_http_frontend.exe**

All these applications are included in SpbRsolutions allow you to use them without compiling projects, and placed at **SpbRsolutions\SpbVipTools\Bin** directory.

Along with these executable applications, all the files are located: they are needed to run them.

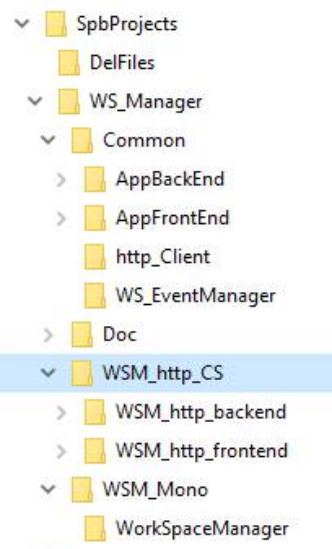
The **WorkspaceManager.exe** application can be started in the usual way.

Before running the **WSMfrontend.exe** application, the **WSMbackend.exe** application must be started (**as the administrator**).

To get started the WorkspaceManager (mono or client-server pair), [see below](#).

Source Code Location

The source code is located in the SpbProjects directory, as shown in the figure:



The mono project is **WorkspaceManager.vipprj** in the **SpbRsolutions \ SpbProjects \ WS_Manager \ WSM_Mono** directory.

The http-Server project is **WSMbackend.vipprj** in the **SpbRsolutions \ SpbProjects \ WS_Manager \ WSM_http_CS \ WSM_http_backend** directory.

The http Client project is **WSMfrontend.vipprj** in the **SpbRsolutions \ SpbProjects \ WS_Manager \ WSM_http_CS \ WSM_http_frontend** directory.

Building the WorkspaceManager.exe application

1. Start the Visual Prolog IDE.
2. Open the project (Project \ Open)
SpbRsolutions\SpbProjects\WS_Manager\WSM_Mono\WorkSpaceManager.vipprj
3. When you open the project for the first time, the IDE will inform you that
 SpbProjects and SpbVipTools
 IDE variables are not defined
 and will prompt them to determine (sequency is not essential).
4. Set the routes for the IDE variables:
 SpbProjects to the directory ... SpbRsolutions \ SpbProjects
 SpbVipTools to the directory ... SpbRsolutions \ SpbVipTools
5. Run the build of the project through the menu **Build \ Build or Build \ Rebuild All**
6. After the project is completed, call the **Build \ Execute** menu (or the **E** icon).
[See below for further action.](#)

Building the WSMbackend.exe application

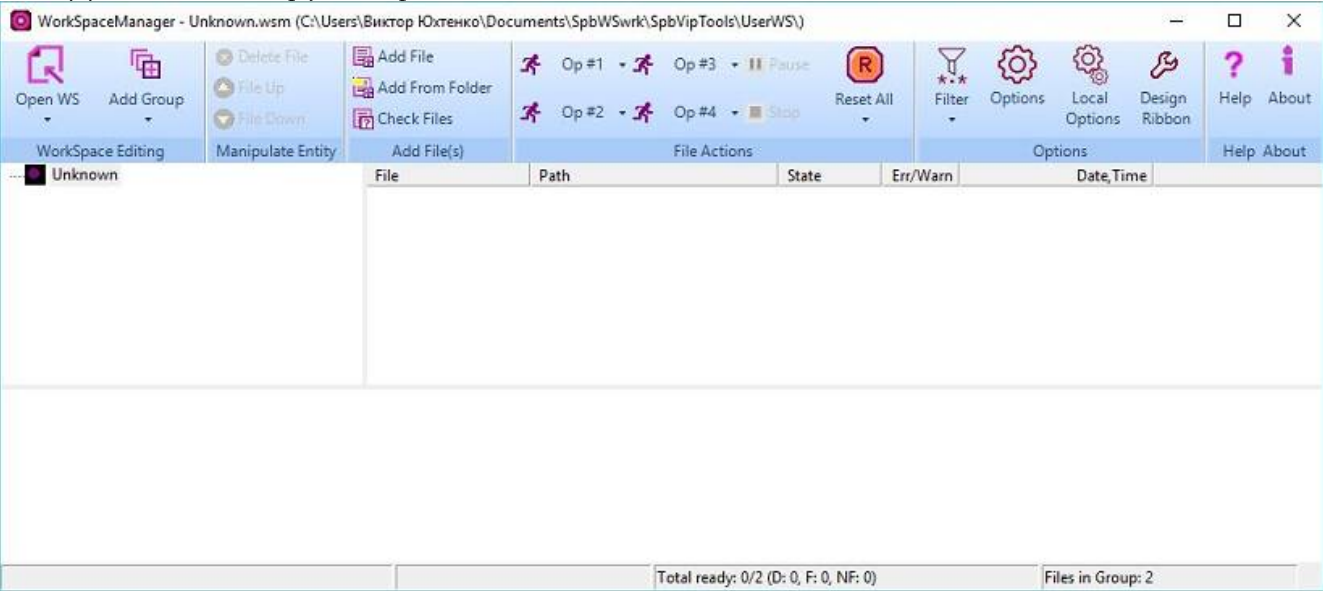
1. Start the Visual Prolog IDE.
2. Open the project (Project\Open)
SpbRsolutions\SpbProjects\WS_Manager\WSM_http_CS\WSM_http_backend\WSMbackend.vipprj
3. If the
 SpbProjects
 SpbVipTools IDE variables
 were not previously defined, define them by setting
 SpbProjects to the directory ... SpbRsolutions\SpbProjects
 SpbVipTools to the directory ... SpbRsolutions\SpbVipTools
4. Start the build of the project through the menu **Build\Build or Build\Rebuild All**
5. After the project is built, call the **Build\Execute** menu (or the **E** icon).
 A window appears showing the start of the server assigned to the port http: // localhost: 5558.

Building the WSMfrontend.exe application

1. Start the Visual Prolog IDE.
2. Open the project (Project\Open)
 SpbRsolutions \ SpbProjects \ WS_Manager \ WSM_http_CS \ WSM_http_frontend \ WSMfrontend.vipprj
3. If the
 SpbProjects
 SpbVipTools IDE variables
 were not previously defined, define them by setting
 SpbProjects to the directory ... SpbRsolutions \ SpbProjects
 SpbVipTools to the directory ... SpbRsolutions \ SpbVipTools
4. Start the project build through the **Build \ Build or Build \ Rebuild All** menu .
 If the WSMbackend.exe application has not been started before, start it
6. If the WSMbackend.exe application was started,
 After the project is compiled, call the **Build\Execute** menu (or the **E** icon).
[See below for further action.](#)

Try Run WorkSpaceManager.exe or WSMfrontend.exe

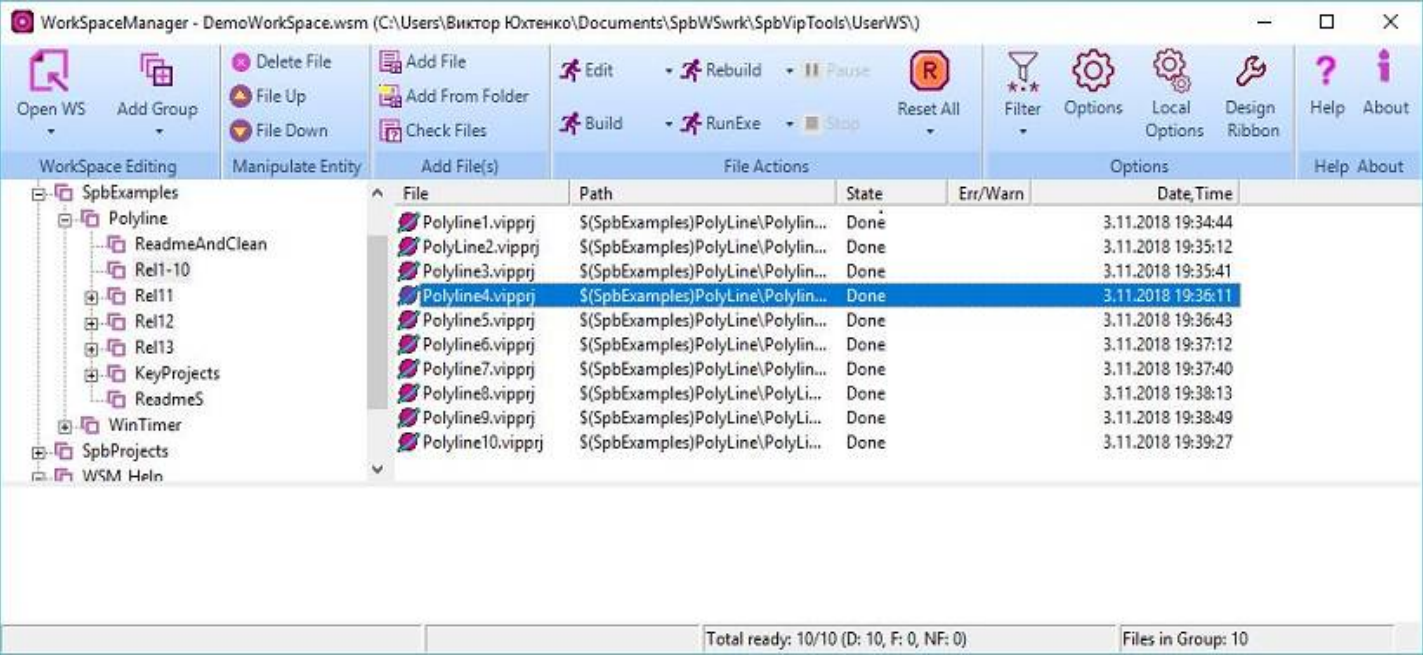
Initially you will see the empty WorkSpace:



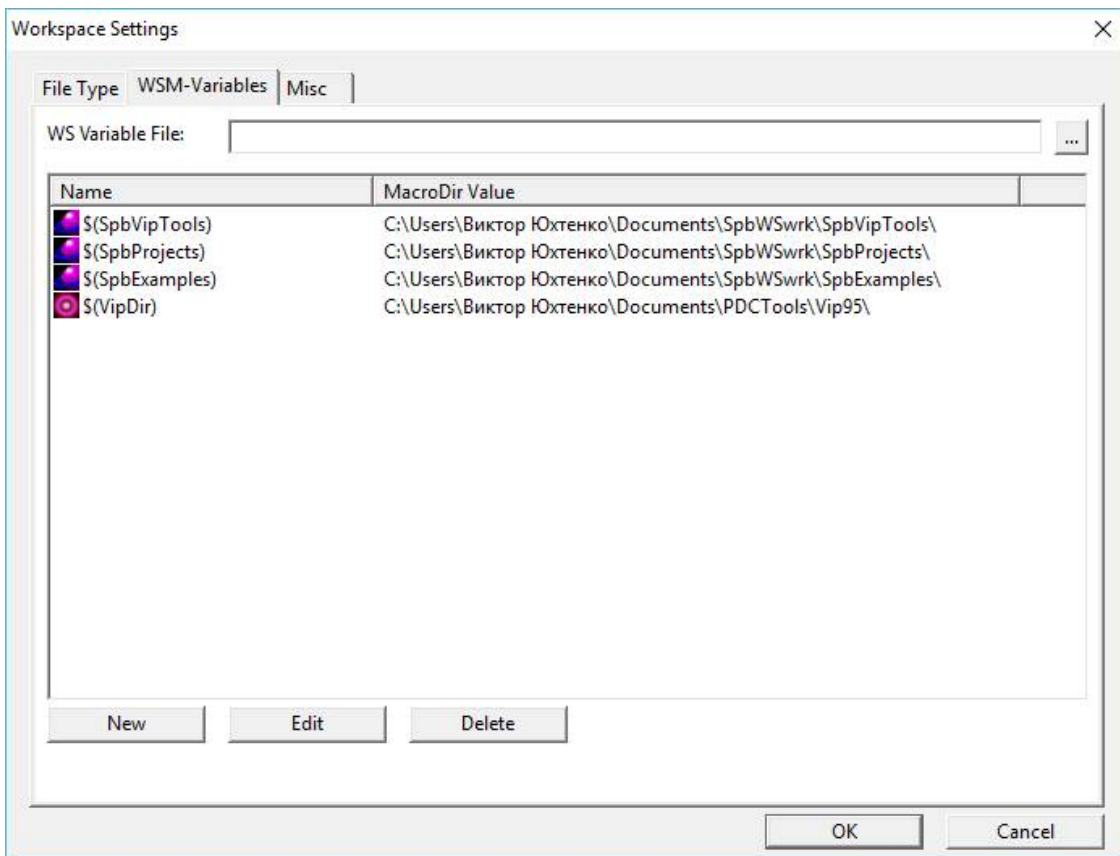
The control panel of the application can be customized by calling the editor using the **Design Ribbon** icon.
The SpbRsolutions collection contains an example of the workspace located in the **SpbRsolutions\UserWS** directory in the **DemoWorkSpace.wsm** file .
Press icon "Open WS" and choose the file

SpbRsolutions\UserWS\DemoWorkSpace.wsm .

This workspace example contains all the projects of the SpbRsolutions set and the auxiliary files.



In this case the installation workspace stored in OptionsWSM.xml file involves determining the wsm-Variables shown in the figure below



brighter icons marked virtual directories defined in the Visual Prolog system and only where they can be edited:

SpbVipTools - ... SpbRsolutions \ SpbVipTools
 SpbExamples - ... SpbRsolutions \ SpbExamples
 SpbProjects - ... SpbRsolutions \ SpbProjects

VipDir variable - refers to the location of the Visual Prolog system.

The **WSM** application knows nothing about the Visual Prolog system and defining the **VipDir** variable is the only way to let it know. The remaining settings can be viewed in the form of settings by clicking the corresponding icon on the panel.

To run a trial run of files

- Adjust the size of the message field (if it is not visible, increase the window size)
- Select one of the nodes containing Visual Prolog projects
- Click the **run** icon (**runAll** will invoke the compilation and building all projects related to the choosen node)
- Double click on the project will cause the selected project to edit

To use the other properties of the application, see **Help**