

<b>1.</b>	<b>UM CONFIGURATION AND STRUCTURE.....</b>	<b>1-2</b>
<b>1.1.</b>	<b>Configuration of program package UM .....</b>	<b>1-2</b>
1.1.1.	UM location.....	1-2
1.1.2.	Organization of working with UM .....	1-2
1.1.3.	System parameters .....	1-3
1.1.4.	Portability of models.....	1-3
<b>1.2.</b>	<b>UM structure .....</b>	<b>1-4</b>
1.2.1.	List of UM modules .....	1-4
1.2.2.	UM Demo.....	1-4
<b>1.3.</b>	<b>Hardware requirements .....</b>	<b>1-5</b>
<b>1.4.</b>	<b>Command line parameters.....</b>	<b>1-5</b>
1.4.1.	Command line parameters for UM Input program.....	1-5
1.4.2.	Command line parameters for UM Simulation program.....	1-6

# 1. UM configuration and structure

## 1.1. Configuration of program package UM

Program package UM contains a standard kernel, which includes programs and auxiliary files. UM object (computer models of multibody systems made by user) can be located independently from the UM kernel. The kernel may be installed on a server whereas objects – on user's computers.

Working with UM includes:

- Description of objects (uminput.exe);
- Generation of equations of motion (uminput.exe);
- Compilation of equations by an external compiler (uminput.exe);
- Simulation of the object dynamics (umsimul.exe).

Following external compilers are supported in the present version:

- Borland Delphi 4.0-7.0;
- MS Visual C++ 5.0-6.0;
- Borland Builder C++ 3.0-5.0,
- BDS 2005-2006.

### 1.1.1. UM location

The UM kernel is located in several directories, which are situated in the UM main directory with an arbitrary name (hereinafter - .\). These directories cannot be renamed; it is not allowed to remove files from them.

- *.\bin* – includes executable files of UM;
- *.\help* – the folder with help files (\*.hlp);
- *.\com* – includes files necessary for compilation of equations and programming in the UM environment.
- *.\components* – the folder with the set of components;
- *.\demo* – the folder with demo-examples;
- *.\drivers* – the folder with electronic key drivers (for protected versions);
- *.\lessons* – tasks from the lessons;
- *.\library* – examples of simulation;
- *.\manual* – the folder with the manual;
- *.\plugins* – the set of dynamic libraries with functionals;
- *.\simulink* – the folder with files necessary for integration of Matlab/Simulink models into UM;
- *.\templates* – templates for import of data from UM into MS Excel;
- *.\tutorial* – tasks for manual parts “Getting started...”;

### 1.1.2. Organization of working with UM

Every object (model) created by UM is situated in a separate directory. *The name of the directory is used as the object name.* There exist syntactical restrictions on the name: it can contain Latin letters, digits, the character ‘\_’, the first symbol must be a letter. If the user renames an object directory manually, he must generate and compile equations anew.

### 1.1.3. System parameters

Universal Mechanism uses point as a decimal separator and space as a group separator. If your current settings are different then they will be fixed automatically. You can change default setting using Windows Control Panel.

### 1.1.4. Portability of models

File *input.dat* contains the description of object parameters. If the user does not use the *Control File* as well as own files for programming in the UM environment, the *input.dat* file is the only file that contain information about model as mechanical system. If user uses programming in the UM environment then it is necessary to keep the *input.dat*, *control file* and *user's files used in control file* (see Chapter 6).

Thus, the above files are only essential to transfer the model to another computer, to send the model by e-mail and so on. All other files, which are necessary for simulation of the model, can be generated by UM.

For instance, to transfer a model to another computer, the user should

- create a directory with the new name of the model;
- copy the *input.dat* and (if necessary) the control file and user's source files;
- generate and compile equations of motion for this new object;
- the object is ready for simulation.

**Note** Some identifiers and units in equations of motion of object include the name of the object. If the user renames the object directory he/she should either rewrite the old *control file* and (if necessary) transfer user's procedures in the new version of this file, or correct the corresponding identifiers and unit names in the old *control file* and rename it.

## 1.2. UM structure

UM consists of two main programs located in the `.\bin` directory.

- **Data input program (uminput.exe)**

The program is used for description and modification of objects, for generation of equation of motion. It has a multi-project environment: the user can open and modify several objects simultaneously.

- **Simulation program (um simul.exe)**

The program implements: the numeric integration of equations of motion with the parallel animation of the object motion and plotting graphs; linear analysis of the object equations; multivariant projects and parametric optimization and so on. The program imports equations of motion created by the input program in the `umtask.dll` file.

The simulation program can be used separately from the input program for simulation of previously prepared models. The simulation program has a single project environment.

### 1.2.1. List of UM modules

Universal Mechanism includes the following modules:

- **UM Base** – base version includes the pre- and postprocessors.
- **UM Subsystems** – additional module which allows describing large objects as trees of subsystems.
- **UM Automotive** – additional module for simulation of road vehicle dynamics.
- **UM Caterpillar** – additional module for simulation of caterpillar dynamics.
- **UM Loco** – additional module for simulation of railway vehicle dynamics
- **UM Optimization** – additional module which is intended for parametrical scanning of the dynamical behavior and optimization.
- **Service of distributed calculations** – this service extends possibilities of the UM Optimization module. It allows using any computer in a local or global network for parallel numerical experiments.
- **UM FEM** – additional module for including elastic bodies into models
- **UM Control** – Matlab/Simulink interface. It gives a possibility to introduce Matlab/Simulink schemes (control, electrodynamics, hydraulics) into UM models.
- **CAD interfaces** – import graphical and inertia data from SolidWorks and Autodesk Inventor.
- **UM Train** – additional module for simulation of train longitudinal dynamics.
- **UM Train 3D** - additional module for including in a train model 3D railway vehicle models from UM Loco.
- **UM Ballast** – additional module for simulation of dynamics of granular media systems in 2D.
- **UM Rail\Wheel Wear** – Additional module for prediction of evolution of railway wheel profile due to wear.

### 1.2.2. UM Demo

UM Demo is free for non-commercial use. The demo version has the following restrictions:

- 2 subsystems at most;
- 5 elements of any other types (bodies, joints, forces of each type) at most.

### 1.3. Hardware requirements

UM runs under Windows 98/2000/NT/XP/Me/Vista.

Minimal hardware configuration:

- CPU frequency 166 MHz;
- 64 Mb memory;
- Hard disk memory for UM installation – 120 Mb.

A video adapter with acceleration of the OpenGL functions is desirable. Video adapters based on NVidia technologies are preferable.

### 1.4. Command line parameters

The **UM Input** and **UM Simulation** programs can be run with the help of command lines. This opportunity is useful for automatic run programs from external programs (e.g. Maple or user's programs).

#### 1.4.1. Command line parameters for UM Input program

The user may generate the object data file *input.dat* with the help of some external program (for example, MAPLE, MATLAB, MATEMATICA). In this case the equations of motion can be generated and/or compiled by **UM Input** program in an automatic mode.

Parameters and options of the command line:

"[path to uminput.exe]uminput.exe" [path to object] [options]

Options	Comments
/g	Generate equations of motion. If the object contains external subsystems, the completeness of description and generation of equations for each the subsystem is verified. If equations are not prepared, generation is fulfilled automatically.
/b	Forced generation of equation
/c	Compile equations of motion
/n	Rewrite old control file. The option is valid if one of the options /c/b/g is presented
/i	Display generation messages in a separate window. If the option is not presented the UMINPUT is executed in a full console mode without visualization of messages. The option is valid if one of the options /c/b/g is presented
/f[path to file]	All messages are directed into the file. The option is valid if one of the options /c/b/g is presented

**Remark.** One of the options /c/b/g switches on the console mode.

#### Examples

"D:\um\bin\uminput.exe" "d:\My Objects\UMObject"

Run the **UM Input** program in the usual mode and open the object UMObject located in the d:\My Objects directory.

"D:\um\bin\uminput.exe" "d:\Objects\UMObject" /g /c /i "/f d:\Objects\UMObject\outinfo.txt"

Run the **UM Input** program in the console mode. Generate and compile equations of motion. Generation messages are displayed in the window. Program messages are stored in the file outinfo.txt.

### 1.4.2. Command line parameters for UM Simulation program

The command line allows running the *umsimul.exe* program in an automatic mode of solving object equations. The command line indicates files, which contain initial conditions, identifier values, configurations and so on. Unlike the usual mode, simulation process starts right after the program start. Simulation results can be stored in user's defined files. The simulation program closes right after the simulation finish or interruption.

The automatic mode allows the user to organize the simulation and results processing with the help of an external program.

Syntax of the command line:

"[path to UmSimul.Exe]\UmSimul.exe" [path to object] [options]

Options	Comments
/s	Sets the automatic mode
/c[Name].icf	Name of configuration file (program desktop, parameters of solver)
/p[Name].par	File with values of identifiers
/r[Name].rwc	Rail vehicle configuration file (wheel/rail profiles, track profiles etc.)
/i[Name].xv	File with initial conditions (values of coordinates and their time derivatives at t=0)
/v[Name].var	Name of file with automatically calculated variables
/x[Name].xva	Switches on the XVA mode. Data will be saved in the specified file.
/f[Path\name of file]	Text file for UMSimul output messages.

**Remark.** Files \*.icf, \*.xv, \*.rwc, \*.par, \*.var should be created beforehand and saved in the directory of the object. If one of the files is not presented in the command line, the corresponding last.\* file is used (if it exists).

#### Examples

"D:\um\bin\umsimul.exe" "d:\My Objects\UMObject"

Run the **UM Simulation** program in a usual mode and load the model *UMObject*, located in the "d:\My Objects" directory.

"D:\um\bin\umsimul.exe" "d:\My Objects\UMObject" /s /c*some.icf* /v*test.var* /x*test.xva*

Run the **UM Simulation** program in the automatic mode. Read: the configuration for the file *some.icf*; the list of automatic calculated variables –from the file *test.var*. Switch on the XVA mode and save data in the file *test.xva* (the file is stored in the object directory). The files *test.var*, *some.icf* should be created beforehand and located in the object directory "d:\My Objects\UMObject".