

# gO:RUN User Guide

---

Release 3.5  
June 2012

Process Systems Enterprise Limited  
6th Floor East  
26-28 Hammersmith Grove  
London W6 7HA

e: [info@psenterprise.com](mailto:info@psenterprise.com)  
w: [www.psenterprise.com](http://www.psenterprise.com)  
t: +44 20 8563 0888  
f: +44 20 8563 0999





## **gO:RUN User Guide**

### **Release 3.5**

**June 2012**

Copyright © 1997-2012 Process Systems Enterprise Limited  
Process Systems Enterprise Limited  
6th Floor East  
26-28 Hammersmith Grove  
London W6 7HA  
United Kingdom  
Tel: +44 20 85630888  
Fax: +44 20 85630999  
WWW: <http://www.psenetprise.com>

### **Trademarks**

gPROMS is a registered trademark of Process Systems Enterprise Limited ("PSE"). All other registered and pending trademarks mentioned in this material are considered the sole property of their respective owners. All rights reserved. Legal notice No part of this material may be copied, distributed, published, retransmitted or modified in any way without the prior written consent of PSE. This document is the property of PSE, and must not be reproduced in any manner without prior written permission.

### **Disclaimer**

gPROMS provides an environment for modelling the behaviour of complex systems. While gPROMS provides valuable insights into the behaviour of the system being modelled, this is not a substitute for understanding the real system and any dangers that it may present. Except as otherwise provided, all warranties, representations, terms and conditions express and implied (including implied warranties of satisfactory quality and fitness for a particular purpose) are expressly excluded to the fullest extent permitted by law. gPROMS provides a framework for applications which may be used for supervising a process control system and initiating operations automatically. gPROMS is not intended for environments which require fail-safe characteristics from the supervisor system. PSE specifically disclaims any express or implied warranty of fitness for environments requiring a fail-safe supervisor. Nothing in this disclaimer shall limit PSE's liability for death or personal injury caused by its negligence.

# Contents

Chapter 1	Introduction .....	4
1.1	gO:RUN – an overview .....	5
1.2	System set-up .....	5
Chapter 2	Creating a gO:RUN application.....	6
2.1	The gPROMS activity.....	7
2.2	Exporting your gPROMS activity using ModelBuilder .....	7
Chapter 3	Using gO:RUN .....	9
3.1	General .....	10
3.2	From a command window.....	10
3.3	Within external software.....	10
Chapter 4	Trouble shooting .....	12
4.1	Common problems .....	13
Appendix A	Using gO:RUN with Microsoft Excel™ .....	14
A.1	Running the example .....	15
A.2	Using example as template to run new problems .....	16



## Chapter 1 Introduction

---

## 1.1 gO:RUN – an overview

gO:RUN allows a gPROMS model-based activity to be executed using a pre-existing model. Typically, the latter will have already been developed, validated and tested using gPROMS ModelBuilder, from which it will have been exported in encrypted form.

gO:RUN is designed to provide a simple yet comprehensive, easy-to-use “runtime” access to gPROMS’ capabilities using a single command line.

The gO:RUN command may be issued either directly from a terminal, or more usually as a system command from within other software (e.g. Microsoft EXCEL™) used to provide a simplified user interface to a gPROMS model. The latter will normally communicate with this user interface via a combination of Foreign Object and Foreign Process interactions. Uses of such an interface include:

- adding a front end to a gPROMS activity, for example presentation or operator training purposes.
- using gPROMS to provide another software package with calculations

## 1.2 System set-up

gO:RUN is included as part of the standard gPROMS installation process, no additional software components are required.

### 1.2.1 Licensing

To use gO:RUN an additional licence key is required – contact [sales@psenterprise.com](mailto:sales@psenterprise.com) to obtain an appropriate licence.

### 1.2.2 Confirm environment variable settings

The system environment variable `GPROMSHOME` should be set correctly to point to the gPROMS directory of your gPROMS installation<sup>1</sup>.

Additionally, the `bin` subdirectory of `GPROMSHOME` should be added to the `PATH` environment variable:

#### 1.2.2.1 Windows

`%GPROMSHOME%\bin` has to be part of `PATH`

#### 1.2.2.2 UNIX

`$GPROMSHOME/bin` needs to be in the `LD_LIBRARY_PATH`

On both Windows and Linux, for standard gPROMS installations these environment variables should have been set automatically during the installation process and in these cases no action is required by the user.

Please contact your system administrator if you are unsure on the location of the gPROMS installation or on the procedure for setting environment variables.

---

<sup>1</sup> The gPROMS installation directory is typically `C:\Program Files\PSE`. In such a case `GPROMSHOME` must be set to `C:\Program Files\PSE\gPROMS`.



## Chapter 2 Creating a gO:RUN application

---

## 2.1 The gPROMS activity

There are no limitations on the gPROMS activities that can be used in gO:RUN therefore any activity created and run in ModelBuilder can be used in a gO:RUN application.

A gPROMS activity to be used within gO:RUN must be at a minimum a complete gPROMS `PROCESS` entity for simulation. For the other activities (optimisation, estimation and experiment design) the corresponding entities must also be created.

Most gO:RUN applications will actually use one or a combination of the Foreign Process and Foreign Object interfaces to link the activity to the external software. For more information about writing models that use the Foreign Process and Foreign Object interfaces see the gPROMS Advanced User Guide. The Foreign Process and Foreign Object can be used together and can be the same piece of software, although they do not have to be. It should be noted that only one Foreign Process can be used by an activity.

Before exporting a model for use with gO:RUN it is vital that the model has been thoroughly validated and tested inside the ModelBuilder environment. There are no specific changes that have to be made to a gPROMS `PROCESS`, or any other activity, to allow it to be used with gO:RUN (there are no limitations on the activity). However, if it is required for the gO:RUN application to send results to gRMS (the default gPROMS output) then the `PROCESS` should be modified so that the following `SOLUTIONPARAMETER` is added:

```
SOLUTIONPARAMETERS
  gRMS :=      ON;
```

Once the activity has been created and successfully tested, it can be exported from ModelBuilder to be used with gO:RUN. It is highly recommended that before exporting the activity it is tested within ModelBuilder to ensure it behaves as desired.

## 2.2 Exporting your gPROMS activity using ModelBuilder

To use the activity in gO:RUN it must be exported from gPROMS ModelBuilder. To do this select the corresponding `PROCESS` for the desired activity in the project tree to be exported and use the “Export” tool provided under the Tools menu (see Figure 1).

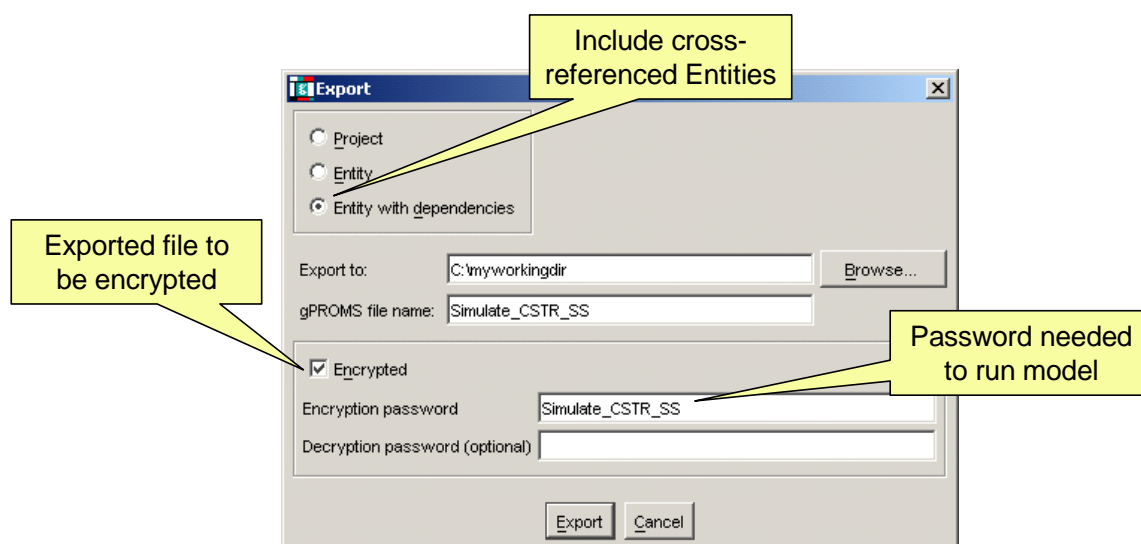


Figure 1 Exporting an activity for use with gO:RUN

In gPROMS when exporting an individual Entity, the user may request that all other Entities on which it depends (via cross-referencing) should be exported with it – this option *must* be selected here. This ensures that all the required entities are included in the encrypted file and also that the entities present for activities other than simulation are exported. For example, if it is desired to run a Parameter Estimation through gO:RUN, the Parameter Estimation and experiment entities also must be exported. This is done automatically if the `PROCESS` is exported with dependencies.

The model should be exported as an encrypted file using an encryption password. Note that the decryption password can be ignored. If the file is not encrypted gO:RUN will not be able to use the activity. It is important to remember the encryption password as this is required to run the activity.

The export creates the following directory structure in the location selected by the “Export to” field in the export dialog:

- input directory – this is where the .gENCRYPT file is located. Also, any other activity files (Optimisation, Parameter Estimation, Experiment Design) will be located here;
- save directory – any saved variable sets used by the activity will be exported here;
- output directory – this is where the results files for Optimisation and Parameter Estimation are saved when the activity is performed in gO:RUN.





## Chapter 3 Using gO:RUN

---

## 3.1 General

There are two ways in which gO:RUN can be used:

1. As a standalone product from the command window. This is useful, for example, in running gPROMS activities in batch mode.
2. Within external software. This is useful for providing a simplified interface to a gPROMS model.

Once an activity has been exported from ModelBuilder, as described in section 2.2 it can be used by gO:RUN. For all cases, gO:RUN must be executed at the same level as the input directory containing the gENCRYPT file. For example, if the activity is exported to the location "C:\myworkingdir" the gO:RUN command must be executed in this directory as this is where the input directory is located.

## 3.2 From a command window

gO:RUN is designed to provide a simple yet comprehensive, easy-to-use "runtime" access to gPROMS' capabilities using a single command line of the form:

```
gORUN.exe [-silent] [-s <maxEqnSize>] <model> <activity type> <activity name> <password>
```

The arguments of the above command are as follows:

- `-silent` reduces verbosity of activity execution. This argument is optional;
- `<maxEqnSize>` is the maximum number of equations<sup>2</sup> allowed by the gO:RUN licence to be used (see section 1.2.1). This argument is optional;
- `<model>` is the name of the file<sup>3</sup> containing the encrypted gPROMS model to be used;
- `<activity type>` is the type of the activity to be executed, i.e. `sim`, `opt`, `est` or `exd` for simulation, optimisation, parameter estimation and experiment design respectively;
- `<activity name>` is the name of the activity to be executed; this is normally the name of the PROCESS entity in the gPROMS model;
- `<password>` is the password used to encrypt the gPROMS model when it was exported from the gPROMS ModelBuilder.

## 3.3 Within external software

gO:RUN also provides an easy mechanism for running a gPROMS activity inside other software packages (such as MS Excel).

---

<sup>2</sup> This parameter is optional. If omitted, a licence allowing models over 50,000 equations will be assumed.

<sup>3</sup> This is normally a file of type `.gENCRYPT`.

gO:RUN is started through the exact same command line as described in section 3.1. Therefore the external software from which gO:RUN is used must send this command line to a terminal. The command line may be hard coded or created on the fly by the software.

The specifics on doing this depends on the software application from which gO:RUN is being called from, and the relevant documentation for that software should be referred to. For example, it is possible to call gO:RUN using Visual Basic allowing a gPROMS activity to be run from Microsoft EXCEL™.

An easy way of sending the command line to a terminal is to use a batch file. This batch file may be manually written and then called by the external software or it could be automatically written by the software and then run. A batch file is just a simple text file with the file extension .bat. An example of such a file is shown in Figure 2 below, where gasflowEUI is the name of the .gENCRYPT file, run\_gasflowEUI is the name of the PROCESS and password is the encryption password chosen when exporting the activity.

```
@echo off  
gORUN.exe gasflowEUI sim run_gasflowEUI password
```

Figure 2 gORUN.bat file



## Chapter 4    Trouble shooting

---

## 4.1 Common problems

Table 1 below lists common problems that may be encountered when using gO:RUN and describes how to overcome them.

Problem	Cause	Fix
Error message in command window:  'goRUN' is not recognized as an internal or external command, operable program or batch file.	The computer has not been set up correctly to find the gO:RUN installation	<ul style="list-style-type: none"> <li>ensure a gPROMS 2.3 or later installation is available</li> <li>check that the system has been set up correctly to use this installation, see 1.2</li> </ul> <p>The most common cause of this error will be that the environment variables are not set correctly.</p>
Failure of gPROMS activity	Commonly caused by change in input to gPROMS model from external software.	<p>This kind of troubleshooting should be performed on the model inside ModelBuilder.</p> <p>Note, it is recommended that before a model is exported to be used in a gO:RUN application that it is thoroughly tested inside ModelBuilder. In particular, it should be ensured that the model can be solved for the expected range of inputs.</p>
Failed to connect to gRMS, e.g.  gRMS output channel: Fatal error - Failed to connect to gRMS localhost:9876. Socket Error Code: 10061. ERROR: Cannot create Output Channel instance 'gRMS::COLUMNOPT.gRMS' gRMS::COLUMNOPT.gRMS: Output Channel initialisation failed.	gRMS not started, or not started in correct location.	gO:RUN only sends results to gRMS if the <code>SOLUTIONPARAMETER</code> has been set in the exported model, section 2.1.

Table 1 Problems using gO:RUN

## Appendix A Using gO:RUN with Microsoft Excel™

---

This example is included in the gPROMS installation and can be found in the “examples\gO Products examples” folder in the installation. As well as providing a means to interface gO:RUN with Excel this example also uses Excel as a foreign process and object.

## A.1 Running the example

### A.1.1 From Excel

To run the example from Excel, without the need for opening gPROMS:

1. Start MS Excel and load Batch reactor.xls. This file can be located at the following location:  
  
    \gPROMS\examples\PML examples of the gPROMS installation.
2. In the Batch worksheet, choose the desired debugging mode:  
  
    0 – debugging is switched off and gO:RUN is started in the background.  
  
    1 – debugging is switched on and the gO:RUN command window is opened allowing the gPROMS execution output to be viewed.
3. Go to the Reactor Schematic worksheet, click on the Solve button to execute the simulation.

When this is done, gO:RUN is automatically started, with the correct command sent to simulate the example.

The cells shaded in yellow can be changed to modify the inputs to the simulation. Making changes to cells in the Initial data table only affects the initial conditions, the other yellow cells can be changed during a simulation. The cells shaded in grey are values sent by gPROMS to Excel.

### A.1.2 From ModelBuilder

To run the example within ModelBuilder:

1. Start ModelBuilder.
2. Load Process Model Library.  
  
    Note: The Process Model Library is located in the PML subdirectory in gPROMS distribution.
3. Open the PML Batch Reactor Example.gPJ  
  
    Note: The PML Batch Reactor Example project file is located in the PML examples subdirectory in the examples subdirectory in gPROMS distribution.
4. Go to Test\_Batch\_Reactor\_FO Process and modify the path for FO and FPI accordingly so that ModelBuilder can launch Batch Reactor.xls correctly. For example:  
  
    RxnData := "ExcelFO::C:\Program Files\Pse\gPROMS\gPROMS\examples\PML examples\Batch\_Reactor.xls";  
  
    FPI := "ExcelFP::C:\Program Files\Pse\gPROMS\gPROMS\examples\PML examples\Batch\_Reactor.xls" ;
5. Execute the simulation.

## A.2 Using example as template to run new problems

This example contains the necessary Visual Basic macros needed to run user defined activities in gO:RUN from Excel, and so can be used as a template. This allows model developers to easily embed their gPROMS models inside excel with no further programming required.

### A.2.1 Exporting from ModelBuilder

Before this can be done, the model activity must first be exported from ModelBuilder as described in section 2.2. If Excel is also used as a Foreign Object or Process these settings must be changed so that there is no path before the Excel file name.

### A.2.2 Modifying Excel file to run exported model

The excel file should be saved in the same location as to where the model was exported (at the same level as the input directory, etc). The Batch worksheet can then be modified so that the exported model will execute when the solve button on the Reactor schematic worksheet is pressed. The following cells in the Batch worksheet can be modified:

- A1 – the exported file name
- A2 – the activity type
- A3 – the activity name
- A4 – the password