

# IYOPRO User Guide

## Table of contents

---

Welcome to YOPRO .....	13
Getting started with YOPRO .....	13
Introduction .....	14
About YOPRO .....	14
System requirements .....	14
Getting help .....	15
YOPRO editions .....	15
YOPRO terms and conditions .....	16
What's new in YOPRO .....	20
v1.5.6.7 .....	20
v1.6.4.9 .....	21
v1.7.4.9 .....	22
v1.8.5.2 .....	22
v1.8.9.0 .....	23
v1.9.4.1 .....	23
v1.9.8.1 .....	24
v1.10.7.4 .....	25
v1.11.8.0 .....	25
v1.13.3.5 .....	26
v1.16.0.1 .....	28
v1.19.0.4 .....	29
v1.21.8.5 .....	31
v1.27.4.1 .....	32
v1.30.7.6 .....	34
v1.35.8.6 .....	37
v1.38.8.3 .....	38
v1.48.8.1 .....	39
v1.51.2.6 .....	40
v1.54.1.3 .....	41
v1.57.2.0 .....	41
v1.59.5.8 .....	42
v2.1.63.3.6 .....	42
v2.1.66.4.4 .....	43
v2.1.69.0.9 .....	44
v2.1.69.8.1 .....	45
v2.1.73.5.4 .....	45
How to buy YOPRO .....	46
Overview of the user interface .....	47
Application menu .....	48
Quick access toolbar .....	48
Ribbon tabs .....	49
The Repository .....	50
The Toolbox Area .....	51
The Process Explorer .....	52
The Properties Area .....	53
The Advanced Properties Window .....	55
The Output Window .....	55
Zoom, Pan, Fit .....	56
Find and replace .....	58
Editing rich text .....	59

Undo and Redo .....	60
Themes .....	60
Styles .....	61
Element arrangement .....	62
Align elements .....	63
Edit sizes .....	63
Quick start guides .....	64
Launching YOPRO .....	64
Launching YOPRO Portal .....	64
Register a new account .....	65
Create a new diagram .....	67
Register and activate software keys .....	68
The Repository .....	69
The Desktop .....	69
Edit Preferences .....	70
Edit the User Profile .....	71
Dashboard .....	72
Working with files and folders .....	72
My Files vs. Team Files .....	73
Creating folders .....	74
Open Files for edit .....	74
Delete files .....	75
Move files .....	75
Copy files .....	75
Upload files .....	76
Search files .....	76
Export folders .....	79
Flow layout .....	79
File properties .....	80
Rename files .....	80
Edit file security .....	81
Working with teams .....	82
Create a new team .....	84
Delete a team .....	85
Edit team properties .....	85
Mail Server .....	86
Process Documentation .....	87
Library Elements .....	87
Register software key .....	88
Assign software keys to users .....	89
Add Users .....	89
Create new users .....	90
Inviting existing users .....	92
Create or modify a Team Access Code .....	92
Delete users .....	92
Create a usergroup .....	93
Delete a usergroup .....	93
Edit the content of a usergroup .....	94
Join team .....	94
Join by invitation .....	95
Join by Team Access Code .....	95
Working with the Library .....	96
Creating Library Elements .....	97

Editing Library Elements .....	97
Saving Library Elements .....	98
Using Library Elements .....	100
Recent items .....	101
The Portal .....	102
The app internal portal .....	102
Desktop .....	103
Process Documentation .....	103
Overview Page .....	103
Detail Page .....	104
History .....	105
Dependencies .....	106
Comments .....	106
Diagram .....	108
News .....	108
Subscription .....	109
Output .....	110
Workflow .....	111
Tasks .....	112
Substitutes .....	113
Set substitutes .....	113
Set user absent .....	114
Delegation .....	115
Instances .....	115
Runnable .....	117
The web portal (YOPRO Web) .....	118
Dashboard .....	120
Process Documentation .....	120
Overview Page .....	120
Detail Page .....	121
History .....	122
Dependencies .....	123
Comments .....	123
Diagram .....	124
News .....	125
Subscription .....	126
Output .....	127
Workflow .....	128
Tasks .....	128
Substitutes .....	131
Set substitutes .....	132
Set user absent .....	133
Delegation .....	133
Instances .....	134
Runnable .....	136
Overview of workflow list icons and their meaning .....	137
Business Process Modeling .....	140
Single file diagramming .....	140
Create a new diagram .....	140
Edit an existing diagram .....	141
Save a diagram .....	141
Version of a diagram .....	142
Tabular description editor .....	143

Modeling tips .....	144
BPI projects .....	145
Create a new BPI project .....	147
Add a new diagram .....	147
Add an existing diagram .....	148
Add a folder .....	149
Remove an item .....	150
Rename an item .....	151
Rearrange an item .....	151
Diagram types .....	152
BPMN 2.0 diagrams .....	152
Collaboration diagram .....	152
Choreography diagram .....	153
Conversation diagram .....	154
Shapes .....	155
Pools and Lanes .....	155
Gateways .....	157
Events .....	159
Activities .....	161
Data .....	165
Annotations and Artefacts .....	166
Flow Objects .....	167
Schema diagram .....	168
Schema Context Tab .....	169
Schema table .....	170
Manage Data .....	172
Query Editor .....	172
Organization diagram .....	173
Organisation Context Tab .....	174
Human resource .....	175
Repository users and usergroups .....	176
LDAP import .....	176
Assigning roles .....	176
Process map .....	178
Send element .....	178
Process element .....	179
Receive element .....	180
Forms .....	181
General Element Properties .....	183
Heading .....	183
Text .....	184
RichText .....	184
Textbox .....	185
Textarea .....	187
Dropdown .....	189
Checkbox .....	191
Radiobutton .....	192
Datepicker .....	193
DateRangePicker .....	194
File Upload .....	196
Hyperlink .....	197
Button .....	198
Group .....	199

Page Break .....	201
Data Grid .....	202
HTML Control .....	206
QR-Code .....	212
Validation .....	212
Multi Language Diagrams .....	213
Edit Multilanguage Term .....	213
Change Current Language .....	214
Edit Document Languages .....	215
Change Document Language .....	218
The Property Concept .....	218
Expression Editor .....	219
Condition Editor .....	221
Feedback Editor .....	221
Message Editor .....	222
Message Map Editor .....	223
Web Service Editor .....	224
Participant Editor .....	224
Stakeholder related Diagram Views .....	229
Example .....	230
Create View .....	231
Open / Close View .....	232
The Views Toolbox .....	233
View Dependent Edits .....	233
Publishing and Deployment .....	234
Publish .....	235
Manual Deployment .....	239
Workflow Deployment .....	240
Business Process Simulation .....	241
Process Flow animation .....	242
Start flow animation .....	242
Stop and pause flow animation .....	243
Step / Fast forward .....	244
Step / Fast backward .....	244
Result visualisation .....	244
Normal .....	244
Color by split / join elements .....	245
Path to the selected element .....	245
Timeline .....	245
Animation Toolbox .....	246
Business Process Simulation & Analysis .....	247
Starting Simulation .....	248
General Simulation Settings .....	248
Random Distributions .....	249
List of Stochastic Distributions .....	249
Stochastic Distribution Editor .....	252
Stochastics Seeds .....	254
Activity's Duration and Time-based Events .....	254
Stochastic Event Trigger .....	255
Calendar based Event Trigger .....	257
Scripted Event Trigger .....	260
Gateway Decision Making .....	261
Gateways .....	261

Process Variables .....	263
Assigning Process Variables .....	263
Assigning Conditions .....	265
Scope of Variables .....	266
Simulation Context Data .....	266
Additional Information .....	267
Using Probabilities .....	268
Example Models .....	269
Element settings .....	269
Events .....	270
Linking Events to Each Other .....	270
Start Events .....	270
Intermediate Events .....	271
End Events .....	272
Activities .....	272
Subprocess Activities .....	273
Flows .....	273
Sequence Flows .....	273
Message Flows .....	274
Default & Conditional Flows .....	275
Data Associations .....	275
Pools & Lanes .....	276
Data and Artifacts .....	276
Data Object .....	277
Data Store .....	277
Advanced Concepts .....	277
Resources .....	277
Costing .....	278
Declaring cost types .....	279
Neutral costs .....	280
Activity-based costs .....	281
Resource costs .....	282
Products .....	283
Result .....	285
Experiment Results .....	286
Experiment Results Perspective .....	286
Histograms .....	287
Counters .....	287
Sankey Style .....	288
Overlay Options .....	288
Model for Process, Activity and Event Details .....	289
Process Details .....	290
Activity Details .....	292
Event Details .....	293
Model for Queue statistics .....	294
Queue statistics .....	296
Model for Gateway and Resource details .....	298
Gateway Details .....	301
Resource Details .....	301
Simulation Tutorial .....	303
1. Processes and Sequence Flow .....	304
Model 1a - Processes & Activities .....	304
Starting the Simulation .....	307

Report .....	309
Model 1b - Time Consuming Sequence Flows .....	312
Report .....	313
Model 1c - Time Instants for Start Events .....	313
Report .....	315
Model 1d - Scripted Time Instant for Start Events .....	316
Report .....	317
2. Gateways & Process Variables .....	317
Model 2 - Parallel Gateways .....	318
Report .....	318
Model 3a - Exclusive Gateways using Conditions .....	319
Report .....	320
Model 3b - Assignment of Stochastic Process Variables .....	321
Report .....	323
Model 3c - Inclusive Gateways .....	324
Report .....	325
Model 3d - Data Objects .....	326
Report .....	327
Model 3e - Data Stores .....	328
Report .....	329
Model 3f - Exclusive Gateway with Probabilities .....	330
Report .....	331
Model 3g - Inclusive Gateway with Probabilities .....	332
Report .....	333
3. Messages & Signals .....	334
Model 4a - Message Exchange & Waiting Queues .....	334
Report .....	335
Model 4b - Transporting Process Variables via Messages .....	337
Report .....	338
Model 4c - Messages without an explicit Edge .....	339
Report .....	340
Model 5 - Sending Signals to Multiple Process Instances .....	341
Report .....	342
Model 6 - Triggering Start Events .....	343
Report .....	344
4. Extended Sequence Concepts .....	345
Model 7a - Multiple Sequential Activity .....	345
Report .....	346
Model 7b - Multiple Looping Activity .....	346
Report .....	347
Model 7c - Multiple Parallel Activity .....	348
Report .....	348
Model 8 - Link Event instead of Sequence Flows .....	349
Report .....	350
5. Sub-Processes .....	351
Model 9 - Activity Sub-Processes .....	351
Report .....	352
Model 10 - Attached Error Event .....	353
Report .....	354
Model 11 - Events attached to Sub-Processes .....	356
Report .....	358
6. Scripting .....	359
Model 12 - Scripted Delay .....	359

Report .....	362
7. Resources .....	363
Model 13 - Simple Example .....	364
Report .....	366
Resource Shift Plan .....	367
Model 14 - Resources for Pools and Lanes .....	370
Report .....	371
Prioritization of Resources .....	373
Model 15 - Setup and Post-processing time .....	375
Report .....	377
Model 16 - Basic Load .....	377
Report .....	380
8. Activity Based Costing .....	380
Model 17 .....	380
Report .....	385
Business Process Automation .....	387
Define Workflows .....	387
Process Instance Variables .....	388
Predefined Objects .....	389
Session .....	389
Workflow .....	390
UserTask .....	390
Token .....	391
TokenAttribute .....	392
ProcessInstance .....	393
User .....	395
Users .....	396
Role .....	396
Roles .....	396
Group .....	397
Groups .....	397
DocumentationPackage .....	398
LibraryElement .....	398
Diagram .....	399
DiagramControl .....	399
Database .....	399
ScriptContext .....	400
ExecutionContext .....	401
Application .....	401
DataSet .....	402
DataSets .....	402
Define Participants .....	403
Declare Start Form .....	403
Providing Form Data .....	403
Declare Workflow Preview .....	404
Workflow Property Page .....	405
Build a Workflow .....	407
Language Extensions .....	409
Working with Scripts .....	409
Using toast/pop-up notifications .....	410
Proceed .....	411
Working with multiple task instances .....	413
Multiple script task instances .....	414

Multiple user task instances .....	415
Multiple subprocess task instances .....	417
Calling web services .....	422
Special web service functions .....	423
Processing XML replies .....	423
Using adapters .....	424
Activate adapters .....	424
Predefined adapters .....	424
CSV .....	424
Excel / SpreadsheetLight .....	426
Google Calendar .....	427
OpenEDM (YOPRO PLM) .....	429
PDF .....	429
PowerShell .....	432
RemoteExec .....	433
SAP .....	433
SharePoint .....	436
Word .....	437
Creating adapters .....	438
Execute Workflows .....	438
Run a Workflow .....	439
Run a Workflow by Timer .....	440
Monitor Process Instance .....	442
Inspect finished Process Instances .....	449
Configure Participants .....	450
Best Practice .....	451
Feedback after Form Submission .....	451
Attach Message Event to Hyperlink .....	452
Attach Email Notification to a User Task .....	454
Referencing data in Notifications .....	455
Attach User Task to Hyperlink .....	458
Trigger message programmatically .....	459
Start Workflow outside YOPRO .....	463
Define waiting time .....	465
Sample Documentation Release Workflow .....	466
Create a BPI Project .....	466
Check Form .....	467
Release Documentation Process .....	467
Check Documentation Activity .....	468
Decline Activity .....	471
Release Activity .....	471
Decision Path 1 .....	472
Decision Path 2 .....	472
Build Workflow .....	473
Monitor Workflow Queues .....	473
Customization .....	475
Attribute Extensions .....	475
Customer Settings .....	477
Custom Shapes .....	481
Integration of the web portal into applications .....	483
Embedding Forms in another application .....	485
Pre Submit Example .....	487
Export and Import .....	489

Export .....	489
Export to the Filesystem .....	489
Export as Bitmap .....	490
Export as Word Document .....	490
Export as HTML site .....	491
Export as RACI Diagram .....	493
Export IT System Diagram .....	495
Export as SVG Image .....	497
Import .....	497
Import from Filesystem .....	497
YOPRO Api Developer Guide .....	498
EventArgs classes .....	498
BackendEventArgs class .....	498
CreateWorkflowFinishedEventArgs class .....	498
DeleteInstanceCompletedEventArgs .....	499
GetAccountCompletedEventArgs .....	499
GetDownloadUrlCompletedEventEventArgs .....	500
GetProjectTasksFinishedEventArgs class .....	500
LoadAttributesCompletedEventArgs class .....	500
LoginCompletedEventEventArgs class .....	501
ProcessContinuedEventArgs class .....	501
ProcessInstanceLoadedEventEventArgs class .....	501
ProcessInstanceListLoadedEventEventArgs class .....	502
ProcessInstancesLoadedEventArgs class .....	502
ProcessStartedEventArgs class .....	502
ReadLogCompletedEventArgs class .....	503
TasksLoadedEventArgs class .....	503
TaskUpdatedEventArgs class .....	503
WorkflowFormLoadedEventEventArgs class .....	504
WorkflowLoadedEventEventArgs class .....	504
WorkflowsLoadedEventEventArgs class .....	505
Session class .....	505
Workflows class .....	507
Workflow class .....	507
ProcessInstances class .....	509
ProcessInstance class .....	510
ProcessInstanceAttributes class .....	512
ProcessInstanceAttribute class .....	512
ProcessInstanceFile class .....	512
ProcessInstanceState enum .....	514
Tasks class .....	514
Task class .....	515
Roles class .....	516
Role class .....	517
RoleMember class .....	517
RoleUser class .....	517
RoleEmail class .....	518
TailoredActivities class .....	518
TailoredActivity class .....	518
ProjectTasks class .....	519
ProjectTask class .....	519
FAQ and troubleshooting .....	521
Install Digital Pen .....	521

Legacy topics .....	526
Firefox security warning .....	526
Delete Silverlight Isolated Storage .....	526
Uninstall OOB Installation .....	527
Unable to create / join collaboration session .....	528
System management .....	530
System management functions .....	530
User management .....	530
Synchronization with a directory service .....	530
Reset a users security id .....	531
Delete user .....	531
Trigger license information update .....	531
LDAP synchronization troubleshooting .....	531
Connection error during user synchronization .....	532
Invalid directory entry .....	532
Account already in use .....	532
Account name change to account in use .....	533
No enabled user .....	533
Group not found .....	533
Connection error during group synchronization .....	533
Group member load error .....	534
Loading groups for synchronization failed .....	534
Connection error during group member synchronization .....	534
Group assignment not possible (no YOPRO user) .....	535
Group assignment not possible (user not in team) .....	535
Team user list empty .....	535
Configured team id is invalid .....	535
Configured team id not found .....	535
Conversion of user information failed .....	536
User enabled property is empty .....	536
User enabled property has wrong type .....	536
User enabled property is missing .....	536
Synchronization aborted .....	536
On Premises Configuration .....	537
E-mails .....	537
Delegation .....	537
Substitution .....	538
Workflow data access .....	540
Workflow version name .....	540
System-specific attributes .....	540
Default Customer Settings .....	540
Color Schemes .....	541
Use internal portal .....	542
System Performance Logging .....	542
Check the service .....	542

## Welcome to IYOPRO

---



Business Process Management (BPM) and its targeted utilization leads to a definite increase of the efficiency and flexibility of the business wide IT-driven processes. The design and documentation of business processes is the derivation and kernel of Business Process Management and provides more transparency of your internal and external processes. Until today complex process diagrams are challenging regarding their different levels of observation abstraction. IYOPRO accepts this challenge and supports it with an intuitive and easy-to-use user interface within the modeling, workflow and simulation tool.

IYOPRO is an online BPM Suite (BPMS) and available as SaaS-Application (Software as a Service) or in-house. IYOPRO supports the BPMN 2.0 standard and provides a very innovative user-interface, multi-level process diagrams (from process landscape to primary- and sub-processes) and the collaboration work within the process design.

IYOPRO is not only a tool for process modeling but also a tool for task management and human workflows all on a single user interface.

Intellivate has made efforts to ensure the accuracy and completeness of the information in this document. However, intellivate disclaims all representations, warranties and conditions, whether express or implied, arising by statute, operation of law, usage of trade, course of dealing or otherwise, with respect to the information contained herein.

Intellivate assumes no liability to any party for any loss or damage, whether direct, indirect, incidental, consequential, special or exemplary, with respect to  
(a) the information; and/or  
(b) the evaluation,  
application or use of any product or service described herein.

Intellivate disclaims any and all representation that its products or services infringe upon any existing or future intellectual property rights. Intellivate owns and retains all right, title and interest in and to the intellivate intellectual property, including without limitation, its patents, marks, copyrights and technology associated with the intellivate services. No title or ownership of any of the foregoing is granted or otherwise transferred hereunder. Intellivate reserves the right to make changes to any information herein without further notice.

## Getting started with IYOPRO

### New to IYOPRO

- Read the [Introduction](#) section to learn more about IYOPRO, its different editions and system requirements.
- Follow the [Quick Start Guides](#) to familiarize yourself with the processes of managing your business processes.

### Regular user of older IYOPRO versions

- Read the [What's new in IYOPRO](#) section to have a quick look at major changes.
- Run through the [Quick Start Guides](#) to familiarize yourself with the new version.

# Introduction

---

[About IYOPRO](#)

[System requirements](#)

[Getting help](#)

[IYOPRO editions](#)

[IYOPRO terms and conditions](#)

[What's new in IYOPRO](#)

[How to buy IYOPRO](#)

## About IYOPRO

IYOPRO is an easy to use yet powerful and intuitive business process management suite (BPMS). IYOPRO provides a clear and efficient user interface to build the most amazing process diagrams, interactive WEB based process documentation, Word documents from a single source and high resolution Bitmap images for Presentation purposes.

In addition IYOPRO offers full Workflow capabilities and business process analysis in one tool and on one integrated data model.

Forget about bloated user interfaces, incomprehensible tools and BPEL exports and imports. IYOPRO has been engineered to provide the most advanced functionalities in their simplest form: creating and maintaining business processes. Diagramming is usually a painful process but thanks to IYOPRO you may surprise yourself enjoying it!

You know how to use your favorite BPM suite, so you already know how to use IYOPRO: it's that easy! IYOPRO is fully WYSIWYG (What You See Is What You Get) and you'll begin to imagine how fast and easy it will be for you to create your next process documentation and how professional it will look like.

## System requirements

IYOPRO's recommended system configuration:

## IYOPRO as Software as a Service

### Client

If you are working with a Windows 10 Client or a newer version, you have all the necessary requirements. Just get started with the IYOPRO BPM Suite: <https://www.intellivate.com/en/bpm-kostenlos-testen/>

## IYOPRO as an on-premises installation

The technical requirements for an in-house installation are as follows:

### Client

If you are working with a Windows 10 Client or a newer version, you have all the necessary requirements. Just get started with the IYOPRO BPM Suite. Ask your system administrator for details.

### Server

#### Hardware

- Can also be available as a virtual machine
- CPU: Quad Core I7 / Xeon or comparable - 64 bit
- Memory: >= 16 GByte
- Storage:
  - OS partition (Operating System): > 100 GB
  - IYOPRO partition (application, database, files): > 100 GB

#### Software

- Microsoft Windows Server 2016 English 64 bit (or newer)
- IIS 10 (Internet Information Server), included in the operating system

- MS SQL Server 2016 or newer
- .Net 4.5

### Miscellaneous

- Access to internal SMTP server for delivery of emails
- VPN access for support
- File system backup
- Internet access (<https://www.iyopro.com/iyopro>)

## Getting help

This help file can either be viewed on-line or off-line when installed with IYOPRO. You can obtain the latest version as well as other formats of this help file on line at <https://www.iyopro.de/pages/de/training-und-support.html>

### Off-line access

To view the IYOPRO documentation off-line please visit <https://www.iyopro.de/pages/de/training-und-support.html> and download the the off-line help files provided.

### On-line access

To access and view the most recent IYOPRO's help file on-line, launch a web browser to the following URL: <https://www.iyopro.com/pages/en/training-and-support/online-documentation.html>

### Printing the help file

Alternatively, you can download and print a PDF or CHM version of IYOPRO's documentation from the following URL: <https://www.iyopro.com/pages/en/training-and-support.html>

## IYOPRO editions

Five editions of IYOPRO are available based on your needs:

- [IYOPRO Premium Edition](#): Fully functional licensed edition, which provides modeling and documentation as well as workflow authoring and analysis features
- [IYOPRO Professional Edition](#): Limited functional licensed edition, which provides modeling and documentation as well as workflow participant features
- [IYOPRO Workflow Edition](#): Limited functional licensed edition, which provides reading documentation as well as workflow participant features
- [IYOPRO Portal Edition](#): Limited functional licensed edition, which provides reading documentation features
- [IYOPRO Basic Edition](#): This edition is also called Starter Edition which provides modeling and documentation features.

### IYOPRO Premium Edition

- Can be used for Business Process Analysis requirements
- Offers full workflow capabilities ( author and execute), involving tasks to none IYOPRO users
- Offers customization capabilities for teams and organizations
- Contains all features of the Professional Edition
- Available as SaaS and On Premise

### IYOPRO Professional Edition

- Offers executive workflow capabilities, involving tasks to IYOPRO users
- Contains all features of the Basic Edition

- Offers more documentation features, i.e. Export formats
- Available as SaaS and On Premise

### **IYOPRO Workflow Edition**

- Offers executive workflow capabilities, involving tasks to IYOPRO users
- Offers access to the released process documentation.
- Provides limited functionality i.e. comments
- Available as SaaS and On Premise

### **IYOPRO Portal Edition**

- Offers access to the released process documentation
- Provides limited functionality i.e. comments
- Available as SaaS and On Premise

### **IYOPRO Basic Edition**

- Can be used immediately without registration
- Registered users have private data space to store process data
- Available as SaaS

If you wish to have further information how to buy IYOPRO please follow this [link](#).

## **IYOPRO terms and conditions**

### **General Terms of Contract**

General Terms of Contract for the SaaS Offer ("SaaS Conditions") of Intellivate GmbH Süderstrasse 282 D-20537 Hamburg Last revised: January 2012 As a provider, INTELLIVATE permits the USER to utilize service provisions (use of application services and "software application") at the following SaaS conditions:

#### **1. General**

- 1.1 The "SaaS conditions" of INTELLIVATE shall apply exclusively; INTELLIVATE shall not recognize any conditions of the USER which are contrary to or at variance with the "SaaS conditions" of INTELLIVATE, unless INTELLIVATE has expressly agreed to their application in writing. The "SaaS conditions" of INTELLIVATE shall also apply if INTELLIVATE performs the service for the USER without reservation in awareness of conditions of the USER which are contrary to or at variance with the "SaaS conditions".
- 1.2 Any agreements which have been entered into between INTELLIVATE and the USER for the purpose of the execution of this contract have been set down in writing in this contract.
- 1.3 INTELLIVATE reserves the right to change these "SaaS conditions". If the "SaaS conditions" are changed, INTELLIVATE shall advise the USER of the changes to the "SaaS conditions" in writing or by electronic means. The USER shall thereupon have the option of objecting to the changes within 6 weeks. In the case of an objection, the previous "SaaS conditions" shall continue to apply. The USER shall receive clear notice of the changes; the changes shall be indicated in concrete terms.

#### **2. Object of the contract, description of service**

- 2.1 The object of the contract is to grant the USER the possibility of using of the software IYOPRO of INTELLIVATE via Internet as Software as a Service (SaaS). INTELLIVATE has stored the software application on its in-house server and shall make it available for retrieval by the USER for the term of this contract. The USER may use it to process his data.
- 2.2 Insofar as any copyright interests of INTELLIVATE as a provider or of third parties are affected, the USER shall be granted a simple, non-transferable right of use, which is time-limited to the period of this contract (simple license). Sublicenses may not be issued.

- 2.3 Certain services (e.g. relocation and bereavement check) may only be used after separate contracts have been signed and the general terms of contract of third parties accepted by the USER.
- 2.4 The access data for identification and authentication required for use of the software application shall normally be sent to the USER by INTELLIVATE via e-mail within five working days after conclusion of contract.
- 2.5 The software application may only be used by the user stated in the SaaS contract. Employees or other persons authorized by the user are also entitled to use the software application in accordance with the contractual use.

### 3. Performance of the contract

- 3.1 INTELLIVATE shall normally allow the USER to use the application services seven days a week (24 hours). The time needed for the requisite regular maintenance and updating and the technical improvement of hardware and software (scheduled "downtime") shall be excepted, as shall be cases as defined in clause 13 below (force majeure and other impediments to performance which are not the responsibility of INTELLIVATE). The scheduled "downtimes" have already been taken into account in the calculation of the payment; a reduction in the contractually owed payment on account of scheduled "downtimes" shall not come into question.
- 3.2 The availability of the software application shall be 98 % per annum. Scheduled "downtimes" and/or "downtimes" agreed with the USER shall not be included in the availability calculation. 3.3 INTELLIVATE shall inform the USER of an interruption of availability on account of scheduled "downtimes" in due time in advance (<https://www.iyopro.com> )
- 3.3 Accounts set up for test purposes can be blocked by INTELLIVATE at any time without a period of notice being necessary.

### 4. Further developments/changes to the specifications

- 4.1 INTELLIVATE reserves the right to make changes to the specifications (e.g. through the use of newer or different technologies, systems, processes or standards) in the course of technical progress and for performance optimization after the conclusion of contract. Any major changes to the specifications shall be advised by INTELLIVATE in good time on <https://www.iyopro.com> . If significant disadvantages arise for the USER as a result of the changes to the specifications, the USER shall be entitled to terminate this contract for exceptional reasons as of the date of alteration. The USER must terminate the contract within two weeks of the notification of the changes to the specifications.
- 4.2 INTELLIVATE may contract out the performance of the application services to third parties in whole or in part. As far as is necessary, INTELLIVATE may make documents, information and data of the USER accessible to third parties, to whom INTELLIVATE has legitimately contracted out services. INTELLIVATE is entitled to change the Internet infrastructure used as well as the third parties commissioned with the performance at any time.
- 4.3 As a general principle, Intellivate may terminate individual services of the software application with a period of notice of 1 year.

### 5. Service provisions

- 5.1 INTELLIVATE shall set up a hotline. This shall be available to answer inquiries about problems which concern the use of the software application during normal office hours (currently on workdays from Monday - Friday between 8:00 am and 5:00 pm Greenwich Mean Time + 1). INTELLIVATE shall provide application-related information. Application-related support extending beyond directly answerable inquiries shall not be provided. Inquiries to the hotline shall be made exclusively via the respective contact form in the SaaS portal.

### 6. Obligations to cooperate/responsibility of the user

- 6.1 The obligations to cooperate which are incumbent upon the USER on his own responsibility and which are required for effective provision of the application services include the following in particular:

- The creation of the prerequisites for use of the application services;
- Measures for protection of his data and programs (back-up), especially the regular creation of backup copies in line with the risk
- The use of virus protection programs on the in-house computer in the current version.

The USER shall basically assume sole responsibility for

- the selection of the software application and the results he thereby intends to achieve;
- information and data which originates from him;
- the devices used by him (hardware and software) and their suitability for data communication with INTELLIVATE;
- the functional capability of his Internet access including the transmission paths.

## 7. Contractual obligations of the user

- 7.1 The USER agrees to notify INTELLIVATE immediately after becoming aware of a defect in the software application. The USER shall follow the instructions given by INTELLIVATE with regard to the description, localization, determination and reporting of errors. The USER shall precisely define his defect reports and questions to the best of his ability and call on qualified personnel to do this where required.
- 7.2 The USER agrees to inform INTELLIVATE immediately if there is a change of person (inheritance or other universal succession), address, name, legal form or company.

## 8. Access data

- 8.1 The USER shall keep his access data secret from unauthorized persons, store it carefully and securely from the access of unauthorized persons and protect it from misuse and loss.
- 8.2 In principle, the USER shall be responsible for the payment of all fees for the use of the application services via his access identification. If the USER raises an objection against his obligation to pay on account of alleged unauthorized use via his access identification, and if unauthorized third parties acting outside the sphere of influence of the USER can be excluded, he shall not be obliged for payment only if he furnishes proof that there was unauthorized use of the application services via his access identification which was outside his responsibility.

## 9. Payment, term of contract

- 9.1 INTELLIVATE shall levy a payment for use of the software application ("SaaS charge") in accordance with the SaaS contract.
- 9.2 The contract shall commence after unlocking and have an unlimited term. The period of notice for an ordinary notice of termination shall be 1 months before the end of the contractual year. The notice of termination must be in writing and reach us no later than the third working day of the first month of the period of notice.
- 9.3 INTELLIVATE is entitled to adjust the SaaS charge to the current price list. In particular, INTELLIVATE is entitled to adjust the SaaS charges as appropriate for application services for which INTELLIVATE relies on data supplied by the respective postal services or other data providers in the event of changes to the suppliers' prices.
- 9.4 The USER shall only have rights of off-set if his counterclaims have been declared legally binding, or are undisputed or acknowledged. The USER shall only be entitled to exercise a right of retention to the extent that his counterclaim is based on the same contractual relationship.

## 10. Claims for defects

- 10.1 As a rule, the USER is familiar with the software application and its productive capacity. The USER has the opportunity to test the software application for his own purposes in a test phase before concluding

this contract. If no significant complaint was made by the USER during a preceding test phase, both contractual parties shall assume that the software application is basically as provided in the contract. As a general principle, a deficiency of the software application is only provided if use of the software application adversely affects the USER to an unreasonable extent.

- 10.2 The USER is aware that INTELLIVATE does not operate its own network and therefore does not make Internet access available to the USER. For this reason, INTELLIVATE assumes no responsibility for the serviceability of the respective Internet access.
- 10.3 The liability of INTELLIVATE shall not extend to deficiencies caused by departures from the conditions of use specified by INTELLIVATE.
- 10.4 INTELLIVATE relies on data provided by the respective postal service as well as other carefully selected data providers for certain service provisions and is therefore dependent on the completeness, up-to-datedness and the thereby associated correctness of this basic data. This also applies to availability and updating. In this respect, INTELLIVATE therefore assumes no guarantee that an erroneous or incomplete address is corrected and updated as part of the address validation.
- 10.5 If the functions of the software application differ from those which are contractually assumed and/or have deficiencies, the USER shall advise of this without delay.
- 10.6 The USER may only reduce the payment on account of defects through deduction from the agreed payment if his claim is undisputed or acknowledged by INTELLIVATE or has been declared legally binding.
- 10.7 A right to terminate this contract for exceptional reasons on account of defects shall only come into consideration if the continuation of the contractual relationship is unacceptable, or a breach of a condition which goes to the root of the contract persists in spite of a notice of reprimand and the setting of a time limit. A notice of reprimand shall not be required if the breach of a condition which goes to the root of the contract is so serious that a notice of reprimand is not deemed to be sufficient to end the breach of the condition and/or to re-establish trust. INTELLIVATE is normally entitled to make two attempts to rectify the respective deficiency before the contract can be terminated for exceptional reasons.

## 11. Liability

- 11.1 Liability on account of culpable injury of life, body or health shall not be affected; this shall also apply to the statutory liability under the German Product Liability Act.
- 11.2 Otherwise INTELLIVATE shall be liable without limitation for willful intent and gross negligence, also of its legal representatives and managerial staff. For the fault (willful intent, gross and slight negligence) of other agents, liability shall be limited to the order amount without VAT as well as to such foreseeable damage which is typical for the utilization of a software application of this nature.
- 11.3 INTELLIVATE shall not be liable for slight negligence, also of its legal representatives, executive staff and other agents, unless an obligation is breached whose compliance is of particular importance for meeting the purpose of the contract (cardinal obligation). In the case of a slightly negligent breach of a cardinal obligation by legal representatives or managerial staff of INTELLIVATE, the limitation of liability as defined in the preceding clause 11.2, second sentence, shall apply accordingly.
- 11.4 INTELLIVATE shall only be liable for the loss of data and its recovery as provided in clause 11.1 to 11.3 if such a loss could not have been avoided through the data security measures incumbent upon the USER.
- 11.5 Liability regardless of negligence or fault for errors which are already in existence on conclusion of contract is expressly excluded.

## 12. Data protection

- 12.1 INTELLIVATE shall provide the application services for the USER as a contract data processor pursuant to Section 11 of the Federal Data Protection Act. The responsibility for the legitimacy of the processing and utilization as well as the protection of the rights of the people concerned (information, correction, blocking, deletion) shall remain with the USER.
- 12.2 INTELLIVATE shall comply with the statutory regulations for data protection, especially the Teleservices Data Protection Act and the Federal Data Protection Act.
- 12.3 The USER acknowledges that he shall provide the British data protection officer or any other authorized British enforcement authority with information concerning questions of data protection without delay if

British addresses of the Royal Mail are processed.

### 13. Impediments to performance which are not the responsibility of Intellivate

13.1 Except in cases in which INTELLIVATE has accepted a procurement risk or a guarantee by way of exception, failures or delays of performance as a result of the following circumstances and impediments to performance shall not be the responsibility of INTELLIVATE:

- Circumstances of force majeure as well as impediments to performance
- occurring after conclusion of the contract or becoming known to INTELLIVATE through no fault of its own only after conclusion of the contract and
- with regard to which INTELLIVATE can prove that they could not be foreseen and prevented even by due diligence of INTELLIVATE.

The aforesaid prerequisites - commencement or discovery only after conclusion of contract through no fault of its own, unpredictability and unavoidability proven by INTELLIVATE – shall include in particular: legitimate forms of industrial action (strike and lock-out); disruptions of operations; power failure (subject to standard fallback and emergency power supply systems); disruptions and attacks from the Internet. INTELLIVATE shall be exempt from the obligation to render services for the duration of these circumstances and impediments to performance. Claims for compensation of the USER shall be excluded in the aforesaid cases.

13.2 In the case of an irrevocable impediment to performance within the meaning of clause 13.1, either contractual party shall be entitled to terminate the contract immediately by termination without notice for exceptional reasons.

13.3 In the case of unreasonable hindrance of performance, INTELLIVATE shall be entitled to terminate the contract without notice for exceptional reasons.

13.4 If use of the software application is not possible for more than five workdays in succession, the USER shall have the right to terminate the contract without notice for exceptional reasons.

13.5 The right of both contractual parties to terminate the contract without notice for exceptional reasons for any other important reason shall remain unaffected.

### 14. Court of jurisdiction, applicable law, e-mail transmission, proviso clause

14.1 If the USER is a merchant pursuant to the Code of Commercial Law, a body corporate organized under public law or a special authority under public law, the court of jurisdiction for all obligations arising from the contractual relationship - also for matters regarding bills of exchange and cheques - shall be the registered place of business of the company of INTELLIVATE or, at the choice of INTELLIVATE, also the registered place of business of the USER. The above agreement conferring jurisdiction shall also apply to users with a registered place of business outside Germany.

14.2 All the rights and obligations arising from and in connection with the contractual relationship existing between INTELLIVATE and the USER shall be exclusively governed by the laws of the Federal Republic of Germany.

14.3 As a general principle, any notifications and declarations of one contractual party which are required under the terms of the contractual regulations or become necessary in the further course of business can be effectually sent to the online address of the other contractual party. Unless otherwise provided in the contract, they shall be regarded as delivered on the workday following their receipt. If any provision in this contract should be or become ineffective, the effectiveness of the remaining provisions or agreements shall not be affected hereby

### What's new in IYOPRO

This section provides information about the latest releases of IYOPRO.

#### v1.5.6.7

#### IYOPRO v1.5.6.7

## Organisation Diagram

You may define your organisation for documentation purposes and enhance it with roles for the workflow execution. Users and groups may be imported from the IYOPRO Repository or even simpler from your LDAP Directory which requires IYOPRO to run Out of Browser.

## Schema Diagram

Define your database schema which will be linked to BPMN Datastores to load and save your process data while executing workflows. You may export and import schema content to give you most flexibility to manage your data.

## Form Editor

The form editor now allows multicolumn forms. There is also a new data grid control. Several controls have been enhanced to let you create more professional forms.

## BPI Projects

The functionality has been enhanced significantly. The project tree currently is the only object, which got a context menu. Diagrams and forms within the project tree may be dragged onto other diagrams to create subprocesses, datastores, usertasks or link them to existing elements. We will provide a sample video soon, which will explain the power of the IYOPRO Project Management.

## Documentation export

IYOPRO now generates a detailed Process documentation with a sophisticated layout. There will be more with our next releases.

## Bitmap Export

The quality of the Bitmap export for large diagrams has been enhanced dramatically. But - export may take a while

License management - you may now flexible assign and deassign single licenses from your license package to your users. We will provide a video to explain the license management.

## Hyperlink in text

You may insert hyperlinks in your texts. BPMN Element Styling - with this release we start to deliver different styles for the BPMN stencils. For now we provide a sketch style which gives your diagrams the effect of a hand drawn concept paper.

## Validation

With this release we changed the way of diagram validation. You now have a better control of what will be marked. Also there is a new Simulation validation which will indicate missing must have parameters for a successful analysis step.

## Workflow Portal

For professional users we provide a new workflow portal. The portal will present tasks, monitored processes and runnable workflows in a clearly arranged way.

### v1.6.4.9

## IYOPRO v1.6.4.9

### Advanced Hyperlinks

They now allow references not only to web pages but also to local files and repository files.

### HTMP Project export

This export type is introduced with this release. It allows you to export your process documentation as web pages which can be accessed from your intranet.

## v1.7.4.9

### IYOPRO v1.7.4.9

#### Forms

HTML-Forms were extended by authentication and Ajax support. Our job is not yet complete. With the provided web services you may now embed forms into every web portal and start a workflow instance in IYOPRO when the user submits the form.

#### Workflow Portal

You'll find the portal functionality within the repository now.

#### Hyperlinks

There is a new option "Workflow Message" which allows to trigger a BPMN 2.0 message event if the user clicks on the hyperlink i.e. from within an email.

#### Validation

Some rules regarding the BPMN 2.0 validation have been corrected to confirm the specification on some special cases of events and activities.

#### BPI Projects

If a project contains multiple modified files and one of these files will be saved a save management dialog will appear that allows to save all or a selection of the files.

#### OpenXML Export

The Preview now contains the table of contents.

#### HTML Project Export

The configuration dialog now contains an Advanced tab. Here you can adjust the order of the tree elements. You may also specify a home page which will initially be displayed. The Shortcut feature allows you to create a hyperlink pane from which the user may access specific documents very quickly. The search feature has been moved to the navigation pane. You may save the advanced settings if you save the project.

#### Schema Diagram

With this version we introduce the schema relation object. It'll be used to link table columns together.

## v1.8.5.2

### IYOPRO v1.8.5.2

#### Versioning

IYOPRO diagrams may now be versioned. To use this click on "Save new version" in the "Save As" menu. To load an old version click on the arrow of the splitted OK-Button in the file selection dialog. You may then choose from a list of available file versions of the selected object. In addition you'll find a Properties entry in the file menu. The properties dialog contains a Version tab which provides the required information.

#### Office 2010 themes

IYOPRO now provides not only the Office 2007 themes but also the Office 2010 themes.

#### Watermark for basic Users

IYOPRO Basic users will find a "Powered by IYOPRO Basic Edition" watermark on all image exports. That was a requirement of our sales partners.

## **IYOPRO Api**

With this release we provide our first version of the IYOPRO Api. With this development library it's easy to integrate desktop applications into IYOPRO cloud workflows.

### **v1.8.9.0**

## **IYOPRO v1.8.9.0**

### **BPS Simulation**

Our Simulation module received significant enhancements. The distribution dialog now provides a preview of the selected mathematical distribution including the distributed values based on the given parameters. In addition the report provides not only the mathematical distribution but for comparison the selected values of the simulation pass. This provides clearness if the selected simulation values are representative or not.

While calculating the simulation pass a progress bar is shown as well as a calendar which shows the current simulated date.

The Report has been divided into sections which may be expanded or collapsed. Each section contains a short help which explains the content of the section.

### **Modelling**

It is now possible to create RACI, RASCI, RACI-VS or VARISC as well as CAIRO or RACIO Diagrams. Because the diagrams only differ by the used responsibilities we simply use the term RACI. To use it add your responsibilities to participants in the participant dialog of your user tasks. Then use

File->Export->RACI Diagram

Verify the appearing preview. You may then export the diagram to Excel format.

### **Workflow**

The SaaS back end workflow engine sandbox is now allowed to execute web services. That offers very new possibilities for workflow integration which was until now excepted to On Premise installations.

### **Word Export**

To reduce the amount of generated pages the Word export received a new setting where you may control the insertion of page breaks.

### **Basic Edition Ads**

ONLY the free IYOPRO Basic Edition will now show business advertisings of partners and customers on the right side. If you're interested to see your ad here please contact us. If you wish to have an Ad free IYOPRO update at least to the Professional Edition.

### **v1.9.4.1**

## **IYOPRO v1.9.4.1**

### **Workflow**

The execution performance of python scripts while processing workflow's has been significantly improved.

### **Workflow**

The repository now allows to explicitly delete process instances

### **Workflow**

The users web form now contains a ribbon interface. The participant which will perform a task must now accept the task which is then no longer assigned to the complete group / role.

## Modeling + Workflow

You may now define the web service definition of a service task within the "Service Definition Property Dialog". This dialog makes it easy to attach and test most web services. In the background the dialog creates a python script which will be executed then. Experienced users may modify this code to perform more complex operations.

## Simulation

You may now save simulation reports and reopen them later.

## Simulation

You may now specify complex repetition timestamps within our new recurrent date editor.

## Modeling

IYOPRO now supports user defined Attributes (UDA). UDAs let you define attributes for every type of shape in IYOPRO. You may then apply one or even more UDA-definitions, which will be merged together in the UDA-Dialog.

## Modeling

IYOPRO now supports user shape templates. Shape templates let you define default settings of shapes. Currently supported is the control of the default color and the default description.

## Modeling

As with the process map shapes it is now possible to change the shape color of all shape types individually over all diagrams. The new "Transfer" command gives the user an simple and effective way to populate font- and color settings to multiple other elements.

## Workflow Scripts

IYOPRO now supports the document type "Script". This will allow the integration of program code into Workflow-projects in a more powerful way as before.

## Documentation

The documentation has been extended. Especially the Simulation module has been described in detail including report examples.

## v1.9.8.1

### IYOPRO v1.9.8.1

#### Userinterface

IYOPRO now supports Dutch and French language beside the already existing English and German languages.

#### Documentation

The word export now supports hyperlinks as expected.

#### Documentation

You can now configure, which activities in which order are shown in the word export.

#### Modelling

The BPMN export and import has been validated against several systems.

#### Simulation - Animation

When reaching a Timer Intermediate Event the flow will pause until the control is selected. This allows to wait controlled.

## Workflow

The verification, deployment and printout of digital pen enterprise-forms is now integrated into the IYOPRO SaaS Version. These forms may be created with IYOPRO's form editor or imported from any third party app which is able to save AcroForm PDF's. This simplifies the usage of digital pens significant.

### v1.10.7.4

## IYOPRO v1.10.7.4

### Userinterface

IYOPRO now supports Russian and Brazilian Portuguese language beside the already existing English, German, Dutch and French languages.

### Modelling

The content of the shape toolboxes may now be controlled by customer settings. Due to different customer settings it's possible to provide different toolboxes to different modeler roles (i.e. beginner / advanced or designer / workflow architect)

### Modelling

The description template of all documents and shapes may now be controlled by customer settings.

### Modelling

The direction of process map arrows may now be changed.

### Documentation

The Word export may now list activities, events, documents and gateways.

### Documentation

The Word export will now incorporate a custom logo and a custom title from the customer settings.

### Documentation

The HTML export may now list activities, events, documents and gateways.

### Documentation

The HTML export will now incorporate a custom logo, a custom headline and a custom background color from the customer settings.

### Workflow

Messages can now be triggered from email backlinks to message flows and receiving message events

### Workflow

The datastore has been extended to support a repository directory to store process instance documents.

### Repository

The way to add users to teams has been revised. With this release we introduce the Team Access Code (TAC). With this temporary code multiple users may join the team without further permissions. This is perfect for education where students need to join an instructors team.

### v1.11.8.0

## IYOPRO v1.11.8.0

### Modelling

To visualize the relationship between activities and IT-Systems we created the new IT-System artifact. BPMN 2.0 lacks this shape but explicitly allows custom shapes.

### Modelling

IYOPRO now supports multi language diagrams. Many languages have been predefined. However, if you miss one please let us know.

### Modelling

With this release we introduce stakeholder related views of process diagrams. This provides an easy way to publish tailored diagrams to any type of consumer within an integrated data model.

### Documentation

Based on the new IT-System shape IYOPRO now provides a new IT-System Diagram report. This report shows which diagram / activity / role utilizes which IT-System.

### Documentation

The Word Export now reports assigned user defined attributes

### Forms

The forms now support a Hyperlink Control. It's possible to link to static url's as well as dynamic url's which are defined in process instance attributes.

### Workflow

The process instance log data and process instance attributes may be easily downloaded for running and finished process instances in Excel format. This allows KPI analysis with third party systems

### Simulation

The simulation has been enhanced to support activity based costing. This feature is very powerful. We're working on a training to familiarize you with the required steps.

### Simulation

The simulation now supports the event based gateway

### Simulation

If there are multiple activities executing in parallel which require the same resources you may now specify the priority at each activity to execute first.

### Simulation

Beside stochastic distributions process instances may now start based on calendars.

## v1.13.3.5

## IYOPRO v1.13.3.5

### Forms

The forms now support a QR-Code Control. QR-Codes will be printed on on paper driven forms. The user may then fill out the form with a digital pen.

With a special app on a mobile device the user may then scan the QR-code and attach photos to the instance.

### Forms

The DataGrid object has been enhanced to support different column types.

## Workflow

The advanced properties editor now supports the insertion of attribute bindings to messages and scripts via mouse click.

## Workflow

IYOPRO now supports customer developed interfaces (i.e. .net library) for access in script-tasks (on Premise only)

Several adapters have already been developed and will become available for our customers (i.e. an RFC-SAP Adapter)

## Repository

The repository now has back and forward buttons for navigation

## Repository

The file search has been enhanced. You may now search for terms at elements i.e. the name or description of controls.

## Repository

Users of the basic edition will be allowed to fully participate in teams for a trial period of 30 days. After that team-folders will be read only for basic users.

## Modelling

Several people asked for a setting to control the pop up behavior of the modeling toolbar. In the user settings you may now choose between Touch (the "old" behavior), Select or None.

## Simulation

you may now define shift plans for every worker in your organization diagram, to restrict the availability of your resources. Several standard shift plans are provided and can be modified to your demands.

## Simulation

It's now possible to restrict the time slots, when stochastic start events may start, e.g. in order to model opening hours when customers may arrive.

## Simulation

Activity-based costing has been improved. Use more possibilities for allocating neutral costs. Measure when neutral cost centers are utilized, for allocating them to products or processes.

## Simulation

You can now define your cost centers and products within your organization diagrams. This avoids typing errors and eases modeling.

## Simulation

The basic load describes working time of your resource, which is not modeled within your processes, but prevents usage of the resource in your processes. The setup time is the time a resource needs to prepare working in an activity, the post-processing time is the time a resource needs after working in an activity before it changes its state back to idle.

## Simulation

Performance improvements. Simulation runs faster and needs less memory space.

## Simulation

With this release we introduce the new Asymmetric Normal Distribution. This new stochastic distribution is similar to the normal distribution, but its standard deviation to left may be different to the standard deviation to the right.

## Simulation

Instead of conditions, you can now also enter probabilities at the outgoing sequence flows of exclusive and inclusive Gateways.

## Simulation

You can now change the corresponding product within your processes with help of python scripts.  
SimulationContext.Product="Produkt1";

## Documentation

The HTML-Export of Collaboration Diagrams now includes important simulation properties, like duration or inter arrival times, which eases the validation of your simulation models!

### v1.16.0.1

## IYOPRO v1.16.0.1

## Portal

With this release we introduce the first version of our new IYOPRO portal. The IYOPRO portal solution is a set of features which combines publishing and auditing processes and result in a very simple end user interface to utilize published and approved business process documentations along their life-cycles.

## Workflow

In conjunction with enhancements to the ProcessInstance-object in Python and our MS-Word adapter it's now possible to fill out word documents within a script task.

## Workflow

Send Message End Events may now trigger Receive Message Start Events within the same BPI-Project without need of webservice round trips. Data may be exchanged very easy due to the message map feature.

## Workflow

It's now possible to get a list of users that are authorized to participate at the workflow within python. This can be used to programmatically forward a user task to another person.

## Workflow

IYOPRO now allows the use of centralized python scripts within BPI projects. These can be referenced by expressions and extend the functions of an expression.

## Forms

The form editor has been revised. With this release we introduce the new group control. This allows not only visual grouping of form elements but also splitting a column into multiple sub columns.

## Repository

You can now rearrange objects in the process explorer via drag & drop. This helps very much in the organization of BPI Projects.

## Repository

Existing projects may now be linked to multiple solutions.

## Modeling

You can now edit the data flow relationship of the Data Association. Possible options are undirected, unidirectional and bidirectional.

## Simulation

The new simulation-property "Use Stochastic Duration" on SubProcess-Activities allows to prescind the linked sub-process.

## Simulation

The association of products and cost centers to simulation properties is now easier and avoids mistakes. Just choose the items defined in your organization diagram with help of the drop-down-boxes.

## Simulation

The fractional number representation has been enhanced in simulation report for a better overview.

## Simulation

Some bugs within simulation report were fixed. Thereby performance of report visualization was improved and problems with persisting large reports were solved.

## Simulation

When you have errors in your simulation run (e.g. because of inaccurate python scripts), the error log provides the IYOPRO element IDs for a better traceability.

## Simulation

The user interface of the stochastic distribution editor has been enhanced. Now you have separated input boxes for days, hours, minutes and seconds.

## Simulation

Long running simulations may now be aborted by a button. The experiment report will be provided for the time span between simulation start and abortion.

## Simulation

You can now use the data type "Dictionary" within in your python scripts.

## Simulation

BPMN shapes (e.g. IT-Systems), which have no simulation impact will be ignored. You do not need to remove them before simulation anymore.

## v1.19.0.4

### IYOPRO v1.19.0.4

## Portal

The root node of the process documentation now supports a news section. It visualizes information about new or revised processes as well as new or edited comments

## Portal

IYOPRO now supports to subscribe process models. Users with subscriptions receive notes in the news section of the process portal.

## Portal

A new search feature has been implemented to search for specific processes within the portal.

## Report

The IT-System diagram report now supports data stores as well as it systems.

## Report

The RACI diagram now includes pools and lanes to resolve participants.

## Modeling

You can now assign tags to diagrams for classification purposes. Within the search function of the repository you can find diagrams with a given tag.

## Repository

It's now possible to open a description pane in the files browser. It displays the description of the selected file.

## Repository

There are two new main nodes in the repository: Process Documentation as the root of the deployed process documentation of your organization. Library is the root of all library elements of your teams.

## Userinterface

Depending on the used browser IYOPRO now supports F1-key for on-line help and Ctrl-S for save.

## Userinterface

The upper right help icon now shows a pop-up menu to navigate to the online documentation or to send a feedback.

## Modeling

Links between elements (i.e. sequence flows) can now be swapped by a tool-bar button.

## Modeling

Process map input, process and output elements now contain a vertical label position attribute to control the vertical alignment

## Modeling

With this release we introduce the library feature. The library allows you to store previously defined activities out of your process models for later reuse in an organized way. Reused activities also updates on all diagrams if they are revised. This is especially perfect for automation and workflow processes.

## Simulation

A new view on your collaboration diagrams includes a model-integrated visualization of simulation parameters, which will help you to parametrize your model at all.

## Simulation

Another new view on your collaboration diagrams includes a model-integrated visualization of experiment results parameters, which will help you to understand what happened in your simulation run by give you a first impression on counts and durations.

## Simulation

An extra ribbon-tab 'Simulation results' now includes all buttons corresponding to the results of a simulation experiment.

## Simulation

You can export all of your simulation results as excel file for now, with the button provided in tab 'simulation results'. The file consists out of several sheets, each representing one of the tables from your report included in IYOPRO.

## Simulation

The trace log has been fundamentally redesigned. Use the trace log to record a step-by-step report on executed activities or events of your processes. The trace log now is more readable and ergonomic.

## Simulation

The trace log can be exported in MXML-format, which is readable by several process mining software products. You can enhance your analysis, using features provided by process mining technology.

## Simulation

The sequence probabilities at gateways and the duration of activities can be set product-dependent, so you can set an individual probability or duration for each product which is expected to be processed.

## Simulation

All of the properties which have to be set for a simulation run, can be set with help of a table-based editor. This editor also allows an export as Excel-file of your simulation parameters.

## v1.21.8.5

### IYOPRO v1.21.8.5

#### Simulation

Histograms in experiment result view now contain activity based costing. Furthermore, you can colorize activities depending on their costs.

#### Simulation

From now on, you can access the state of resources (idle, busy) within Python scripts. E.g., this can be used at gateway conditions, so that the process will choose a path if the corresponding resource is free, and the other path if it's not.

#### Simulation

The performance of the simulation run has been significantly improved.

#### Forms

The technology we render workflow forms in the browser has been changed. Also the style has been modified to give the forms a better look and feel. With this new technology we are able to bring more rich features to the webforms in the future.

#### Forms

It's now possible to query details of the workflow state within workflow forms.

#### Modeling

Beside the existing export features we now additionally provide the SVG image export. SVG images have the advantage that they are vector orientated and may be scaled in any direction without quality loss. This is especially for large diagrams very useful.

#### Workflow

Embedded workflow forms now support a pre submit handler to allow the hosting application to do additional checks on the form

#### Workflow

The workflow forms now use the browser language to find out the best language for the web form.

#### Modeling

The library now supports all entity types.

#### Modeling

Within the process explorer you can now open a folder in the repository using the context menu

#### Modeling

Within the process explorer you can now open file properties using the context menu

#### Modeling

Within the process explorer you can now close a solution using the context menu

## Modeling

Shape names and their descriptions are now editable in a tabular description editor to give you a better feeling for completeness. The diagram and the editor are positioned side by side and selections are highlighted in both views so the navigation is very intuitive.

## Portal

The Welcome screen in the portal now works as expected. Workflow tasks, instances and news have been revised to give the user a better experience.

## Userinterface

Within the file properties dialog you can now copy the url of the file to the clipboard. This makes it much easier to reference IYOPRO diagrams out of third party applications or web portals.

## Userinterface

File Properties are now available for all file types

## API

The IYOPRO API now supports the creation and execution of a workflow from a hierarchy of tasks and roles.

## v1.27.4.1

### IYOPRO v1.27.4.1

#### User Interface

Documents can be displayed in a maximized mode.

#### Modeling

Extended Attributes can be appended to diagrams.

#### Modeling

New properties for grid lines, view control and preview functionalities have been added to the preference settings.

#### Modeling

Element names can contain hyperlinks now.

#### Workflow

Substitutes can now be defined to adapt workflow tasks while the actual user is absent.

#### Workflow

Workflow tasks can now be delegated to other users.

#### Workflow

Workflow attributes can be edited in the portal.

#### Workflow

Python constructs "ImportModule" and "Database" are now available in start forms.

#### Workflow

The ID of Workflows is now available via Workflow.Id.

#### Workflow

Users can now be informed about a broken workflow via e-mail.

#### Workflow

Individual batch queues are now available for workflows.

## Forms

Data grids and groups in forms can now be disabled. Furthermore Data grids in forms can now contain drop down lists and textboxes in data grids are now editable.

## Forms

Hints for form elements like text boxes are displayed in the control itself now.

## Forms

New data types have been implemented for data grids: Integer und Float.

## Forms

The usage of variables on form level will direct all dictionary entries to the web client (e.g. for visibility, disablement)

## Forms

Column definitions of data grids can now access form data via {Binding Form:<VarName>}

## Forms

The label of upload controls can now be customized.

## Forms

Date pickers can work without timezones now.

## Repository / Portal

The starting page in the portal can be changed by users now.

## Repository / Portal

The portal will be opened maximized and received a button to close it.

## Repository / Portal

Hyperlinks to specific portal and repository pages can be copied to clipboard now.

## Repository / Portal

The search can be limited to specific folders now.

## Documentation

customer settings and new entities (including entities of choreography and conversation diagrams) are supported now.

## Documentation

The creation of documentations has been accelerated. Documentations are created in the background now, which enables the user to continue working. Users will be informed when their documentation creation is finished.

## Documentation

Release workflows can now contain workflow details. Release workflow instances can now be deleted automatically when their related documentation package has been deleted or overwritten

## Documentation

Diagrams can be displayed maximized now.

## Customer Settings

Appliance of opened customer settings can now be done with every license type and active customer settings can now be revoked.

## Customer Settings

Users will be informed now whenever their active customer settings have been changed (those changes are automatically applied).

### **Word Export**

Document headlines can now be edited by users and extended attributes can be displayed in the footer of word exports.

### **Word Export**

Data stores and IT systems can now be included.

### **Word Export**

Diagram elements can now be sorted automatically by their process sequence.

### **Word Export**

The currently active views of diagrams can now be exported too.

### **HTML Export**

Data stores and IT systems can now be included.

### **Layout**

The standard theme of IYOPRO has been changed.

### **Layout**

The simulation ribbon will now be displayed permanently.

### **On Premise**

Customer settings can be provided centrally.

### **On Premise**

The initial GUI language can be provided centrally.

### **Active Directory Synchronisation**

Deactivated users will now be deactivated in IYOPRO too.

### **Active Directory Synchronisation**

IYOPRO synchronized AD groups which contain sub groups will be parsed recursively.

### **Active Directory Synchronisation**

The SID of users is now available in workflows via Python.

### **General**

Mechanism to lock data for an improved integrity of IYOPRO data has been integrated.

### **v1.30.7.6**

## **IYOPRO v1.30.7.6**

### **User Interface**

IYOPRO is available in polish language now.

### **Repository / Portal**

By pressing a character's key on the keyboard, the next row in a file table can be selected where the file's or folder's name is beginning with this character.

### **Repository / Portal**

The process instance dialog has a button to refresh now.

## Repository / Portal

There are no more empty folders for tasks in the repository tree.

## Repository / Portal

The task list contains the information who has started the process.

## Repository / Portal

The task list contains the current process instance details.

## Repository / Portal

There is a context menu on the tasks list now.

## Repository / Portal

Solutions are part of the recent items now.

## Repository / Portal

The recent items list is expanded to 25 items.

## Repository / Portal

There is a hyperlink to the recent items on the welcome page.

## Repository / Portal

The user list of LDAP synchronized user groups can be scrolled.

## Repository / Portal

Search results don't show double entries for recent items anymore.

## Modeling

There is a user setting for the default diagram language now.

## Modeling

There is a user setting for the default script language now.

## Modeling

The performance of loading and renering diagrams is enhanced.

## Modeling

Visibility and deactivation rule values can be constants now.

## Modeling

The URL editor supports UNC paths now.

## Modeling

The URL editor supports drag 'n' drop and local paths now, if IYOPRO runs out of browser.

## Modeling

Clicking the OK button of the dialog to add elements to projects is only allowed once.

## Workflow

Terminate end events are support in workflows now.

## Workflow

Cancel end events in transaction sub processes are supported in workflows now.

## Workflow

The user's security id (of users synchronized via LDAP) are now accessible from pythons scripts.

## Workflow

The process instances of a workflow are deleted now by deleting the workflow file.

## Forms

The form control's sizes are harmonised.

## Forms

The first error is scrolld into view by turning on the form checking.

## Forms

Double values are supported by drop down menus now.

## Forms

There is no substitute count on start forms anymore.

## Forms

The column type "single" supports different verification rules now.

## Forms

Forms can be cleared now.

## Forms

Drop down controls in data grids are sorted by the displayed value now.

## Forms

The first control is focused automatically.

## Documentation

There is a save dialog - if needed - before the documentation can be created.

## Documentation

The creation of a documentation works independent of the already loaded diagrams. Unsaved changes are not part of the created documentation.

## Documentation

The messages of errors creating a documentation are better.

## Documentation

No empty folders are created anymore.

## Documentation

The visible detail of a diagram is scaled by maximizing the diagram area.

## Documentation

There are no description templates on documentation pages anymore.

## Documentation

Comments can only be edited by its creator.

## Documentation

Comments can only be deleted by its creator or an administrator of the team.

## Word Export

Elements without pool are now grouped and can be orderd like elements part of a pool.

## Word Export

Process maps can be exported now.

## Library

The process explorer's Library-Node is refreshed by deleting a library item.

## Library

A library item can not be opened multiple times as document.

## v1.35.8.6

## IYOPRO v1.35.8.6

## Repository

The query mask of the advanced search can now be cleared.

## Repository

The type filter in the advanced search now contains a predefined list of values.

## Repository

Data and folders can now be filtered in the search results of the advanced search.

## Repository

The repository path of data can now be copied into the clipboard.

## Modeling

A new context menu on document tabs enables you to close single diagrams or all diagrams.

## Modeling

In addition to their name diagrams now have a title property which will replace the diagram name Word and HTML exports if it's set.

## Modeling

Solutions can now be renamed in the process explorer.

## Modeling

The scaling behaviour of BPMN artefacts has been adjusted.

## Modeling

Property changes can now be undone.

## Modeling

Validation of the BPMN rule set has been improved.

## Workflow

User tasks can have a due date now.

## Workflow

Various users can share the same Email address now and still participate a workflow together.

## Workflow

Tasks can now be prioritized in the Task overview.

## Workflow

The list of running process instances will now be refreshed when a new workflow is started.

## Workflow

The process instance details have a new layout.

## Workflow

The list of participants for workflow tasks can now be administrated during run-time.

## Workflow

Workflow tasks will now be created when if no participant could be found.

## Workflow

The binding syntax has been expanded and supports converters.

## Forms

Drop down values in data tables can now be arranged.

## Forms

Buttons in embedded forms can now trigger actions in other systems.

## Forms

Date pickers in forms can now contain dates before 1970.

## Forms

The field focus in forms now supports groups and pages.

## Documentation

A monthly user statistic has been provided in the documentation.

## Documentation

Diagrams in documentations can now be printed.

## Documentation

Diagrams in documentations can now be exported as Word documents.

## Documentation

A search dialog on documentation pages has been added.

## Documentation

Results of the search on diagrams will now now be highlighted in the diagram.

## Documentation

The general and advanced search dialogs can now be accessed from every documentation pages.

## Documentation

News will now provide information about the diagram they are related to.

## Documentation

Comments can only be edited by their creator.

## Library

Empty library folders will not be presented in the process explorer anymore.

## Library

More icons for different BPMN shapes have been added to ensure an easy determination of library elements.

## v1.38.8.3

## IYOPRO v1.38.8.3

## Repository

Recent Items in the repository can be opened via URL.

## Repository/Portal

The structure of current tasks and process instances on the desktop has been improved.

## Repository/Portal

Word exports of diagram views are available in the documentation portal now.

## Modeling

Diagrams can be pinned to the main menu now.

## Modeling

BPMN shapes can be customized now.

## Modeling

Various rules have been added to the model validation.

## IT System Diagram

IT System Diagrams now support bidirectional data associations.

## Organization Diagram

The import of LDAP users and groups allows to filter entries now.

## Organization Diagram

Inherited roles of organization units are now presented.

## Workflow

Finding a process instances via attribute name and value (YOPROSendWorkflowMessage.aspx) supports various data types now.

## Workflow

The task participant configurator supports eMail users now.

## Workflow

Workflow task icons differ now when another participant is already responsible.

## Forms

Groups can have multiline labels now.

### v1.48.8.1

## IYOPRO v1.48.8.1

## Modeling

you can directly save altered diagrams when closing

## Workflow

- Workflows can now be versioned. There are productive as well as development workflows.
- The status of a process instance is recently available as a separate page. Users can include the status external.
- You can repeatedly execute tasks based on collections
- The Timer Start Event supports now the time-based start of processes
- Sub-processes can now be executed in the context of main processes so that only one process

instance is generated

- The IYOPRO-web service is now able to return result codes when sending a message to a process
- It is now possible to process XML when receiving IYOPRO process messages
- Automatic tasks can be suspended to provide computation time for other processes (cooperative multitasking)
- It is now possible to exit at a defined script point and optionally return a value

## Forms

there are now extended possibilities available when defining options of drop down elements

## On-premises installation

- You are now able to incorporate customized translations of the user interface
- Workflow: one can now define individual e-mail templates for delegation as well as for the substitute rule

### v1.51.2.6

## IYOPRO v1.51.2.6

## General

The performance of the database access has been significantly improved.

## Modeling

- When creating a role, language-neutral keywords can be maintained.
- Now you can change roles in your organization diagrams.

## Workflow

- The delegate notification template also allows the subject line to be dynamically assembled using process instance attributes.
- The process designer can now specifically determine which version of a workflow to use for new process instances.
- The workflow file ID is used to start the currently active version of a workflow.
- The support for parallel subprocesses within a shared process instance has been significantly improved. Subprocesses can now receive private process attributes.

Access to process attributes of other subprocesses as well as the main processes is also possible.

- In the condition editor (for example, form visibility), the inequality operator has been added. In addition, the terms have been improved.
- For forms, a rich text element can be used to format texts arbitrarily.
- The condition, whether a form can be sent or not, can be additionally specified by the process designer.
- Turning the page break element can be assisted by script actions.
- Using the alert() function, the script execution can be aborted with a message.

- The pre-delete callback was introduced on the process instance level.

### **v1.54.1.3**

## **IYOPRO v1.54.1.3**

### **Repository**

- We have further developed the import and export of solutions and linked projects for you. You can now use this functionality even more easily. For example, you get more support for links and you can use the import based on the copy function of the Explorer
- We have enhanced the filter of the list of workflow tasks and workflow instances so that you can also filter through configurable columns

### **Modeling**

You now have the option of choosing whether to display the current version or a previous one when using a URL to load a diagram

### **Workflow**

- We present the new HTML control for forms. Design your forms with your own elements. Easily integrate videos, graphics and more
- You can now disable any lines in a data grid
- The process instance ID as response header: Depending on the system configuration, IYOPRO responds with a HTML page containing a description of the call result or structured result information. Additionally to the state, you now have the option to see the process instance ID as response header
- Decide individually whether the submit button of the page break element of a form should be displayed or not
- You can now also use the OnDrop handler for the Textbox element. Simply specify in the associated element script what should happen when you drag text into a Textbox
- IYOPRO now has an adapter for connecting to SharePoint
- You can now configure who has access to instances and forms in addition to the team administrators

### **v1.57.2.0**

## **IYOPRO v1.57.2.0**

### **Release Candidate of the new web client**

### **Workflow**

- You are now able to configure a default name for the version of a workflow
- We expanded your options when using data stores. When referencing files, you can change a relative path to an absolute path with just one click on a button
- Use a script file as a template for an HTML-Control in a form. Simply import a script file that contains the desired HTML code instead of writing the code directly into the HTML control.

### **v1.59.5.8**

## **IYOPRO v1.59.5.8**

### **New Portal**

- You can no longer only display the description of your diagrams in your documentation, but also advanced properties. Whether user-defined attributes, tags or responsibilities, if you have documented them, you can now find them together with your diagrams in the new portal
- The new portal offers you the possibility to manage your user profile, completely detached from the IYOPRO modeling environment
- It is possible for you to expand the new portal with plugins

### **General**

We have extended the key assignment for you so that you can now assign your various licenses specifically. For example, if you have several license keys of the same license type, you can now decide for yourself which license key the user should get his license from

### **v2.1.63.3.6**

## **IYOPRO v2.1.63.3.6**

### **The new IYOPRO desktop client is here!**

You don't want to work with Silverlight anymore because the end of support is imminent? No problem! With the IYOPRO desktop client, we offer you a Silverlight-independent alternative. Of course, that doesn't mean the original client is gone.

### **Portal**

- Set your surface colors. For example, change your color scheme from light to dark theme or easily adjust the surface colors individually using the color sliders
- In the workflow lists, you now have more options for sorting and grouping your content
- You can now also display the development workflows and not just the active workflows in the list of runnable workflows

### **Workflow**

- There are new design functions in the form, such as a maximum width of controls and the

possibility of giving individual controls a dynamic height

- With the new DateRangePicker control for the forms, you no longer only have the option of selecting a single date, but also a date range
- An initial priority can now be specified via the properties or the script of a user task
- If text fields with actions have been defined in your forms, these are now visualized using a button and can also be executed with it
- There is now an embedded view for tasks and instances that can be opened, for example, via links in e-mails
- It is now possible to use user databases in callback functions

#### v2.1.66.4.4

### IYOPRO v2.1.66.4.4

#### Connection of IYOPRO Web and App

A particular highlight is the new connection between the IYOPRO web portal and the IYOPRO app. Open links either in the web portal or in the app and easily switch from the diagrams displayed in the web portal to the app.

#### Web Portal

- You can now display files from the repository in the Web Portal and open solutions and projects in a process explorer
- Undock your forms and have them displayed in a separate browser tab
- Show your finished instances filtered. Decide for yourself whether you want to display them e.g. since last week, last month or a specific date
- You can now also change the participants of a workflow task in the instance details in the portal
- Do you have a lot of open tabs and don't feel like closing them individually? No problem! Simply use the small tab menu for this
- Create links for diagrams of the repository or for documentation pages with the Copy URL button or the key combination Ctrl + C
- Simply open your diagram in the IYOPRO app by clicking on the IYOPRO symbol in the gray menu bar
- Get to know the extended window management and, for example, place your task list in the middle like a diagram page. Your desired position will of course be remembered for the next time.

#### Workflow

- Define the default substitute for someone else: The default team substitute has not yet been specified for an employee, he is now on vacation and one of his tasks cannot wait? You now have the option of specifying a substitute for someone else in the repository under the team's user node
- There is now an API user for using the API in Python. The API user is entered in the workflow build

options and authorizes the use of the API within the workflow

- The process instance GUID as response header: Depending on the system configuration, IYOPRO responds with a HTML page containing a description of the call result or structured result information. Additionally to the state, you now have the option to see the process instance GUID as response header

## v2.1.69.0.9

### IYOPRO v2.1.69.0.9

#### Web Portal

- In the process instance details, you can now sort the attributes and get an even easier overview of your process instance data
- Now you can easily empty the filter of the workflow lists using the X
- Open a referenced documentation page not only in the same tab as usual, but in a new one instead. To do this, you can easily select an element and use the toolbar that appears, or you can use the shortcut Ctrl + click
- Switch between the standard menu bar and a more compact view of the menu bar with just a button click
- Get more information on the welcome page. See which teams you haven't assigned substitutes for, whether colleagues are absent today, and when you'll be absent next

#### Workflow

- Specify one or more test users in the Workflow Build Options of a workflow who are allowed to see the development version of that workflow in the runnable list in the web portal
- In the data table there is the possibility to add an action. The user can simply click on the button displayed in the data table row and the defined action will be carried out
- Use ScriptContext to define case distinctions for workflow expressions depending on the given workflow setting, point of execution and user actions
- Monitor your workflow queues under the technical options of the process instance list. In addition, depending on your privileges to perform system administration functions, you can disable or enable the processing of jobs. You can also see whether IYOPRO is in normal service mode or in recovery mode
- You can now use an extended version of the alert() method. Personalize your alert by defining your own title, a different icon and a timeout in addition to the message

#### General

- Are you looking for a specific user or group in your team in the repository? No problem. We simplified your search by allowing you to easily filter for what you are looking for under the user or group node
- The portal buttons in the modeling environment now open the new web portal by default. If you would like the buttons to continue to open the internal portal, you can either define this company-wide in the configuration or only for yourself in the preferences
- In AD synchronisation, there are now more flexible options for selecting the source

### **v2.1.69.8.1**

## **IYOPRO v2.1.69.8.1**

### **Web Portal**

- You want the workflow lists to be narrower, but then you can't always read the entire title? No problem, from now on you can simply display the tooltip for truncated texts in the lists
- It is now even easier for you to identify tasks that you are supposed to do on behalf of someone else. On the one hand, this is indicated by a different icon in the task list. On the other hand, you will receive the corresponding information in the form and can easily take over the task
- If you navigate via the folders or if you have a diagram open, you will now see the associated breadcrumbs at the bottom, which show you the folder path followed
- Activate the new feature of the panels and use the automatic size adjustment from now on. Depending on the focus, the width of a panel, such as the task list or the diagram description, is changed in such a way that either this panel or, as usual, the diagram area is displayed larger

### **General**

- There is now a new column "Last Login" in the license key assignment, which should further support you as a license manager. In addition, the filter in the license assignment has been extended for you
- If you have a license key that is not yet known on your on-premises system, you no longer have to wait for the periodic update that gets information about available licenses or start a manual update. If a license key is not known when it is entered, it is now directly and automatically requested from the server

### **v2.1.73.5.4**

## **IYOPRO v2.1.73.5.4**

### **Web Portal**

- The release candidate of the mobile view for editing forms is here!
- If you manage task participants using the process instance details, you can now directly see whether and who has already accepted a task
- Now you can easily maximize your documents using the "Maximize" button in the view ribbon
- Personalize your IYOPRO web even further by placing your logo in the footer or changing the font
- The GUI options are saved in the user profile: No matter which PC or browser you use, you will get the view you have set

## Workflow

- Do you want to inform users about something in the context of task processing, but don't want to have to show and hide a form element? No problem. You now have the opportunity to achieve exactly this by using the notify() method in an expression
- IYOPRO can now also support the latest IronPython version 3.4. If you would like to switch from Python 2 to 3, please contact us

## General

- Importing a solution has been simplified in that the bin file of a workflow is no longer part of the import
- We have put together new snippets for you. You can find the insert button when you can add a Python expression to an element
- Want to check if your service is running normally? No problem. The functionality for monitoring the system has been expanded. The new information can be evaluated automatically
- There are now extended system performance logging options

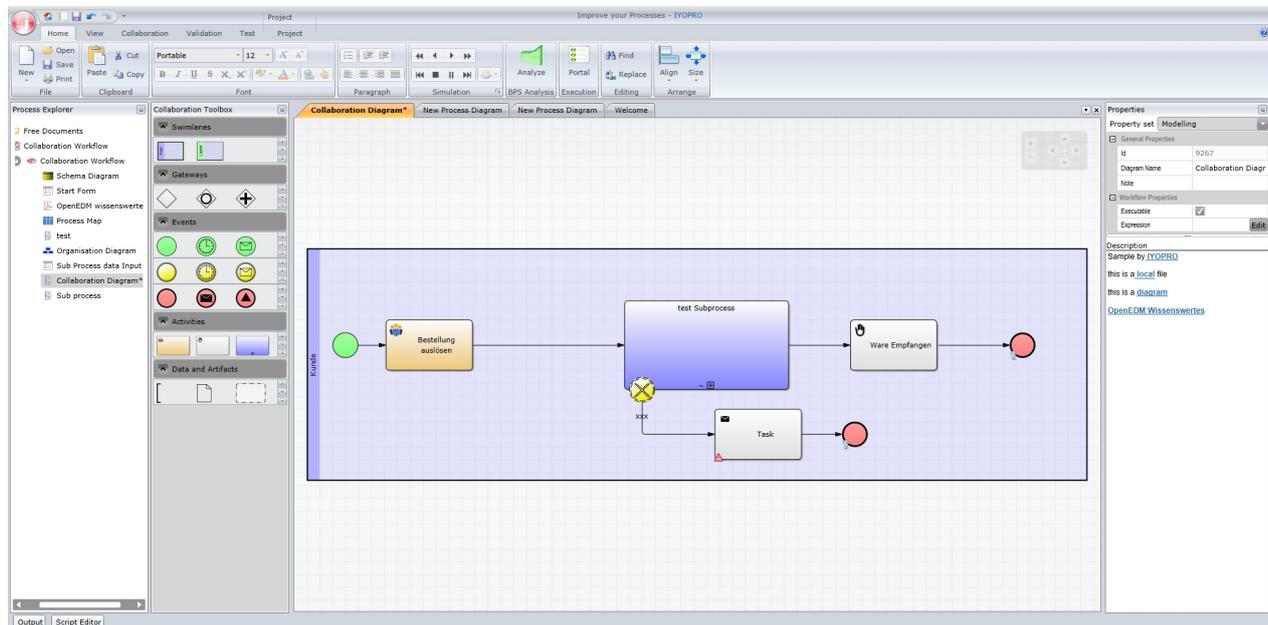
## How to buy IYOPRO

IYOPRO can be purchased worldwide, either online or offline, and paid using various payment methods (Credit Cards, Check, PayPal...) and currencies (US Dollars, Euros...). As soon as the transaction is complete, you will receive the license key of IYOPRO.

To get more information on the order process and purchase IYOPRO, launch your web-browser to the IYOPRO store page at <https://www.iyopro.com>

# Overview of the user interface

## Overview of the user interface



### 1. Application menu

- Manage documents and projects: create new, open existing, save...
- Export and Import documents from / to various formats
- Access document properties
- Access the Repository

### 2. Quick access tool-bar

- Access to frequently used actions such as "Save project", "Undo" and "Redo"

### 3. Ribbon tool-bar

- Contains all possible actions available within IYOPRO
- Can be minimized to provide greater documentation editing screen estate

### 4. Document Work Area

- Makes the content of documents visible and editable
- Provides Zoom, Pan and Preview options

### 5. Process Explorer

- Contains all loaded documents of loaded process models in their nested structure

### 6. Toolbox Area

- Provides all stencils needed to design the various diagram types

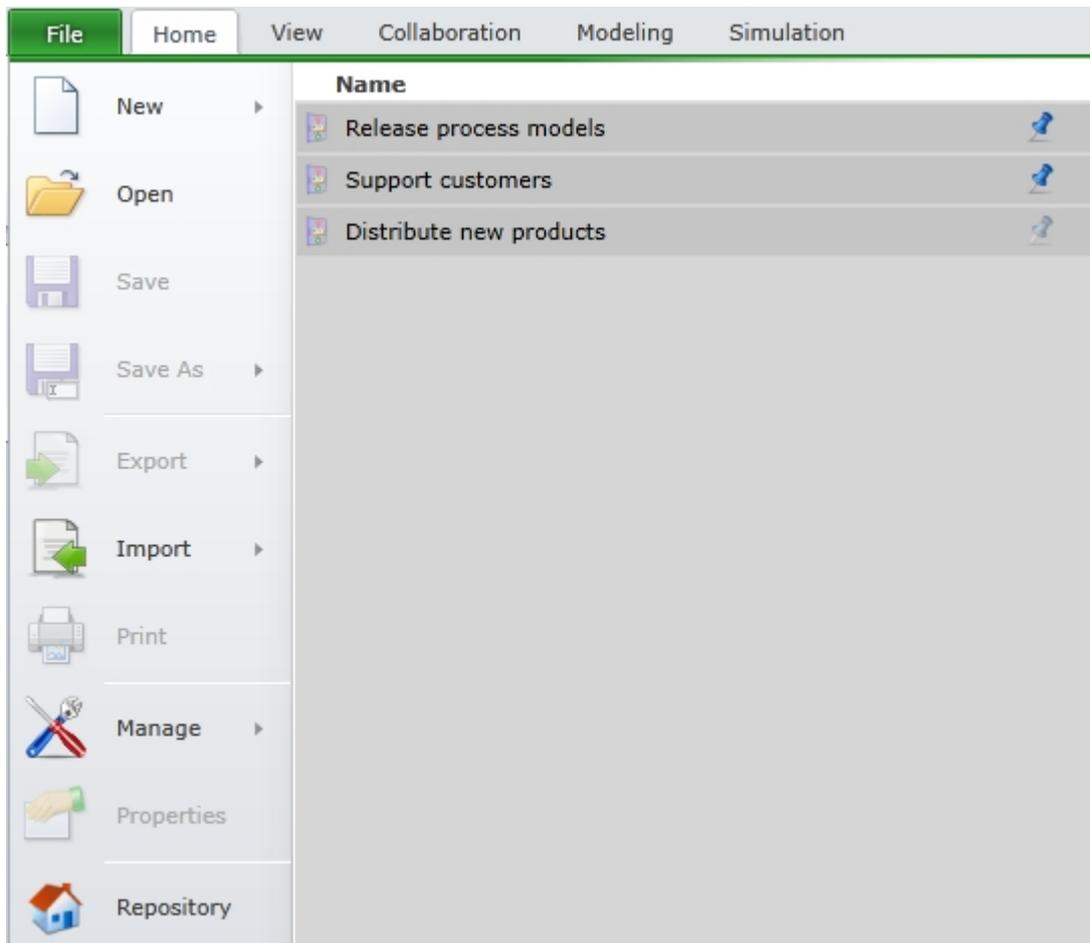
## 7. Property Area

- Provides an interface for browsing and editing properties of the selected element

## 8. Advanced Properties

- Provides an interface for browsing and editing advanced properties which require extensive options.

## Application menu



The YOPRO's application menu appears if the top left button of the main window is activated. It allows to:

- Create / open documents / projects
- "Save" or "Save as"
- View recently opened files and pin/unpin them as favorites
- Export data into various formats
- Import data from various formats
- Print
- Manage settings and defaults
- Access document properties
- Access to the YOPRO repository

## Quick access toolbar

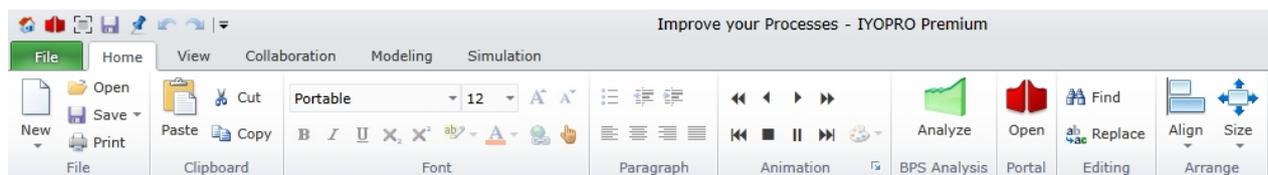


The quick access tool-bar provides quick access to frequently used operations.

These are:

- Repository  
Open the repository browser
- Portal  
Open the portal
- Save  
Save the active diagram
- Pin - Unpin  
Pin or unpin a diagram as favorite in [application menu](#)
- Undo  
Undo the latest operation on the active diagram
- redo  
Redo the last undo operation

## Ribbon tabs



## Presentation

IYOPRO's ribbon tabs are located at the top of the main window and provide all the features available within IYOPRO in a categorized fashion. The ribbon tabs parts are:

1. The main tabs - They are always visible and are used for the most important actions
2. The tabs groups - When a tab is selected, it will display actions grouped by similar purpose
3. Contextual tabs - Those tabs are only shown when needed. For example, the "Portal" tab is only visible when the Workflow Portal is active.

## The Home tab

This tab provides access to the basic actions:

- Create new, open, save and print documents
- Handle clipboard operations
- Manage font and paragraph properties
- Manage the runtime animation feature
- Access simulation & analysis as well as portal modules
- Manage arrange and editing operations

## The View tab

This tab gives access to actions needed to manage the view and style of your diagrams:

- Zoom, Pan, Fit
- Magnify portions of large models
- Manage visibility of editor elements like grid lines, preview, lane path...
- Change themes

- Change stencil styles
- Change the user interface language

### The Collaboration tab

This tab provides access to collaboration features:

- Create or join a collaboration session
- Enable voice and/or video conference
- Toggle show / View mode

### The Modeling tab

This tab provides access to Modeling settings:

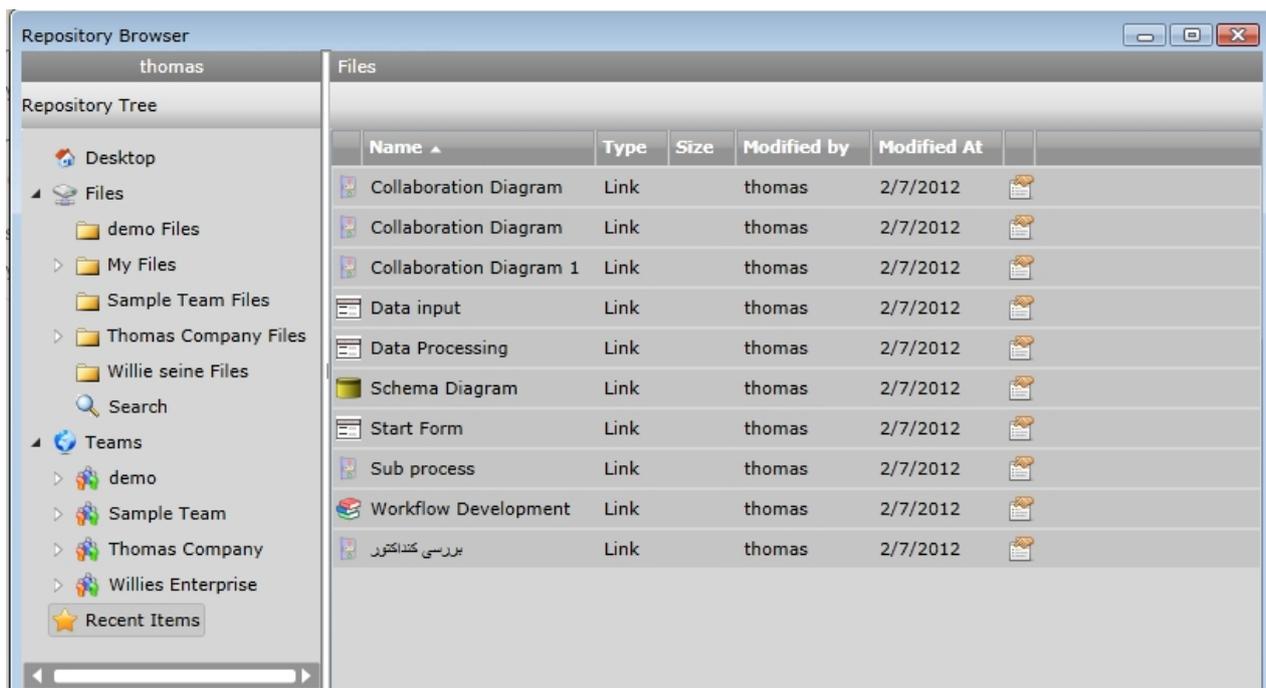
- Handle BPMN 2.0 validation
- Handle process analysis validation
- Create and Apply Attribute extensions
- Transfer Settings from one to multiple other elements
- Control diagram languages
- Perform view dependent edits
- Publish documentation to the IYOPRO Portal

### The Simulation tab

This tab provides access to the actions that enable you to run a simulation:

- Run simulation
- Set start and stop time
- Generate stochastic seed
- Enter the parameter for the simulation
- Change perspective from modelling to simulation...

### The Repository

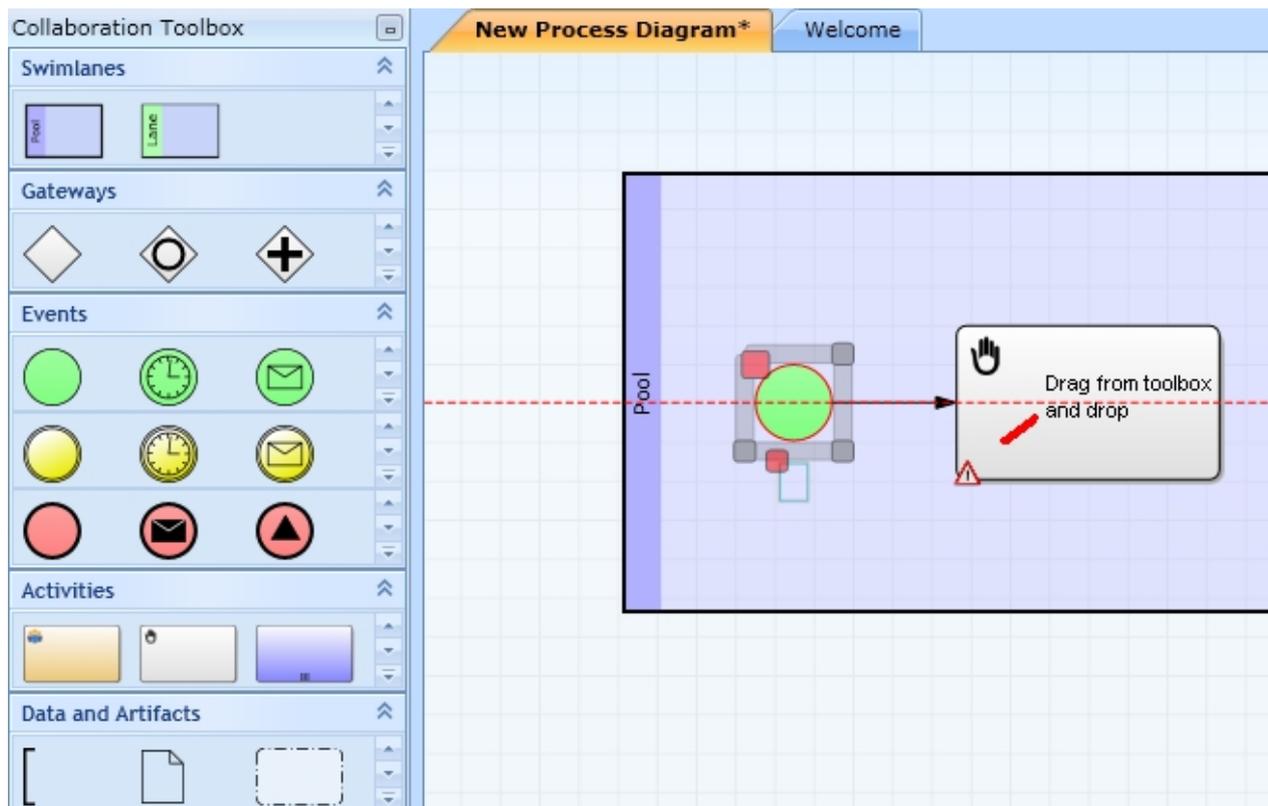


The repository is used to manage your cloud environment. You may access the Repository from the [Application menu](#). The Repository window contains the following elements:

- A Desktop node. Here you can find an overview of your current session as well as options to customize your user profile and your preferences.
- A Files node which embeds your private "My Files" folder and one folder for each team you're member of.
- A Teams node which shows all your teams you're member of. Each team consists of an unlimited number of users and user-groups. These team members may share files via the team folder
- A Process Documentation node which is the root of the deployed and published process documentation of your organization.
- A Library node which contains all library elements of all teams you're member of.
- A Workflow node to access available workflows, monitor running process instances and perform open tasks.
- A Recent Items list which gives you quick access to the last files you've accessed.

## The Toolbox Area

From within the Toolbox Area you may choose the stencils required to draw your diagram. In case of a collaboration diagram, all stencils according to BPMN 2.0 are provided.



To drag an element from the toolbox to your diagram:

1. Select an element out of the toolbox using the mouse and keep the mouse button pressed
2. Drag the item to the desired position in your diagram and release the mouse button

You may wish to increase the width of the toolbox.

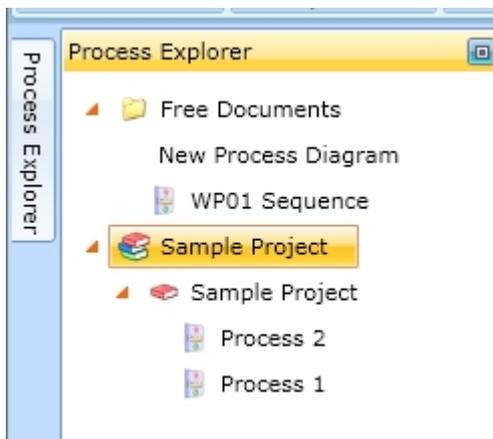


To do so.

1. Point to the border between the toolbox and the diagram until the double arrow appears
2. Press the mouse button
3. Drag the window border to the right and release the mouse button

## The Process Explorer

The Process Explorer assists in managing documents and projects.



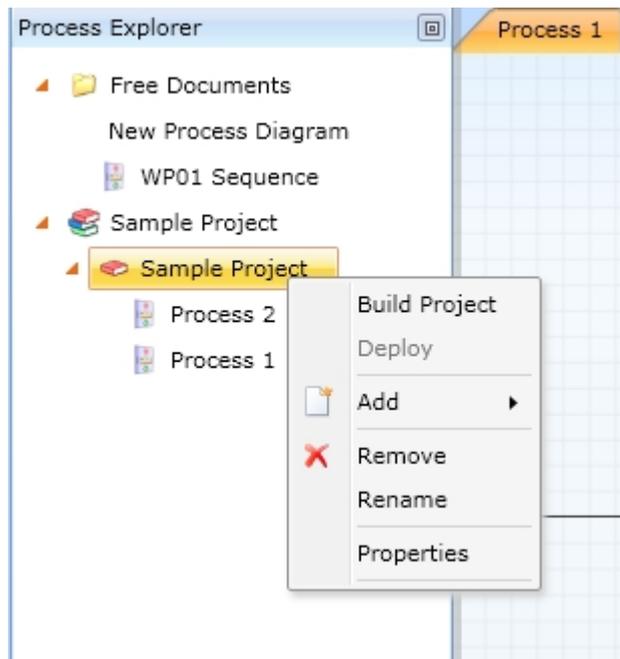
You may expand and collapse the elements in the Process Explorer by either choosing the arrows on the left or double click onto the item.

### Free Documents folder

The "Free Documents" folder contains all documents, which are not part of a BPI project. All other elements on the top level are BPI project solutions, which contain projects and their corresponding documents.

### Contextual menus

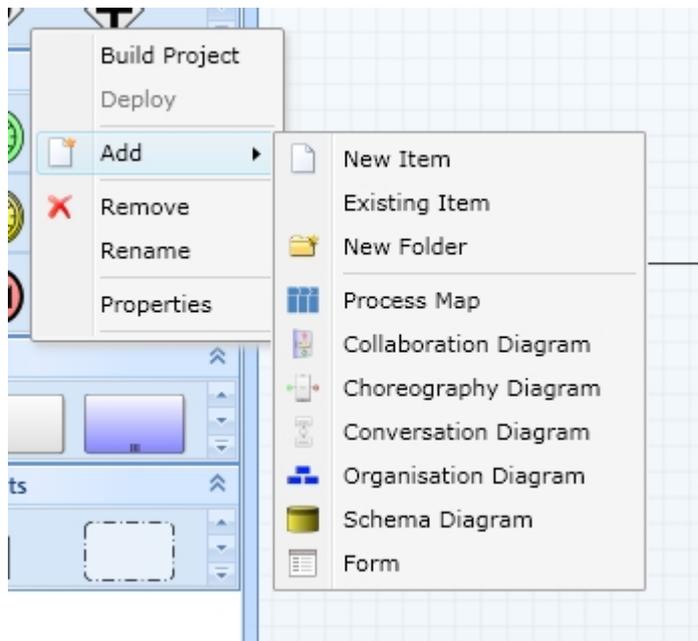
To organize the data within your project the project items provide a context menu.



In case of the project item the context menu allows to:

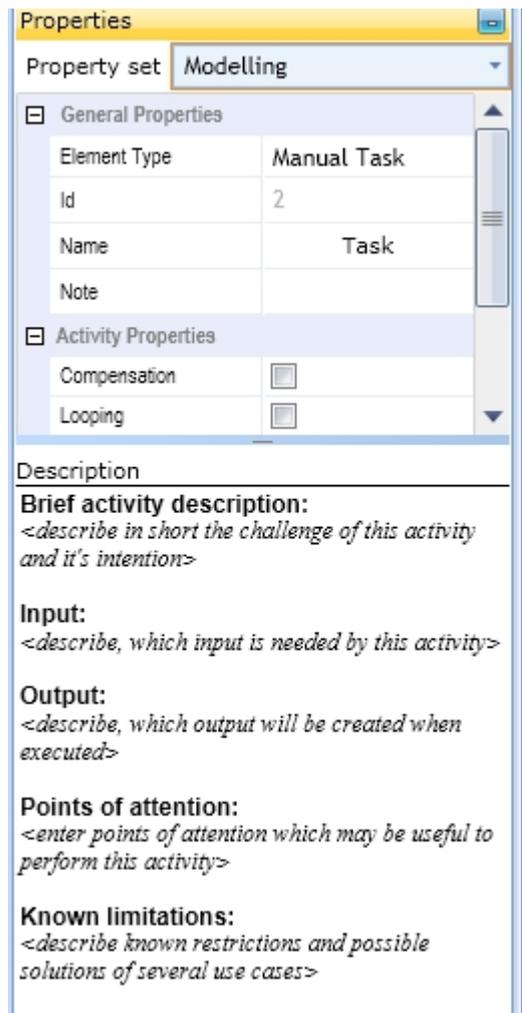
- Build Project - Create a workflow out of the project
- Deploy - Publicize the workflow
- Add - Add new project item
- Remove - Remove item
- Rename - Rename item
- Properties - Edit project properties

Some options in the context menu may contain sub menus.



## The Properties Area

In the Properties Area the characteristics of the active element are shown and can be modified. The example shows the properties of a task.



The Properties Area is divided into three sections:

- Property set selector
- Property grid
- Description

## Property set selector

From the Property set selector, you may choose between:

- Modeling - This property set displays the most usable properties in business process modeling
- Simulation - This property set displays the most usable properties in business process simulation

The separation of the properties became necessary due to the huge number of properties, which can be assigned and modified.

## Property grid

Properties consist out of a name and a value. In the property grid the name is shown on the left, the value on the right. The value can be from the following types:

- Check box

- Selection
- Text (simple or rich text)
- Dialog

The properties are divided into categories to group similar information together. They can be expanded and collapsed.

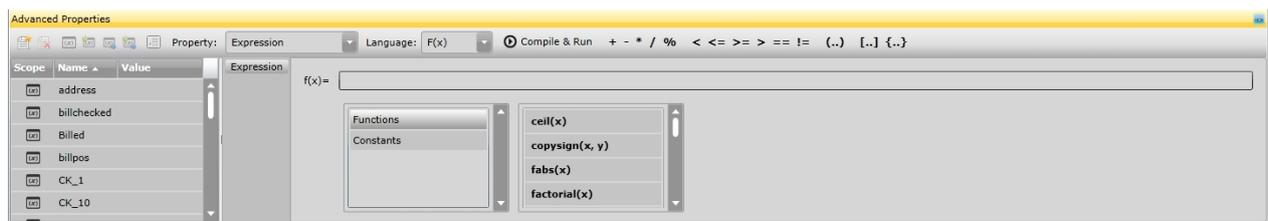
## Description

In the description you may describe the element in detail. The predefined content is a template and can be completely overwritten.

The description will usually be shown if you generate your process documentation.

## The Advanced Properties Window

In the Advanced Properties Window advanced characteristics of the active element are shown and can be modified. The example shows an expression property of a script task.



The Advanced Properties Window is divided into two areas:

- The Global Scope Area
- The Property Plugin Area

### Global Scope Area

The global scope is used to define [process instance variables](#) and their usage within an workflow operation.

### Property Plugin Area

Depending on the selected Element the Advanced Properties Window enables or disables several plug-ins with which the property settings may be edited.

These plugins are shown in the property Plugin Area.

## The Output Window

The Output Window is attached at the bottom of IYOPRO and usually in collapsed state. If you open the Output Window, you see information about the state of your process models.

Type	Source	Message	Document	Location
	Conformance Tests	Should have a label	New Process Diagram	General End Event
	Conformance Tests	No links	New Process Diagram	General Start Event

Information types are:

- Errors - These situations usually must be corrected to allow further processing
- Warnings - These situations should be corrected.
- Information - These situations are informational only and may be corrected.

In the output windows toolbar they may be selectively switched on and off. In addition you can choose, if the shown message rely on only the active document or all loaded documents.

The data grid specifies:

- Type - Error, warning or information
- Source - The module that generated the message
- Message - The message itself
- Document - The document, where the message is attached to
- Location - The location in the document that covers the message

## Zoom, Pan, Fit

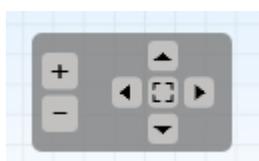
The zoom, pan and fit function can be selected in different ways.

### View control by mouse operations

The view may be controlled in the following ways

- Press and hold the Control-Key, then press and hold the left mouse button. Moving the mouse will result in a pan operation.
- Press and hold the Shift-Key, then drag a rectangle over the view to zoom into that rectangle.
- Press and hold the right mouse button. Moving the mouse will result in a pan operation.
- Press and hold the Shift-Key, then double-click somewhere in the work area to fit the diagram into the modeling window.
- Use the mouse wheel to zoom around the current mouse position in the modeling window.

### View control in the modeling window



The view control for zoom, pan and fit fades in, if you the mouse moves over it.

 Zoom in - Click to zoom in

 Zoom out - Click to zoom out

 Fit - Zoom to fit the diagram in the modeling window

Pan: the arrows can be used to pan the diagram in all directions

## Ribbon tab "View" and shortcuts



To access the tab page, select the tab header "View".



Zoom Area

Click and drag a rectangle over the desired area in the modeling window to zoom.

Shortcut: press the shift key and drag a rectangle over the desired area

 Zoom In Click to zoom in

The zoom in function also can be activated by scrolling the mouse wheel forward.

 Zoom Out Click to zoom out

The zoom out function also can be activated by scrolling the mouse wheel backwards..

 Fit Fit - Zoom to fit the diagram in the modeling window

Shortcut: press the shift key and double click somewhere in the modeling window



Pan Area

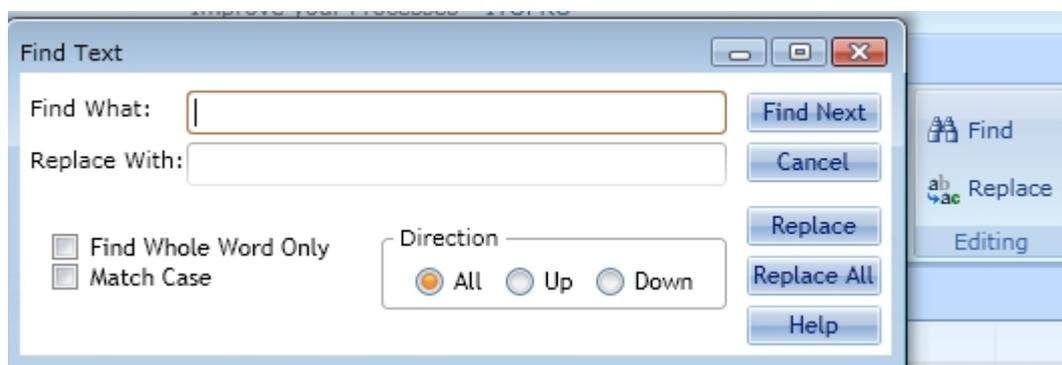
Click to activate the pan function. Drag the mouse pointer with pressed mouse button over the modeling window to pan in all directions. To exit from this operation, click somewhere in the modeling window without mouse move.

Shortcut: press the control key and drag the mouse pointer over the modeling window.



Click and release. A magnifying lens appears, which can be dragged around. To exit this operation, click onto the magnifying lens.

## Find and replace



With this dialog you can find and replace portions of text in a diagram element.

### Find

The find option compares those characters typed into the field "Find What" with the characters in the name of the diagram elements. Select "Find Next" and the first element, which contains the matching characters will be highlighted as well as the characters themselves. Click again to proceed.

### Replace

The first click on the "Replace" button highlights the matching section. Clicking again replaces the matching section.

"Replace All" replaces all matching sections in one step.

### General

The following search options are available:

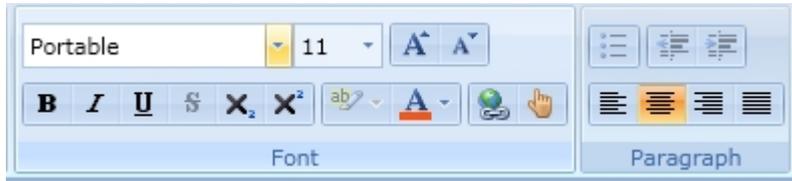
- Find Whole Word Only - The search phrase must match a whole word.
- Match Case - The search will be case sensitive

With the radio buttons in the direction box you can choose, which direction the find and replace operation will follow:

- All - Starting from the first highlighted element, considering all elements
- Up - From a selected element back to the beginning

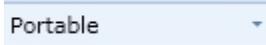
- Down - From a selected element to the end

## Editing rich text



When editing rich text the following buttons become available:

### Font

	Font - Choose from a list of fonts
	Font size - Choose from a list of font sizes
	Grow font / shrink font - Click to grow / shrink font
	Bold - Click to boldface
	Italic - Click to italicize
	Underline - Click to underline
	Strikethrough - This option is currently not available
	Subscript - This option is currently not available
	Superscript - This option is currently not available
	Background color - This option is currently not available
	Text Foreground Color - Choose from a list of font colors
	Insert link - Click to open the dialog to insert a hyperlink
	Edit / view mode - Interactive text elements like hyperlinks are functional only in view mode

### Paragraph

	Bullets - This option is currently not available
---	--



Decrease indent - This option is currently not available



Increase indent - This option is currently not available



Left-aligned - Click to align the paragraph along the left margin



Centered - Click to center the paragraph between the two margins



Right-aligned - Click to align the paragraph along the right margin



Justified alignment - Click to distribute the text equally between the margins

## Undo and Redo



The Undo and Redo Button can be found in the Quick access tool-bar.

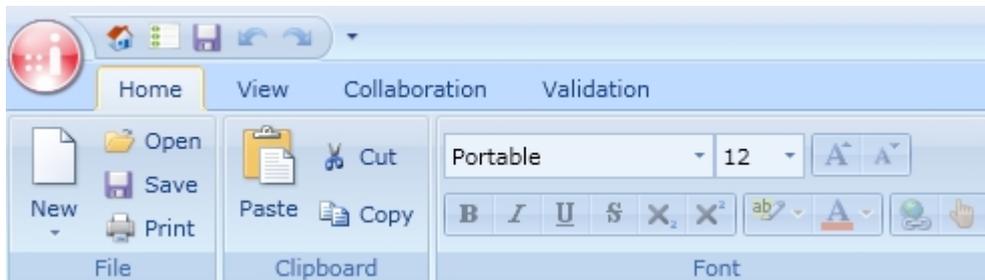


You can use the undo and redo command to revert the document to how it was before each change.

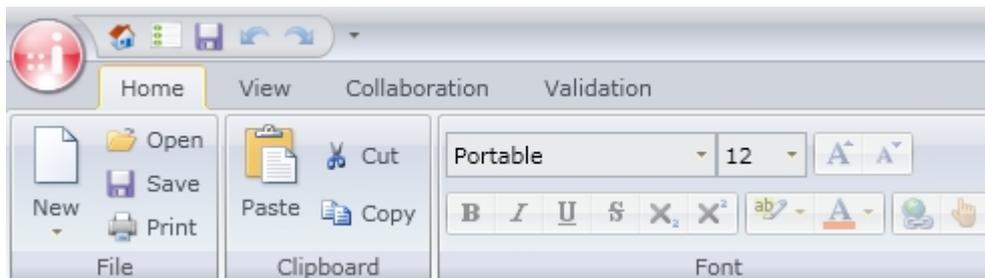
## Themes



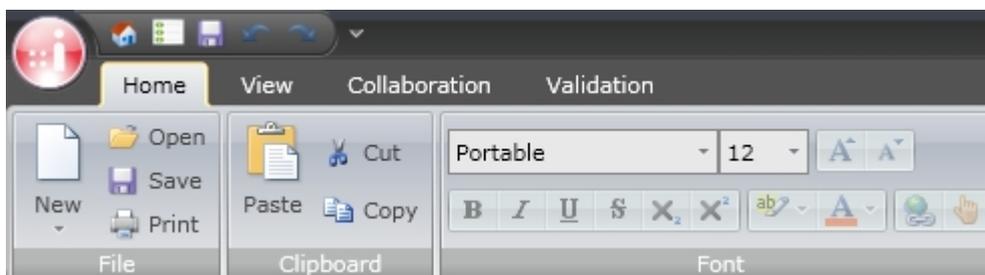
The theme option provides a different look and feel of the IYOPRO application. You may choose between blue, silver and black theme.



Blue theme

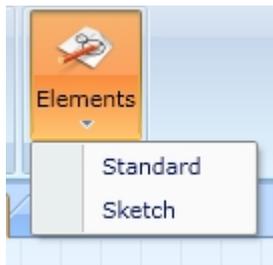


Silver theme

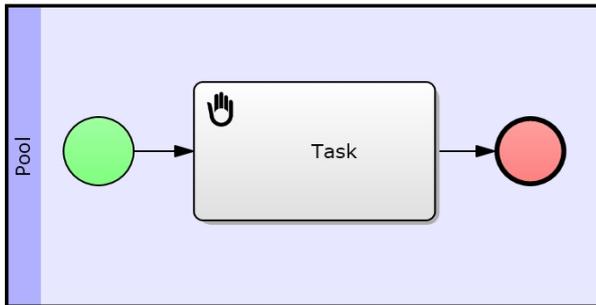


Black theme

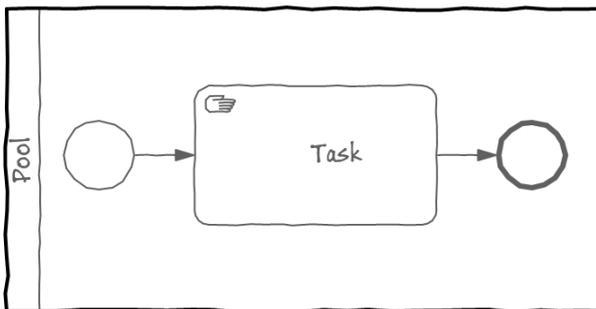
## Styles



With the style option you can control, how the shapes in diagrams are rendered.



Standard style



Sketch style

## Element arrangement

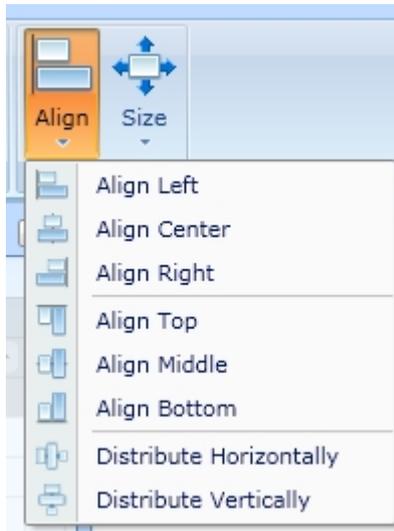


The element arrangement allows the alignment of elements in relation to a reference element. You may align and/or re-size the elements depended on the chosen operation.

The operation is as followed:

1. Select the reference element
2. Select the elements to arrange
3. Choose the operation from the arrange menu

## Align elements



Change the alignment of the items by:

- Align Left - All selected elements will be left-aligned
- Align Center - All selected elements will be centered vertically
- Align Right - All selected elements will be right-aligned.
- Align Top - All selected elements will be top-aligned
- Align Middle - All selected elements will be centered horizontally
- Align Bottom - All selected elements will be bottom-aligned
- Distribute Horizontally - All selected elements will be distributed horizontally
- Distribute Vertically - All selected elements will be distributed equally vertically

## Edit sizes



Change the size of the items by:

- Same Width - All elements selected will receive the same width
- Same Height - All elements selected will receive the same height
- Same Size - All elements selected will receive the same size

## Quick start guides

---

[Launching IYOPRO](#)

[Register a new account](#)

[Create a new diagram](#)

[Register and activate software keys](#)

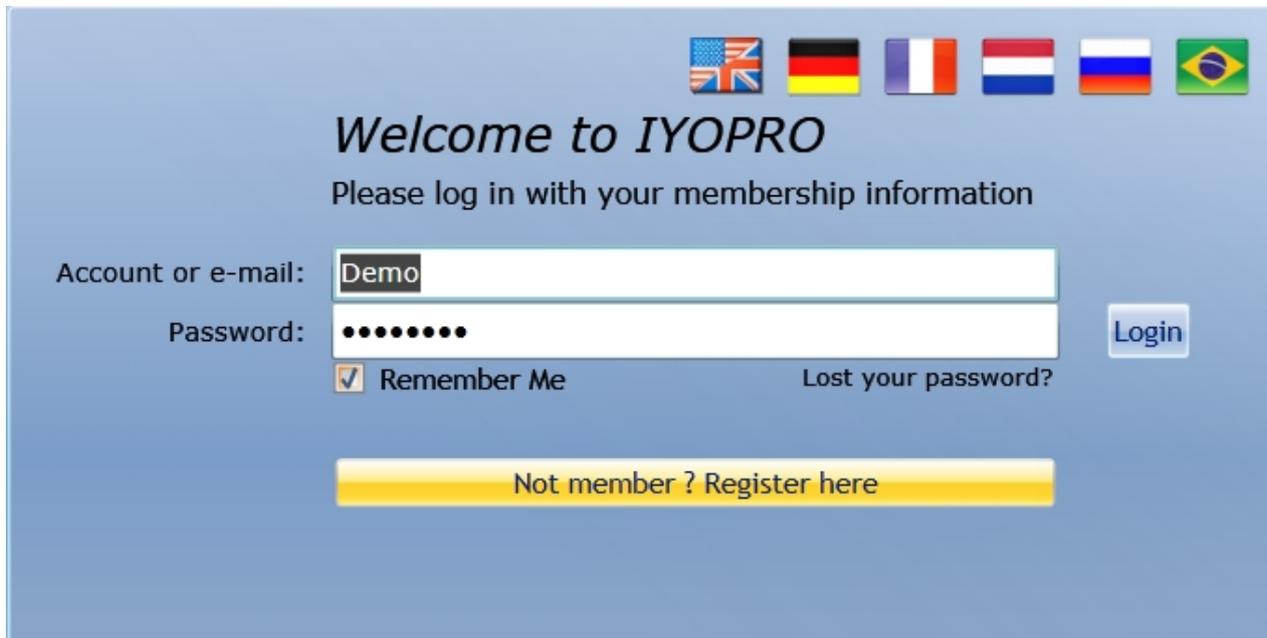
### Launching IYOPRO

To launch IYOPRO type

<https://www.iyopro.com/iyopro/>

into your Internet browser or choose the link above.

The Welcome to IYOPRO window will appear:



You may use the predefined demo account to play with IYOPRO in a public area or you may register and login with your credentials to work within a private/team environment.

### Launching IYOPRO Portal

To launch IYOPRO and navigate immediately to the portal, type

<https://www.iyopro.com/iyopro.portal>

into your Internet browser or choose the link above.

The Login window will appear:

# Login

Use your IYOPRO account name or the E-Mail address you are registered with.

Remember me

Login with your credentials. At least a Portal license is required to access the portal.

## [Register a new account](#)

First launch IYOPRO. On the welcome screen click on "Not member? Register here".

Fill in the registration form:



**Welcome to IYOPRO**  
Registration Form

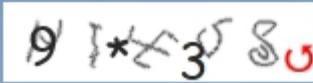
Account:

e-mail:

Password:

Repeat Password:

Remember Me

Verification Code:  =  Please enter result

[I agree to the Terms of Use](#)

[I agree to the privacy statement](#)

## Account

You can choose any user name for your account. If the user name already exists, you will receive an error message.

## e-mail

Specify your e-mail address. You will receive a registration notification to remember your user name and password.

## Password

Your password must contain at least 6 characters.

## Repeat Password

Repeat your password.

## Remember me

Click to enable the auto login feature of IYOPRO.

## Terms of use

Click to confirm that you have read the [terms of use](#).

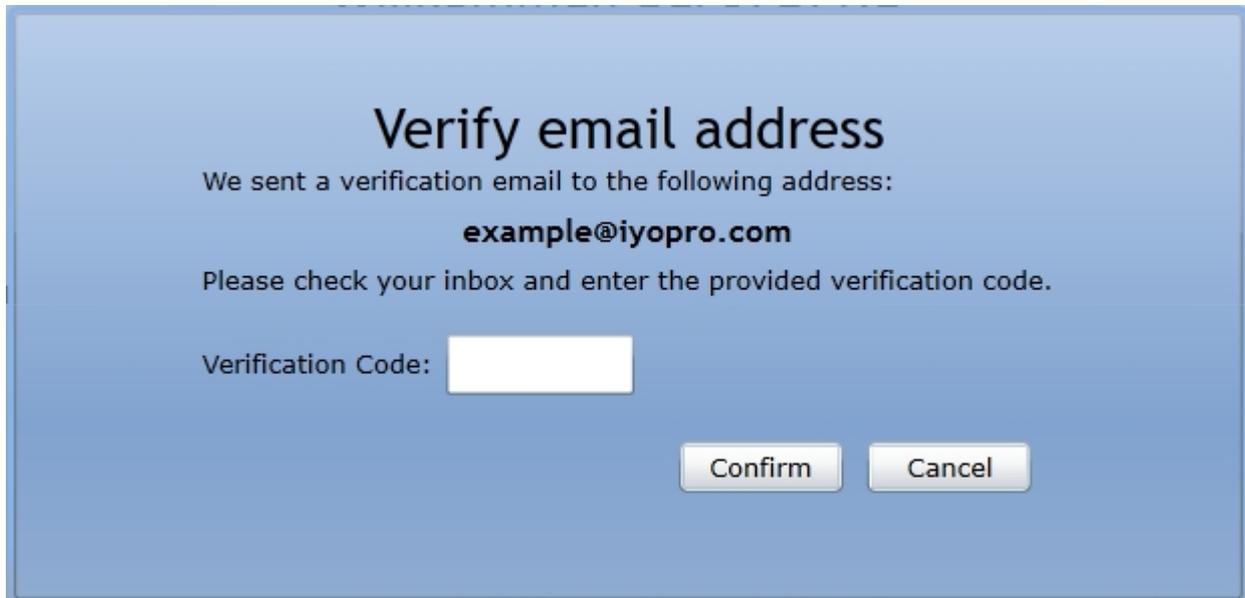
## Privacy statement

Click to confirm that you have read the [privacy statement](#).

## Verification Code and Registration

Verify your registration, enter the verification code and click "Register".

### Verify email address



**Verify email address**

We sent a verification email to the following address:  
**example@iyopro.com**

Please check your inbox and enter the provided verification code.

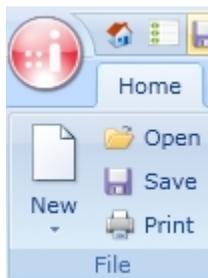
Verification Code:

IYOPRO sends an email with an generated verification code to the specified email address in the registration form. Enter the code into the verification dialog of the registration form and click "Confirm" to finalize the registration.

[IYOPRO editions](#)

## Create a new diagram

Click the "New" button to create a new diagram.



In the Toolbox Area the associated Toolbox will appear, which contains all stencils required to draw the diagram. You may now drag symbols from the toolbox onto the work area.

To learn more about the different diagram types refer to:

[Diagram types](#)

To learn more about the different options to create a new diagram, refer to:

[Single file diagramming](#)

## Register and activate software keys

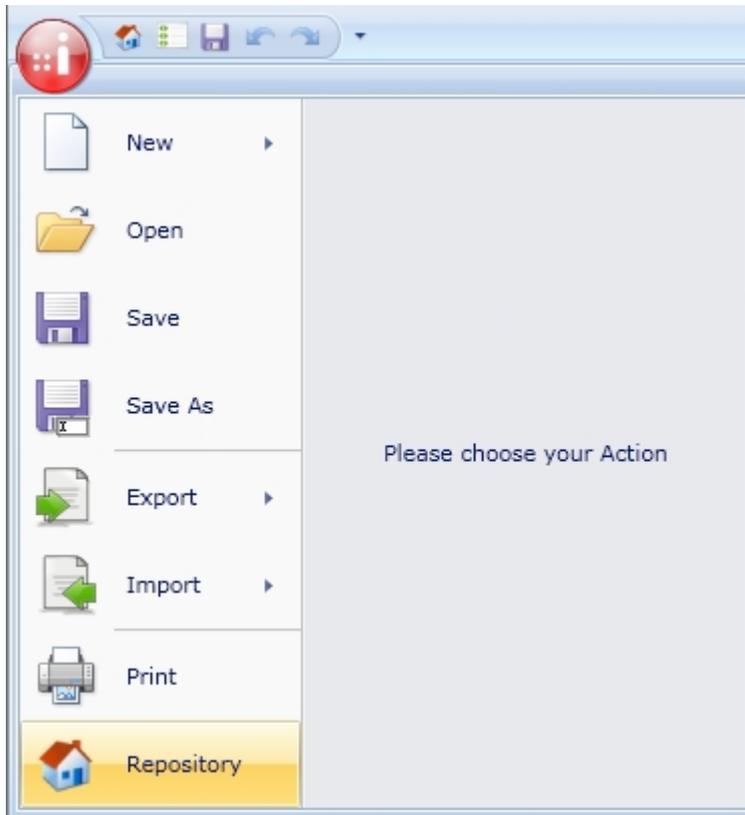
You have purchased one of the [IYOPRO editions](#). Your order has been confirmed and you received an email, which includes your software key.

To register and activate your software, please follow the instruction below:

1. Open the [repository](#) and navigate to the teams node
2. [Create a new Team](#) or select an existing team out of the repository browser
3. [Register](#) your Software Key
4. [Assign](#) software keys to users

# The Repository

To access the repository from the [Application menu](#), click Repository:



## Content:

- [The Desktop](#)
- [Working with files and folders](#)
- [Working with teams](#)
- [Recent items](#)

## The Desktop



The Desktop contains an overview of your current session as well as options to customize your user profile and your preferences.

## Content

[Edit preferences](#)  
[Edit the user profile](#)  
[Dashboard](#)

## Edit Preferences

 Save  Cancel

^ EMail	
HTML Format	<input checked="" type="checkbox"/>
^ General	
Initially selected Portal node	Desktop
Open portal maximized	<input checked="" type="checkbox"/>
Portal button opens internal portal	<input checked="" type="checkbox"/>
Single repository mode	<input type="checkbox"/>
Use Download page	<input checked="" type="checkbox"/>
Warn on exit*	<input checked="" type="checkbox"/>
^ Maximize document	
Minimize ribbon when maximizing document*	<input checked="" type="checkbox"/>
Unpin output and advanced properties when maximizing document*	<input checked="" type="checkbox"/>
Unpin process explorer when maximizing document*	<input checked="" type="checkbox"/>
Unpin properties when maximizing document*	<input checked="" type="checkbox"/>
Unpin toolboxes when maximizing document*	<input checked="" type="checkbox"/>
^ Modeller	
Conformance Analysis	<input checked="" type="checkbox"/>
X-Style for Exclusive Gateways	Default

Settings marked with an asterisk take effect only after you restart IYOPRO. The following preferences can be modified:

### Email

- HTML Format - Decide if the emails, which you receive from IYOPRO are HTML formatted or plain text.

### General

- Initially selected Portal node - Choose the page which will be presented initially when the Portal is opened.
- Open portal maximized - Decide if the Portal is opened maximized or as new tab.
- Portal button opens internal portal - The portal is opened internally (IYOPRO App) instead of externally (IYOPRO Web)
- Single repository mode - This forces IYOPRO to reopen an existing repository window instead of creating a new one
- Use Download page - Depending on your browser, the usability when downloading files from IYOPRO may be better using a download page or not.
- Warn on exit - Warning message in case you exit IYOPRO without saving.

### Maximize document

- Minimize ribbon when maximizing document - The ribbon menu is minimized.
- Unpin output and advanced properties when maximizing document - The output/advanced properties window is minimized.

- Unpin process explorer when maximizing document - The process explorer window is minimized.
- Unpin properties when maximizing document - The properties window is minimized.
- Unpin toolboxes when maximizing document - Displayed toolboxes are minimized.

## Modeler

- Conformance Analysis - The Conformance Analysis will be performed while modeling.
- X-Style for Exclusive Gateways - New Exclusive Gateways are displayed with or without X
- Initial diagram language - Choose the language in which diagrams will be created and shown initially.
- Activate New Control - New symbols dragged from the toolbox into the work area are selected by default or not.
- Display Grid Lines - Activate raster overlay.
- Display Lane Path - Show in which Swimlane(s) the selected element is located.
- Display Preview Image - Activate navigation preview.
- Display View Controls - Activate zoom pan control.
- Flip Pan Direction - Activate, if you wish to reverse the pan direction of the view control.
- New Control Font - Define the default font for new elements
- New Control Font Size - Define the default font size for new elements.
- Toolbar Visibility - Control the behavior, when the toolbar will be shown in the modeler.

## Simulation

- Simulation Analysis - The process model will be checked if the necessary "must have properties" required for simulation analysis have been specified

## Workflow

- Default workflow expression language - Choose the default expression language.

Choose "Save", if the adjustments made should become the new default.

## Edit the User Profile

**Edit User Profile**

Save Cancel Delete Profile

**Account Information**

Account: Demo      Displayed Name: Demo User

Description: Sample Demo User

Password:      Retype Password:

**Personal Information**

First Name:      Last Name:

**Contact Information**

Email: Demo@iyopro.com      Phone:

In your user profile you can edit the following information:

## Account information

- Account - This field is read only and specifies your login name.
- Displayed Name - The name that is associated with your activities.
- Description - The description is usually shown in selection lists.
- Password - You may specify your password.
- Retype Password - Repeat your password.

## Personal Information

First Name - Specify your first name

Last Name - Specify your last name.

## Contact information

Email - Specify your email address. It is required to notify you in case of events via email.

Phone - Specify your mobile phone. It's required to notify you in case of events via SMS.

## Delete Profile

If you wish to delete your profile click on the delete button and confirm the request.



Important notice:

If you delete your profile you will no longer be able to use IYOPRO.

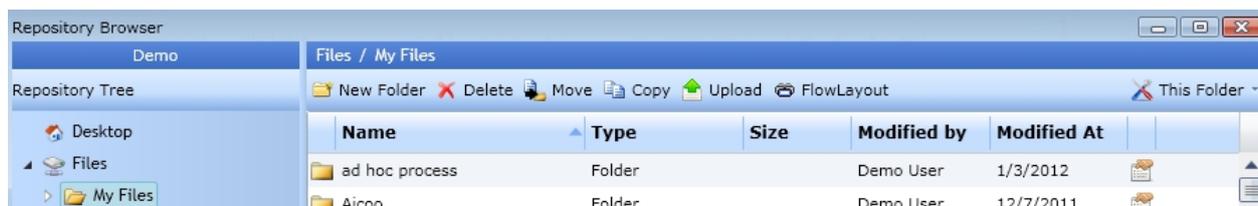
## Dashboard

<b>Pending tasks</b> There are no pending tasks. <a href="#">More...</a>	<b>Running Process instances</b> There are no running Process instances. <a href="#">More...</a>
<b>Messages</b> There are no messages. <a href="#">More...</a>	

This screen informs you about current workflow activities.

- Pending tasks - Overview of your current tasks
- Running Process instances - Overview of your monitored running process instances
- Messages - Overview of messages that have been delivered to you
- Invitations - Pending invitations to other teams

## Working with files and folders



The "Files" node is the root of your file system in which you can navigate, as you are familiar with from your desktop computer.

Depending on the selected objects several commands are available in the tool bar:



[Create new folder](#)



[Delete one or more files](#)



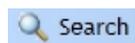
[Move one or more files](#)



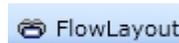
[Copy one or more files](#)



[Upload one or more files](#)



[Search files](#)



[Flow layout](#) - control the display of the folder content



Folder operations i.e. [Edit file security](#) and [Export folders](#)

## My Files vs. Team Files

Files					
Name	Type	Size	Modified by	Modified At	
My Files	Folder		Demo User	5/7/2010	
Sample Team Files	Link		Demo User	5/7/2010	

"My Files" is your private folder, which per default only can be accessed by you.

Each team has its private team folder, where all team members have access to.

To learn more about "Teams" refer to:

[Working with teams](#)

## Creating folders

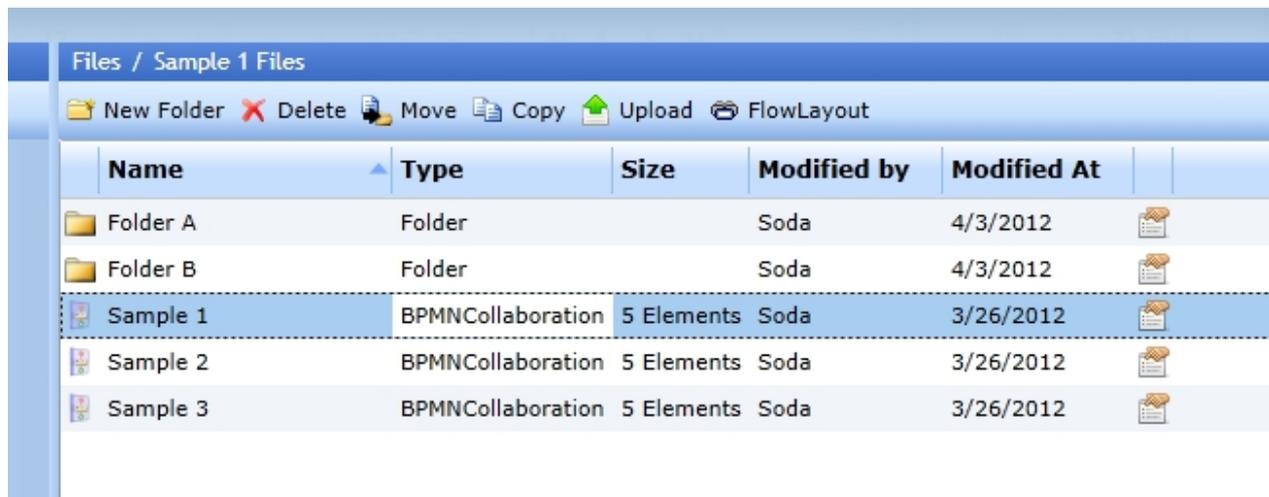


To create a new folder click the "New Folder" button, enter the folder name and save.

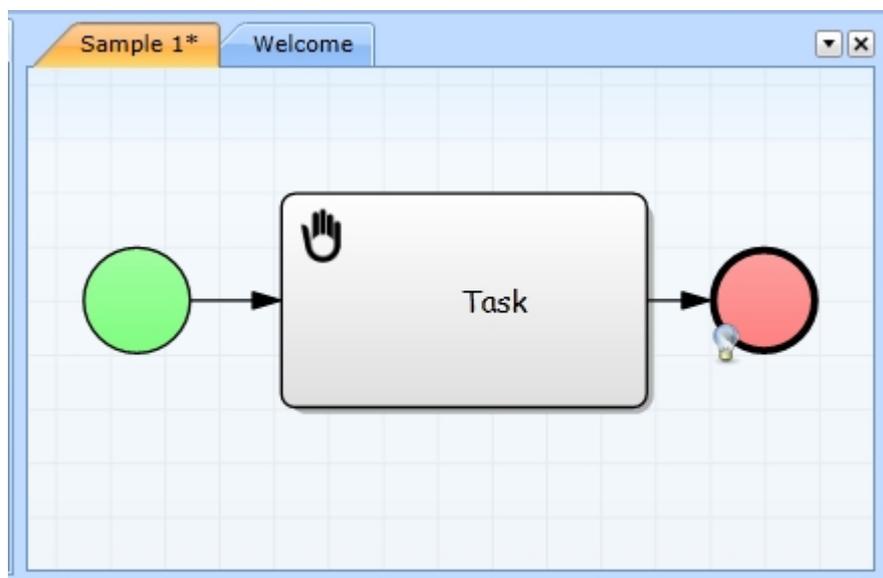
If the folder name already exists, you will receive an error message.

## Open Files for edit

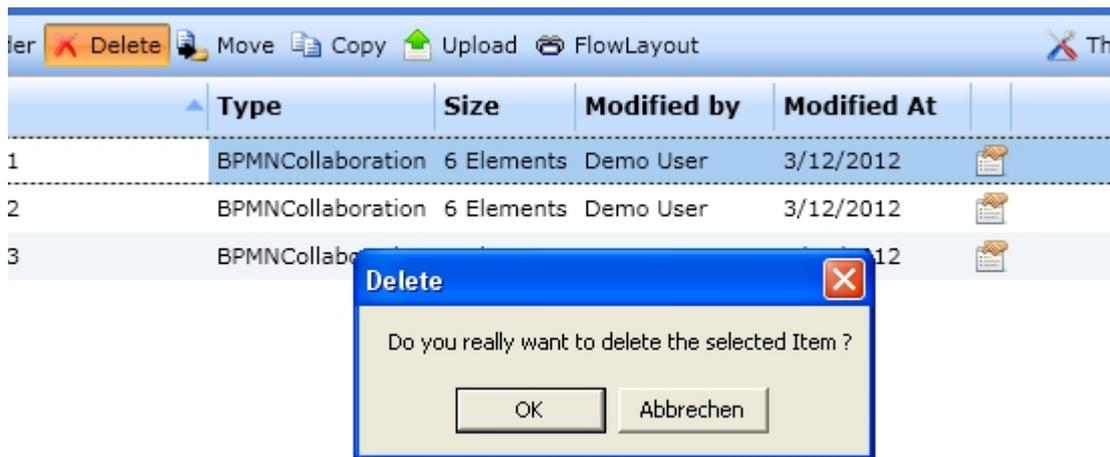
To open a file for edit simply double click on the item....



and the file will be opened in a new window.



## Delete files

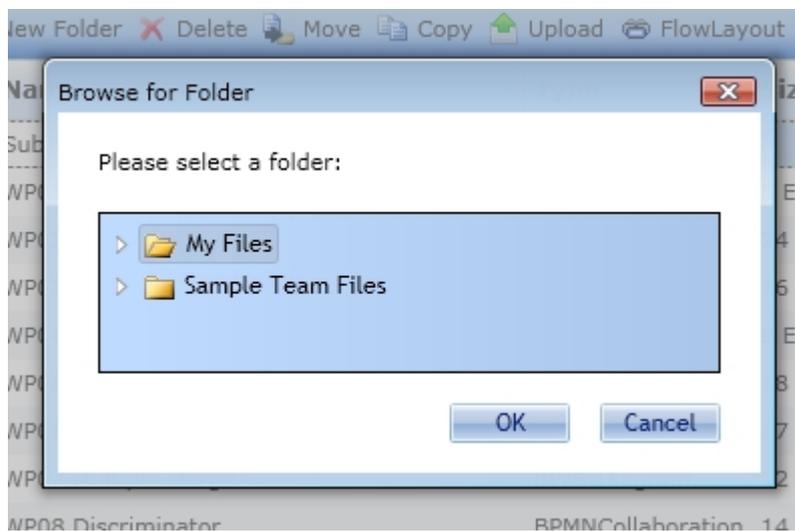


To delete files, select one or more files and click the "Delete" button. Warning: You can not undo a delete operation.



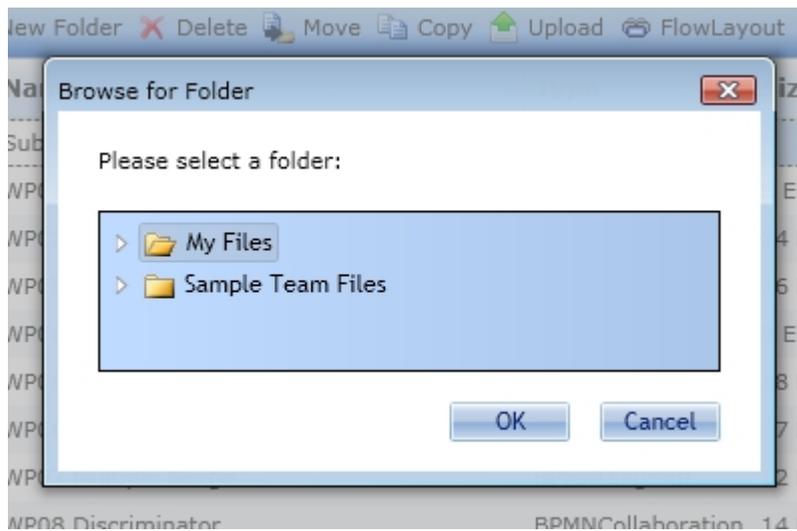
Warning: You can not undo a delete operation and there is no trash available to restore the data

## Move files



To move files from one folder to another, select these files and click the "Move" button. Choose the folder, where to move the items to and click "OK".

## Copy files



To copy files from one folder to another, select these files and click the "Copy" button. Choose the folder, where to copy the items to and click "OK".

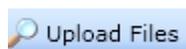
## Upload files



This features allows you to upload files from your local computer into the Repository.

To upload files, click the "Upload" button in your tool bar. Uploaded files will be stored in the current directory.

The following options are available:



Click to browse for files on your local computer. You may select one or more files. Click "Open" to start uploading.



Clears the list of downloaded files

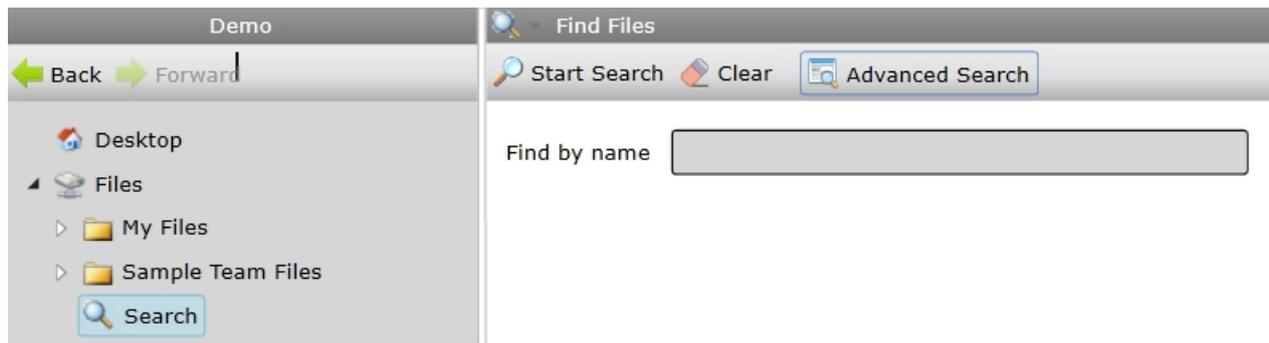


To quit the dialog

While uploading the files, each file will be shown in a file list with the current upload status. After the upload has been finished the file size or a possible error message will be displayed.

If a file already exists in the repository, IYOPRO will create a new version of that file.

## Search files



To search files click the "Files" node. The search functionality is located at the end of the expanded "Files" node. The standard search is displayed initially. This search will list every file and folder that contain the entered string.

Pressing the "Clear" button resets every parameter of the standard and advanced search.

Furthermore advanced search options can be used. They provide detailed search options as they are displayed below.

Name	Contains	<input type="text"/>
Type	Contains	<input type="text"/>
Created	After	<input type="text" value="&lt;M/d/yyyy&gt;"/> 15
Created by	Contains	<input type="text"/>
Modified	After	<input type="text" value="&lt;M/d/yyyy&gt;"/> 15
Modified by	Contains	<input type="text"/>
Element Term	Contains	<input type="text"/>
Search Folder	<input type="text"/> ...	<input checked="" type="checkbox"/> Include subfolders
Tag	<input type="text"/>	
Contains Library Element	<input type="text"/> ...	
Search for	<input type="radio"/> Files only <input type="radio"/> Files and Folders	

Tag

Name

Element Term

Type

Created

Created by

Modified

Modified by

Search Folder    Include subfolders

Contains Library Element

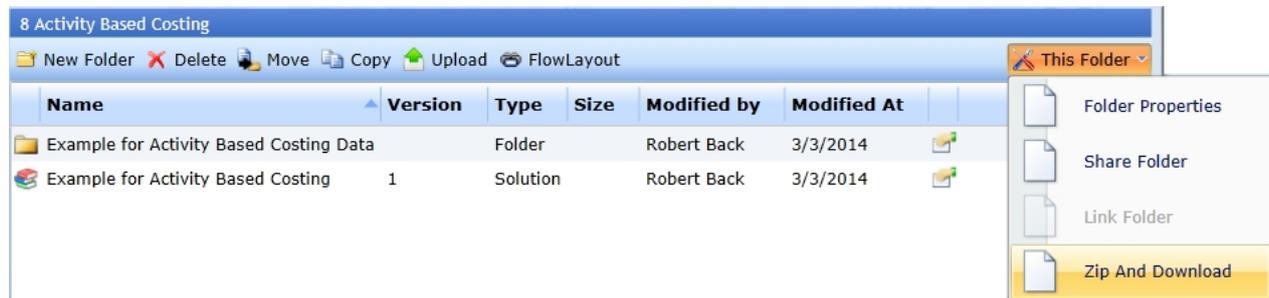
Search for  Files only  Files and Folders

Responsible

Explanation of advanced search options:

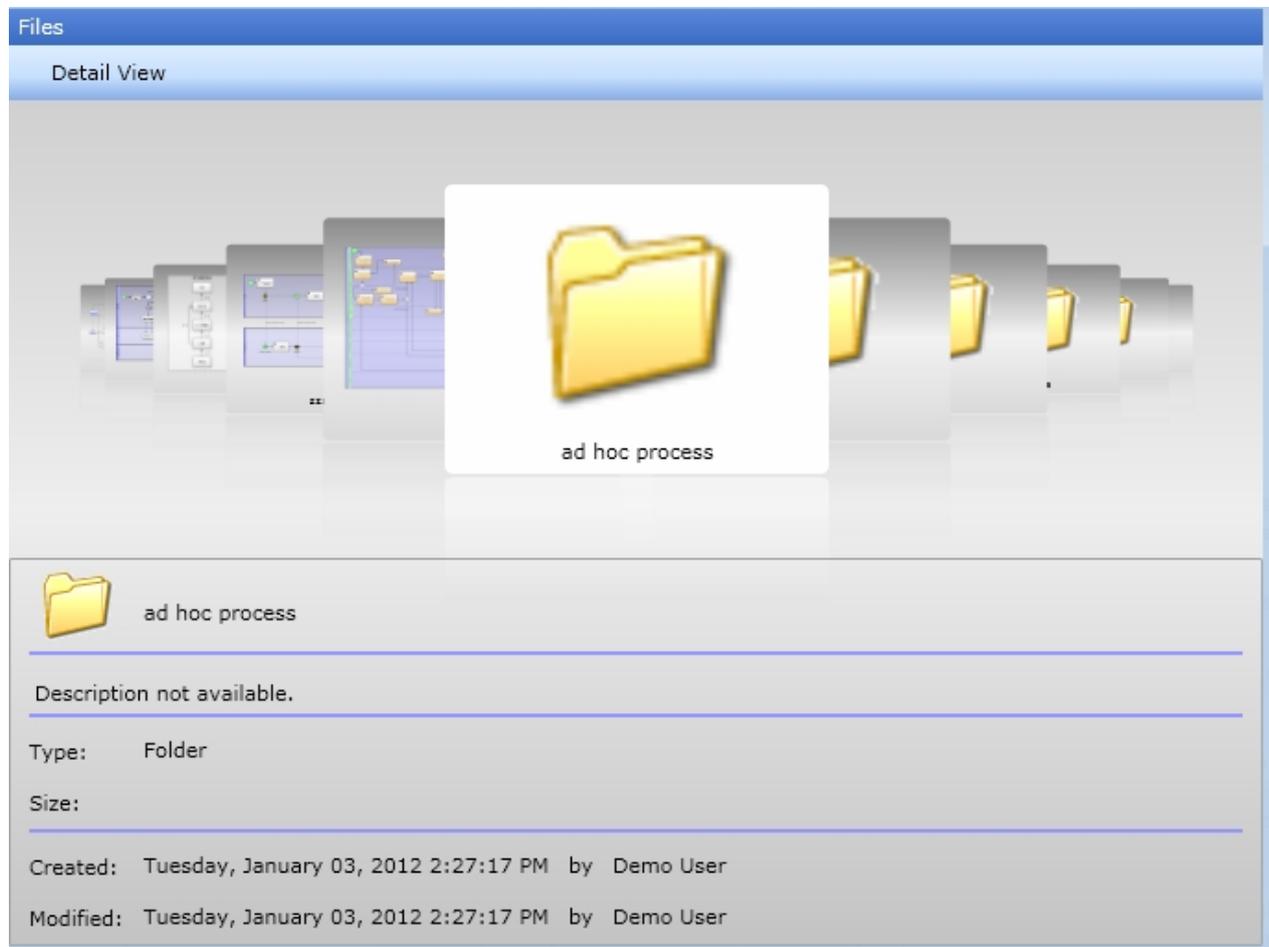
- Tag  
Search for elements containing the given Tag
- Name  
Search for element name
- Element Term  
Search for elements containing the demanded term in any element content (descriptions e.g.)
- Type  
Search for element type (Folder, Solution and ProcessMap e.g.)
- Created  
Search for date of element creation
- Created By  
Search for account ID or name of element creator
- Modified  
Search for modification date
- Modified By  
Search for account ID or name of element modifier
- Search Folder  
Search in a specific folder
- Contains Library Element  
Search for elements containing the given Library Element
- Search for  
Choose if you want to find files and folders or files only
- Responsible  
Search for elements where the given name has the chosen responsibility(s)

## Export folders



IYOPRO allows you to export folders including their content. Open the folder. Click This Folder -> Zip And Download and follow the instructions. To import these files back into IYOPRO use the [upload function for files](#).

## Flow layout



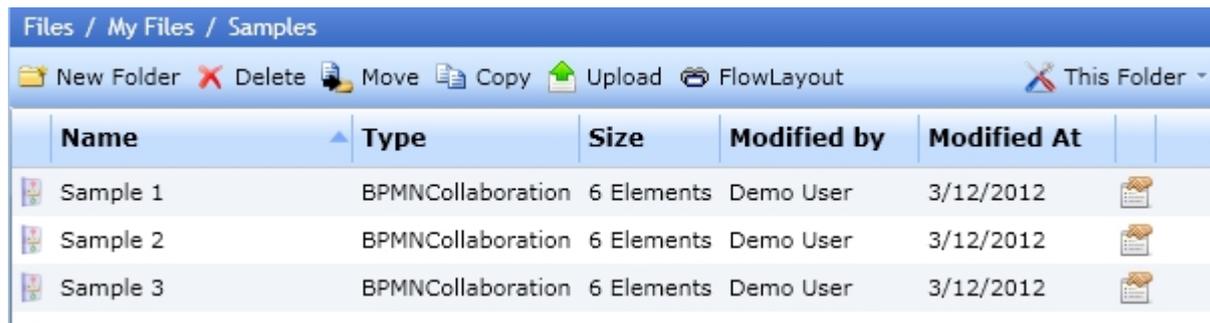
The "Flow Layout" provides a different view on the content of your folders.

Select "Flow Layout" from the folders tool-bar. In the upper part of the window the folder contents will be displayed as thumbnails. To browse the folder, move the mouse while pressing the left button from the left to the right or the right to the left.

Select an item to open the item details, double click to open.

Select "Detail View" from the tool-bar to quit the "Flow Layout".

## File properties

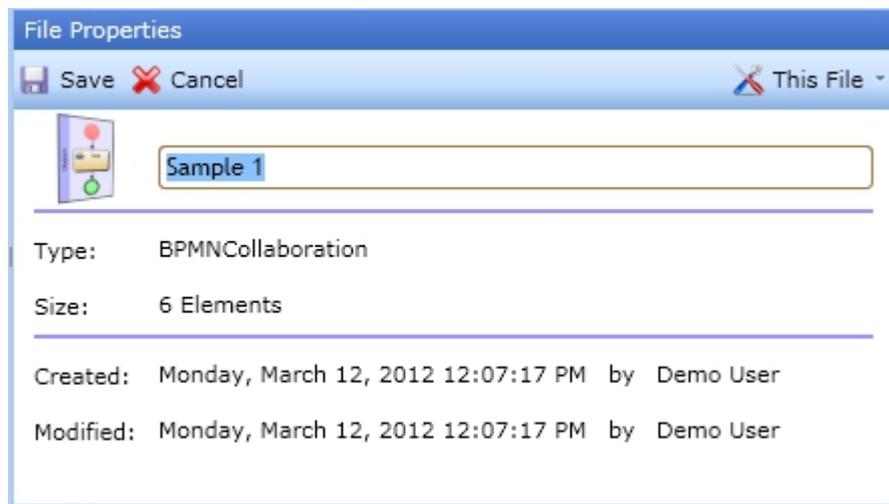


Files / My Files / Samples

New Folder Delete Move Copy Upload FlowLayout This Folder

Name	Type	Size	Modified by	Modified At
Sample 1	BPMNCollaboration	6 Elements	Demo User	3/12/2012
Sample 2	BPMNCollaboration	6 Elements	Demo User	3/12/2012
Sample 3	BPMNCollaboration	6 Elements	Demo User	3/12/2012

From within the folders view select  to view the file properties.



File Properties

Save Cancel This File



---

Type: BPMNCollaboration

Size: 6 Elements

---

Created: Monday, March 12, 2012 12:07:17 PM by Demo User

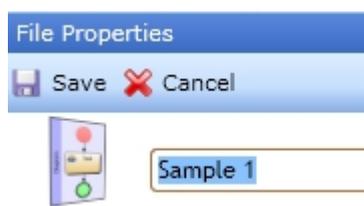
Modified: Monday, March 12, 2012 12:07:17 PM by Demo User

## Content:



[Rename files](#)  
[Edit file security](#)

## Rename files



File Properties

Save Cancel



Rename the file and click the "Save" button.

## Edit file security

# Access Control Management

A few general words about authorization management (user access control) in IYOPRO before going into the details.

IYOPRO uses Access Control List (ACL) to facilitate user access control. ACL is a technique to determine which user may access which services and files.

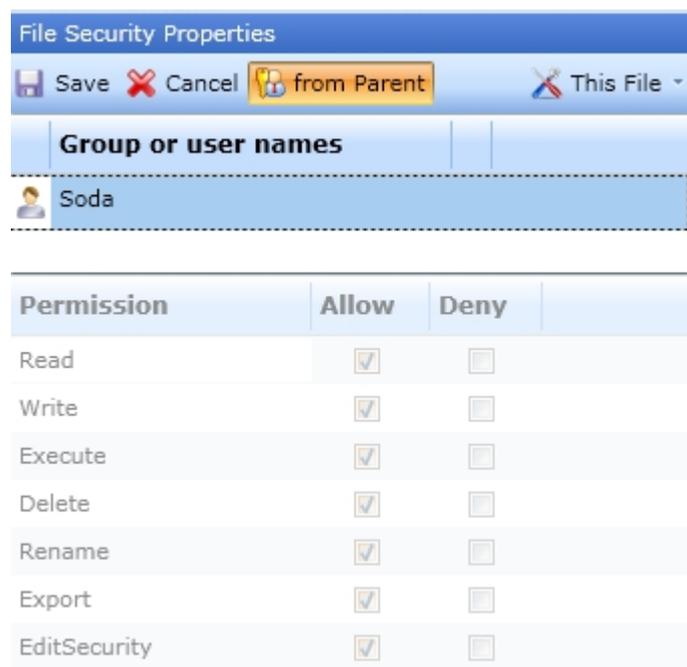
The repository in IYOPRO is built as a hierarchically organized tree structure. This means that each element has exactly one direct predecessor (parent element). The exception is the root, which is the top level element. Elements below a parent element are called child elements. Below a "child" other child elements can be grouped.

In general the parent element passes its access rights to the child element. To facilitate information security the rights can be restricted individually for each object.

For further details refer to section at the the bottom of this page: "Inherited versus effective user rights".

## Edit file security

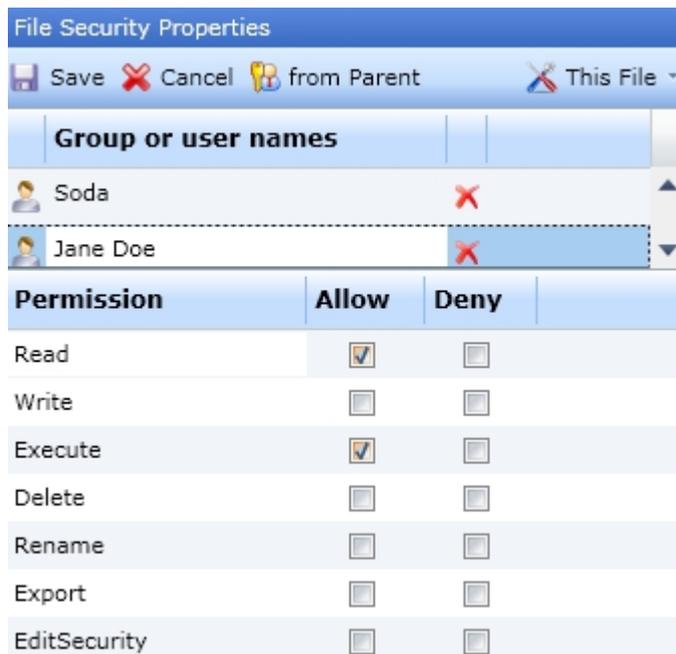
Click the "This File" button in the "File Properties" dialog and select "Security".



The access control list will appear.

The "from Parent" button out of the tool-bar indicates, if the ACL is inherited from the parent folder or not. In case, the ACL is inherited you will not be able to change security settings for this file. You first have to cut the ACL relationship to the parent. To do so toggle the "from Parent" button.

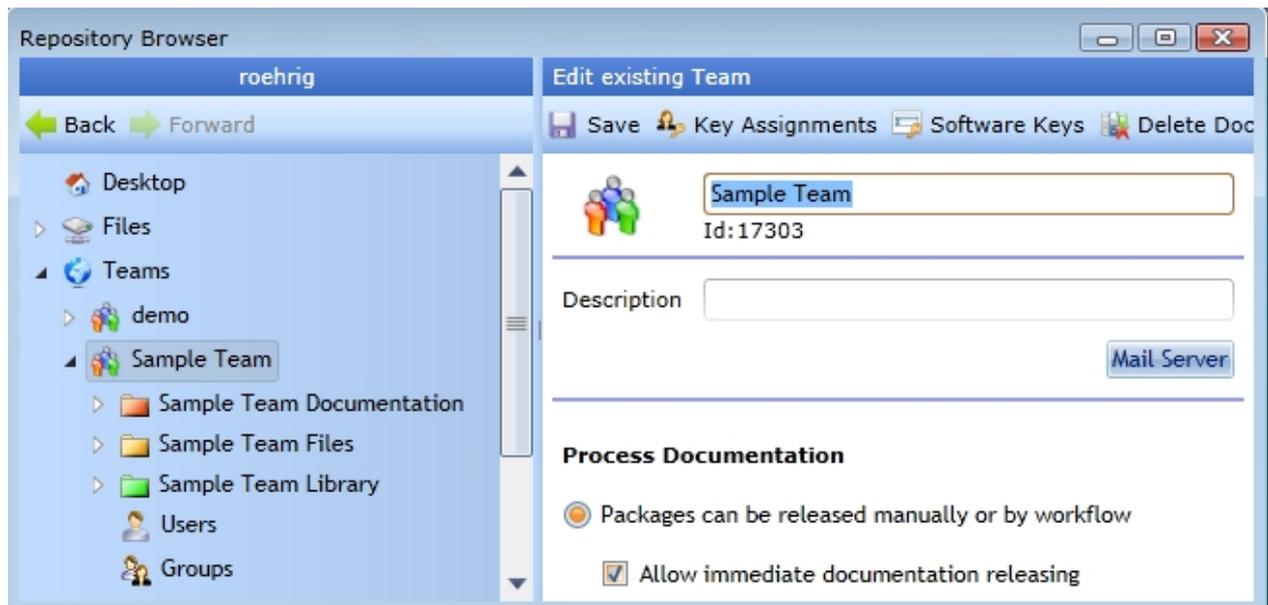
To assign rights to a user or a group, click into a new line in the upper section. Open the list and choose the user/the group.



In the lower section (Permission area) you may assign rights as desired. Click the "Save" button to save your changes. If a permission is denied it will outdo any allowance. The following permissions can be granted or denied:

- Read  
Required right to open files/diagrams
- Write  
Required right to save changes in files/diagrams
- Execute  
Required right to launch workflow instances
- Delete  
Required right to delete files/diagrams
- Rename  
Required right to rename files/diagrams
- Export  
Required right to use the export functionality for files/diagrams
- Edit Security  
Required right to change the ACL settings

## Working with teams



## Teams node

The "Teams" node is the root of all teams you are a member of.

Choose the Teams node and you may:



[Create a new team](#)



Join an existing team with a [Team Access Code](#)



[Delete a team](#)

## Team

A Team is an organizational element to group people together on a defined security level to share common resources.

Each team has a name and consists out of files, users and groups.

Choose a specific team and you may:

[Edit team properties](#)



[Register software key](#)



[Assign software keys to users](#)

## Files

The folder name, in which the team may share files is build out of the team name with the appendix "Files". This folder will be available in the Files node of each team member.

Refer to [Working with files and folders](#) to learn more about the options to edit files and folders.

Per default the teams Everyone group is permitted to access the files node of the team. These permissions may be adjusted by the administrators if required.

## Documentation

The folder name, in which the team may share documentation is build out of the team name with the appendix "Documentation". This folder will be available in the Documentation node of the IYOPRO Portal  
Per default the teams Everyone group is permitted to read the Documentation node of the team. These permissions may be adjusted by the administrators if required.

## Library

The folder name, in which the team may share library items is build out of the team name with the appendix "Library".

## Users

Team members are referred to as "Users". A "User" may be member of multiple teams.

Choose the Users node and you may:

 Add User

[Create new users](#)

 Delete

[Delete users](#)

 Set default substitutes

[Set default substitutes for users](#)

## Groups

Within a team "Groups" may be created to structure the team more deeply. There are two default groups in every team:

1. Administrators
2. Everyone

The "Administrators" group at least consists of the creator of the team. He is permitted to expand and edit this group.

The "Everyone" group includes all team members.

Choose the Groups node and you may:

 New Group

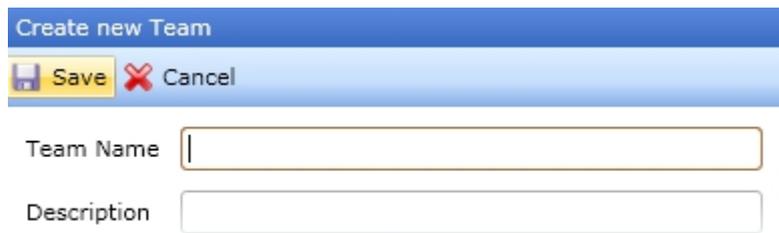
[Create a usergroup](#)

 Delete

[Delete a usergroup](#)

[Edit the content of a usergroup](#)

## Create a new team



Create new Team

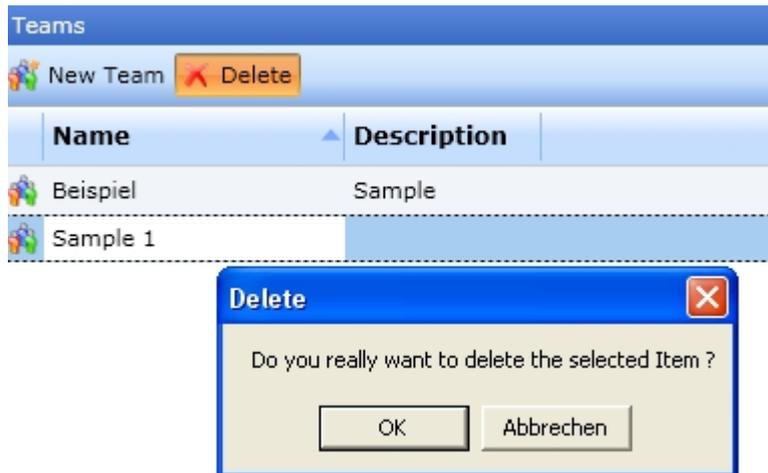
Save Cancel

Team Name

Description

From within the Teams node click the "New Team" button, enter the team name and the team description. Hit the "Save" button to create the team.

## Delete a team



Teams

New Team Delete

Name	Description
Beispiel	Sample
Sample 1	

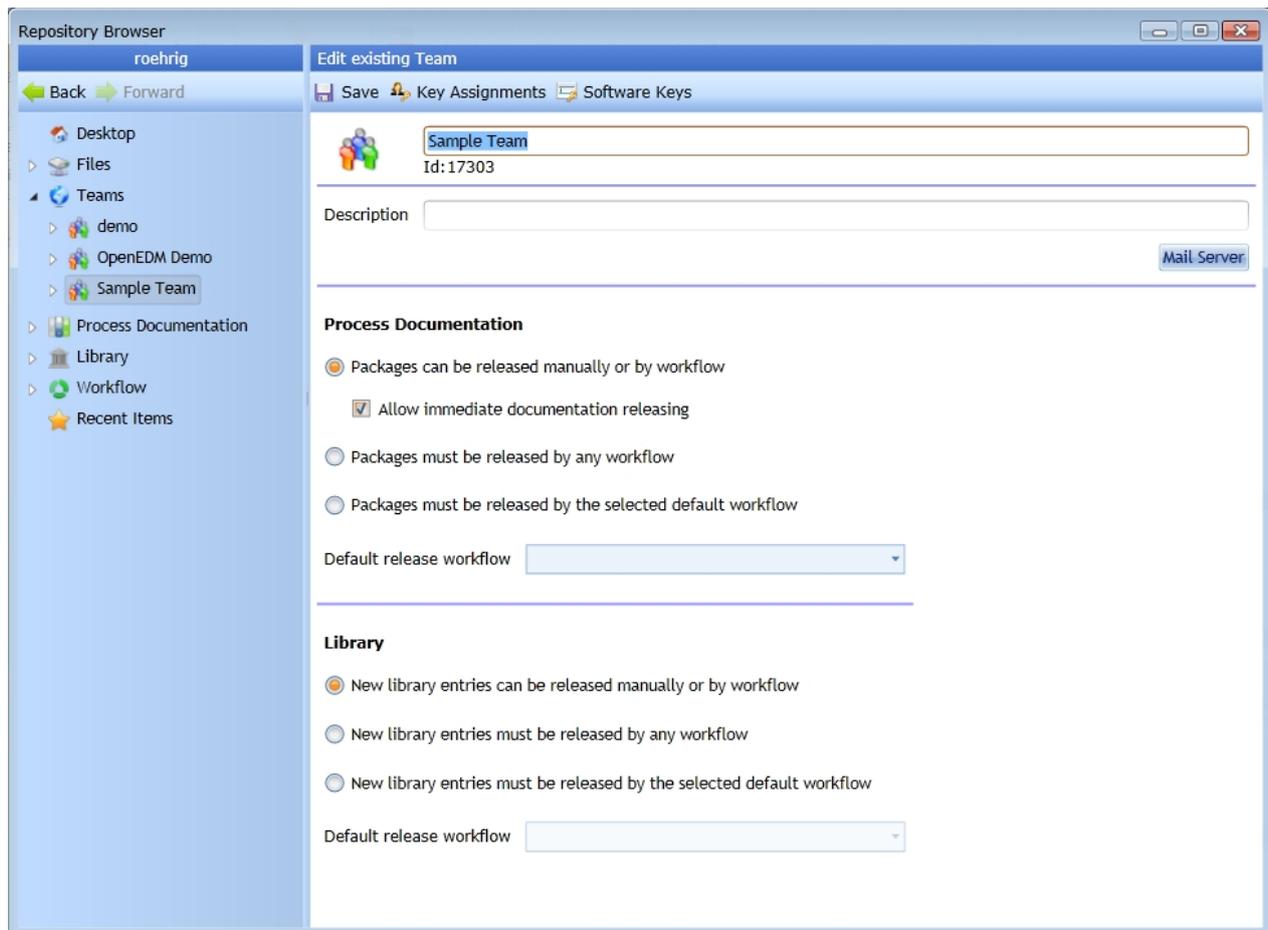
Delete

Do you really want to delete the selected Item ?

OK Abbrechen

Select one or more teams and click the "Delete" button. If you delete a team, the associated users will not be deleted. Warning: You can not undo a delete operation.

## Edit team properties



Select the desired team, modify the properties and click the "Save" button.

You can control the following settings:

- [Mail server](#)
- [Process Documentation](#)
- [Library Elements](#)

## Mail Server

The feature "Mail Server" allows to specify a SMTP server. (Simple Mail Transfer Protocol). Doing so enables your workflows to send emails to any email address. IYOPRO will transfer the emails to the server specified for delivery.

You may specify these credentials for users as well as for teams, which can both be master of a workflow.

Contact Editor

In order to enable your workflows to send emails to any email address you must specify your own smtp credentials. You may specify these Credentials for users as well as for teams which can both be master of a workflow.

If you omit these settings, you can only send emails to registered IYOPRO users.

SMTP Server

SMTP Username

SMTP Password

SMTP Reply Name

SMTP Reply To

OK Cancel

Enter your SMTP credentials:

- SMTP Server - the name or ip adress of you SMPT server
- SMTP Username - the username to log into your SMTP server
- SMTP Password - the password to log into your SMTP server
- SMTP Reply Name - the plain name, which will be used as sender of the email.
- SMTP Reply To - the email adress which will be used as sender of the email.

Click "OK" to confirm. Without these settings, emails only will be sent to registered IYOPRO users.

### Process Documentation

A process documentation, which has to be published into the IYOPRO portal, must pass the release process.

With this setting the team administrator controls, who might initiate a release and how it might be done:

This can be:

- Packages can be created by everyone or by a specific group in the team
- Packages can be released manually or by workflow  
Use this, if the deployment can be done by an authorized user without the need of a release workflow  
If you check the option "Allow immediate documentation releasing", the deployment package can be released immediately by the publisher. Otherwise it must be released by an auditor
- Packages must be released by any workflow  
Use this, if you have several publishing workflows defined. The publisher must choose one of the workflows to start the release process.
- Packages must be released by the selected workflow  
Use this, if you request the release process to use a specific workflow.

Read [this](#) to learn more about process documentation deployment.

### Library Elements

A library element, which has to be published into the IYOPRO portal, must pass the release process.

With this setting the team administrator controls, how the release might be done:

This can be:

- New library entries can be released manually or by workflow  
Use this, if the deployment can be done by an authorized user without the need of a release workflow  
If you check the option "Allow immediate documentation releasing", the deployment package can be released immediately by the publisher. Otherwise it must be released by an auditor
- New library entries must be released by any workflow  
Use this, if you have several publishing workflows defined. The publisher must choose one of the workflows to start the release process.
- New library entries must be released by the selected workflow  
Use this, if you request the release process to use a specific workflow.

## Register software key



Select the "Software Keys" button from within the [edit team properties](#) dialog. Hit the "Add" button and the "Software Key Registration" dialog will appear. Enter the software key, which has been provided to you via email confirming your IYOPRO purchase order. Click "OK" to register.

After you have registered a software key you will find it in the registered software key overview with the following details:

- Software Key - The key you entered
- Type - The IYOPRO edition for which the key has been exposed, refer to: [IYOPRO editions](#)
- Amount - The quantity of licenses.
- Valid Until - The expiration date of the key.
- Issued for - The name or company, the key has been issued for.

Click "Back" to assign the key to users in the next step.

You must be member of the Administrators Group to perform this operation.

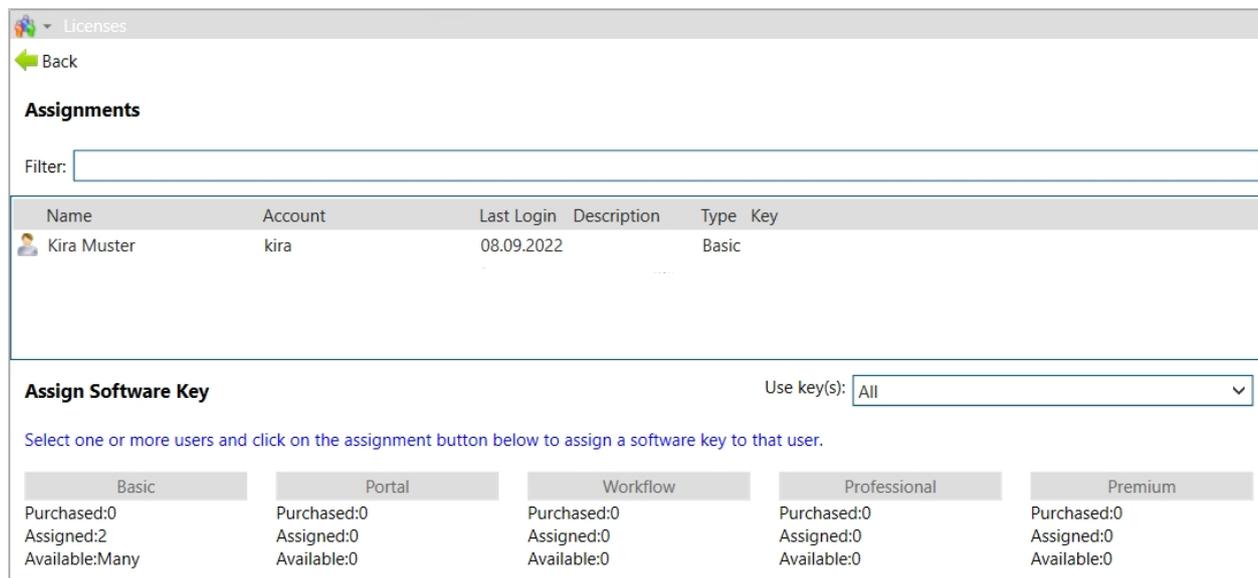
For an overview refer to:

[Register and activate software keys](#)

## Assign software keys to users

You must be a member of the Administrators Group to perform this operation.

Click the "Key Assignments" button.



The screenshot shows a web interface for managing licenses. At the top, there is a 'Licenses' header with a 'Back' button. Below this is the 'Assignments' section, which includes a 'Filter:' input field. A table lists user assignments with columns for Name, Account, Last Login, Description, Type, and Key. The table contains one entry for 'Kira Muster' with account 'kira', last login '08.09.2022', and type 'Basic'. Below the table is the 'Assign Software Key' section, which features a 'Use key(s):' dropdown menu set to 'All'. A blue instruction text reads: 'Select one or more users and click on the assignment button below to assign a software key to that user.' At the bottom, there are five buttons for different key types: 'Basic', 'Portal', 'Workflow', 'Professional', and 'Premium'. Each button displays statistics: 'Purchased:0', 'Assigned:0', and 'Available:0'.

Name	Account	Last Login	Description	Type	Key
Kira Muster	kira	08.09.2022		Basic	

**Assign Software Key** Use key(s): All

Select one or more users and click on the assignment button below to assign a software key to that user.

Basic	Portal	Workflow	Professional	Premium
Purchased:0	Purchased:0	Purchased:0	Purchased:0	Purchased:0
Assigned:2	Assigned:0	Assigned:0	Assigned:0	Assigned:0
Available:Many	Available:0	Available:0	Available:0	Available:0

The assignment list contains all users, which are a member of your team. The key-column indicates the current key assignment for each user.

To assign another key to users

1. Select one or more users to edit
2. Click on the Basic / Portal / Workflow / Professional / Premium button to assign this key to the selected users.

If you want to assign only one specific key, you can change the option from "Use key(s)" to that key. You may also change the keys assigned to users at a later point in time.

Filter option: If you want to assign licenses to a specific group of people, you have the option of copying the names from another file or dragging them directly into the filter.

For an overview refer to:

[Register and activate software keys](#)

## Add Users

IYOPRO offers three methods to add users to a team

## Add User

✖ Cancel

### → Create new user

Choose this option to create a new IYOPRO Account for another user and add the newly created account to this team. That user will receive a notification email

### → Invite existing user by email

Choose this option to invite an existing user to this team. The user will receive a notification email and may accept or discard your invitation

### → Create or modify a team access code

Choose this option to let users join the team by an access code. You need to provide the code to one or more users which are then able to join the team

- [Create new user](#)  
This option is used to create new user accounts and attach them immediately to the team. Administrators typically use this method to precreate IYOPRO accounts for their users. Creating users this way enables administrators to edit user related information (e.g. password, name, and contact information).
- [Invite existing users by email](#)  
This option will be used to invite any user to participate at the team. The targeted user will receive an email. The user may then log in or register and may then [join the team](#).
- [Create or modify a team access code](#)  
With this method a time limited team access code (i.e. a special password) is generated. Every user, who receives the team access code may [join the team](#). This option is the best to invite multiple people at once.

## Create new users

## Create new User

Save Cancel

### Account Information

Account	<input type="text"/>	Displayed Name	<input type="text"/>
Description	<input type="text"/>		
Password	<input type="text"/>	Retype Password	<input type="text"/>

### Personal Information

First Name	<input type="text"/>	Last Name	<input type="text"/>
------------	----------------------	-----------	----------------------

### Contact Information

Email	<input type="text"/>	Phone	<input type="text"/>
-------	----------------------	-------	----------------------

Member of Groups		
Name	Description	Team
 Administrators		Sample 1
 Sample Group 1		Sample 1

Click the "New User" button and enter the user information:

In your user profile you can edit the following information:

### Account information

1. Account - You may choose any user name for the new users account. If the user name already exists, you will receive an error message.
2. Displayed name - The name that is associated with the new users activities
3. Description - The description is usually shown in selection lists.
4. Password - Enter a password for the new user. The password must contain at least 6 characters.
5. Retype Password - Repeat the password.

### Personal Information

First Name - Enter the first name of the new user

Last Name - Enter the last name of the new user

### Contact information

Email - Specify the email address of the new user. It is required to notify him in case of events.

Phone - Specify the mobile phone of the new user. It's required to notify him in case of events via SMS.

### Member of Groups

Select one or more groups to assign the new user to.

You must be member of the Administrators Group to perform this operation.

## Inviting existing users



Invite User

Invite Cancel

Email Address

If you wish existing IYOPRO users to participate in your team you may invite them.

Click the "Invite" button to display the invite user dialog. Enter the users email address and hit "Invite" to send the invitation notification to that user.

If the invited user does not have an IYOPRO account the user may register and can then [join the team](#) by accepting the invitation.

## Create or modify a Team Access Code



Save Cancel

Team Access Code  Generate

Valid until

If you wish to invite multiple users to an IYOPRO team create or modify a team access code.

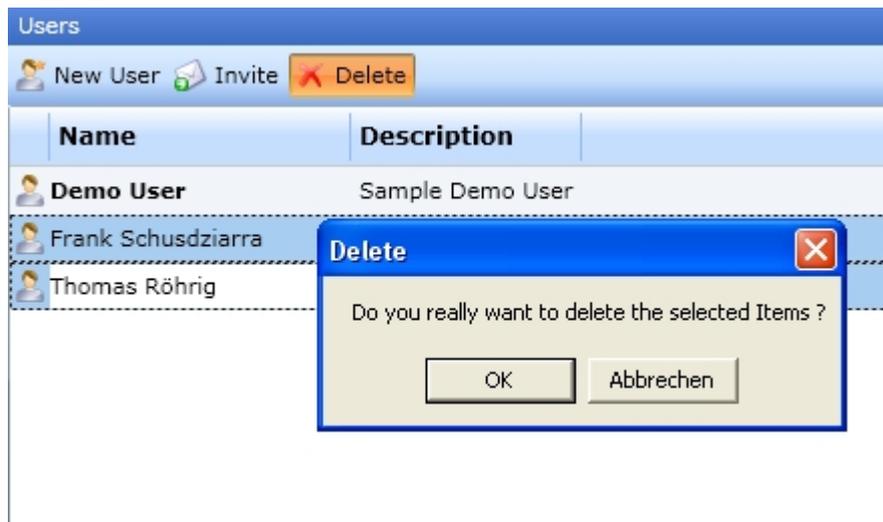
Click the "Add User->Create or modify a team access code" button to display the TAC user dialog. To generate the code click on Generate. You may then copy the code to the clipboard and distribute it to the targeted users.

You are requested to specify a "Valid until" date. The TAC will expire after that date.

Press the Save button to activate the key.

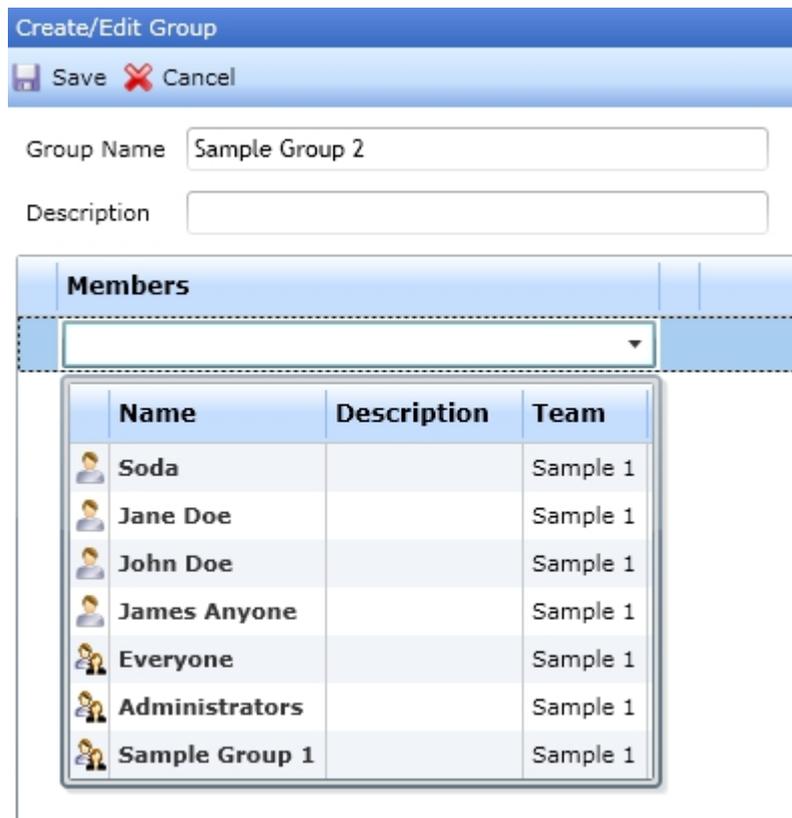
The targeted users may now [join the team](#).

## Delete users



Select one or more users and hit the "Delete" button. You can not undo a delete operation.

### Create a usergroup



Click the "New Group" button, enter the group name and the description. The members can be chosen from the list displayed. Hit the "Save" button.

Note: Groups may be member of other groups.

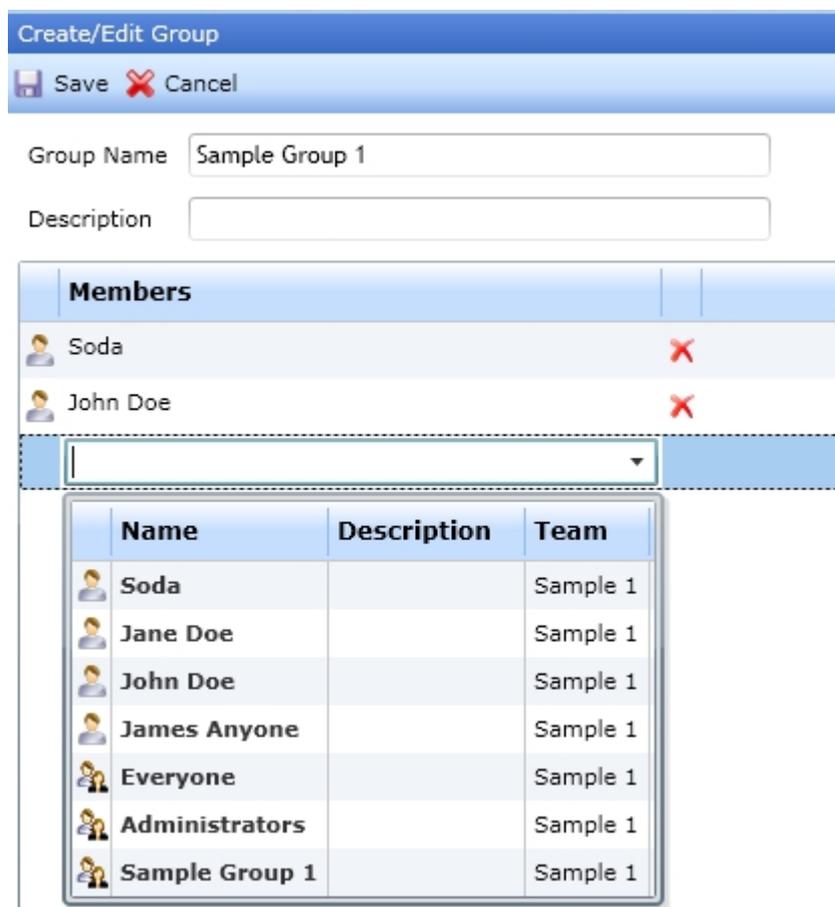
### Delete a usergroup



Select one or more user groups and hit the "Delete" button. If you delete a group, the associated users will not be deleted.

Warning: You can not undo a delete operation

### Edit the content of a usergroup



Select the desired user group and modify. New members can be chosen from the list displayed. Click the "Save" button.

### Join team

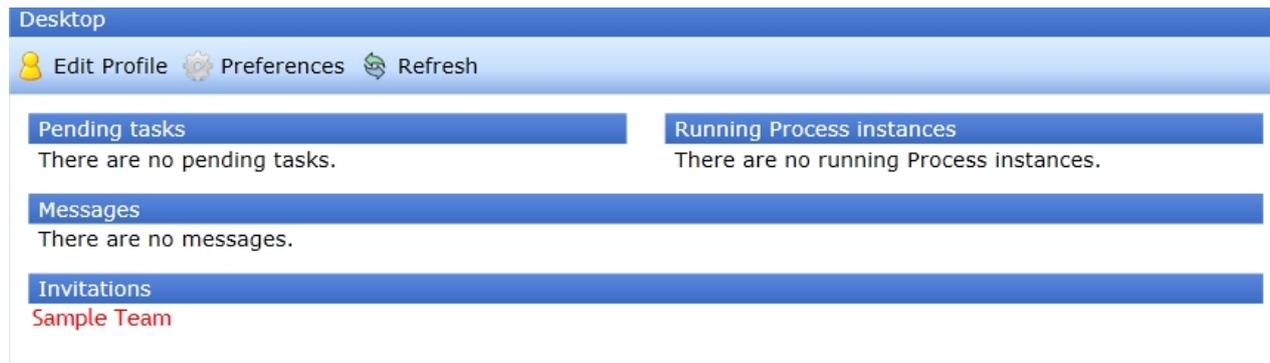
IYOPRO offers two ways to join a team

- [Join by invitation](#)  
A team administrator can invite you to participate at his team. You will then receive a notification email.

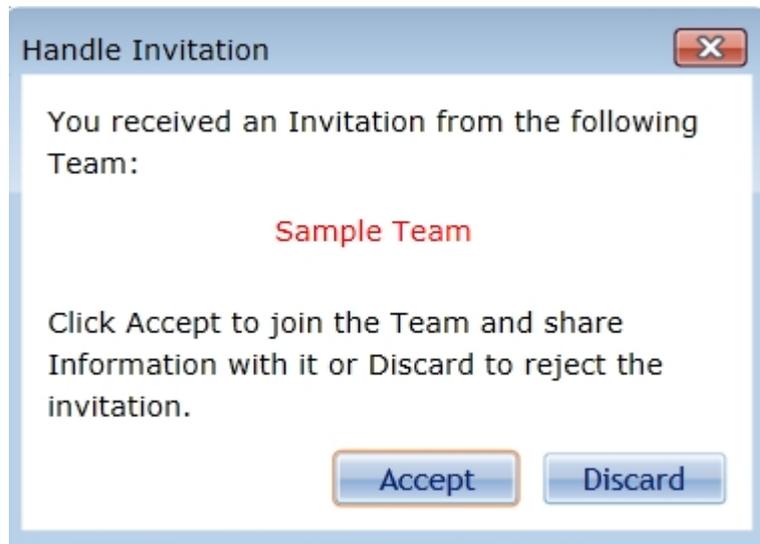
- You need to confirm the invitation on your [dashboard](#)
- [Join by Team Access Code](#)  
 Someone may have provided an IYOPRO Team Access Code to you. As long as the code is not expired you may join the team.

### Join by invitation

If you received an invitation to participate at a team open your dashboard. You will see the invitation in the invitations section.



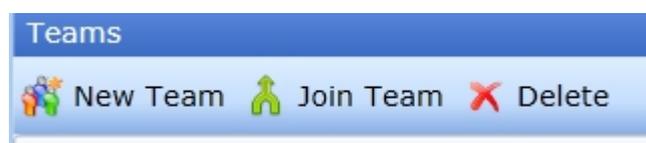
Click on the invitation and either accept or discard the invitation.



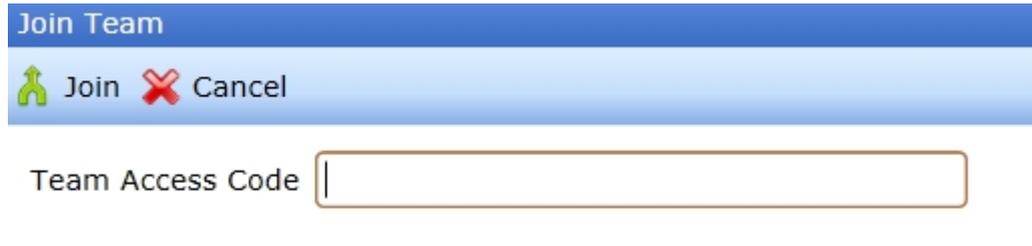
If you accepted the invitation the team and its resources will be shown below the teams node of the repository.

### Join by Team Access Code

If you received a [Team Access Code](#) to participate at a team open the teams node in the repository and click on "Join Team"



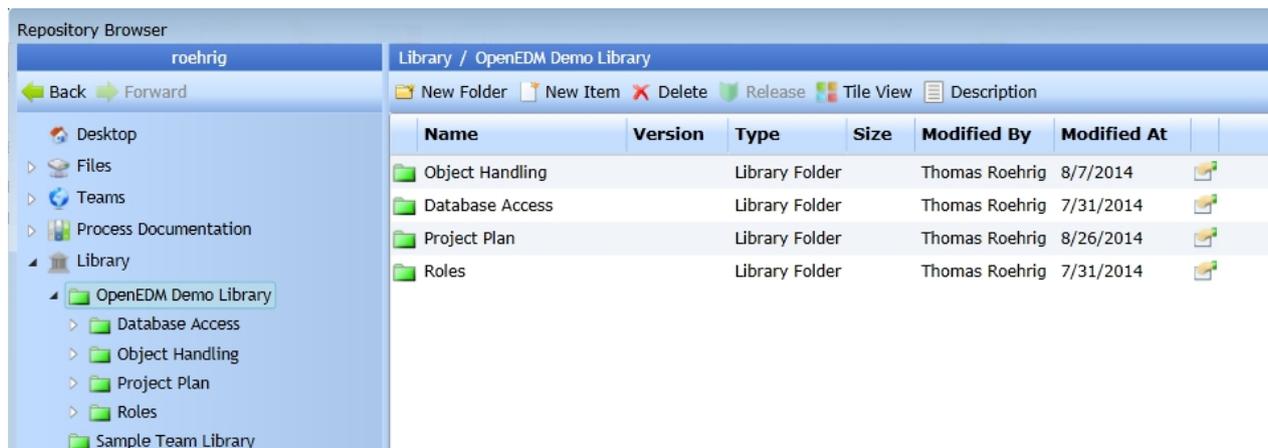
You will then be prompted to enter your Team Access Code.



When successful the team and it's resources will be shown below the teams node of the repository.

## Working with the Library

The Library is used to store predefined activities for later use in process models.



## Library node

The "Library" node is the root of all library elements. Library elements are associated to teams so the library node contains a folder for each team you're member of.

## Library folder

Each team has a library folder associated. This folder may contain library elements and for structural purposes also sub folders. Each library folder has the following functions available in the toolbar:



[Create a new folder](#)



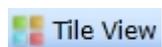
[Create a new library element](#)



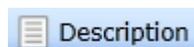
[Delete one or more elements](#)



Approve and Release library element



Switch to tile view



open description pane

## Creating Library Elements

In general it's possible to create library elements in two different ways:

- Create Library element from existing diagram element
- Create library element from scratch

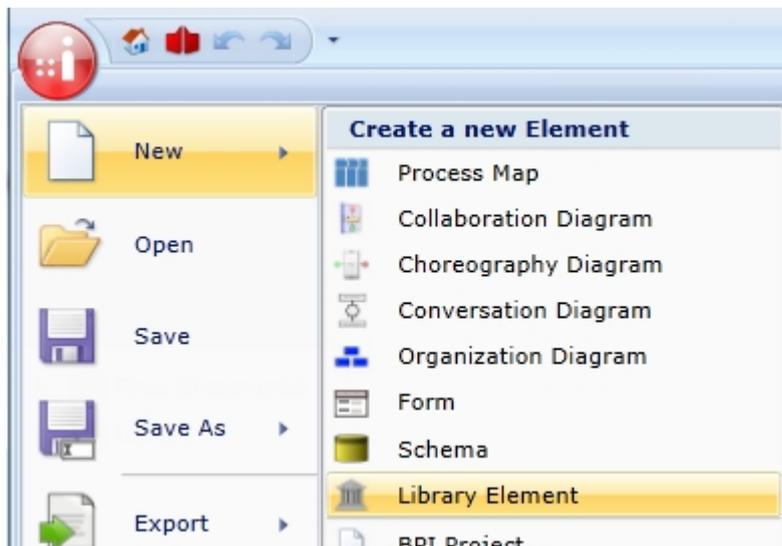
### Create Library element from existing diagram element

To create a library element out of an existing diagram element select the shape and choose "Add" from the Library group within the Modeling tab of the ribbon. The [library element editor](#) will appear.



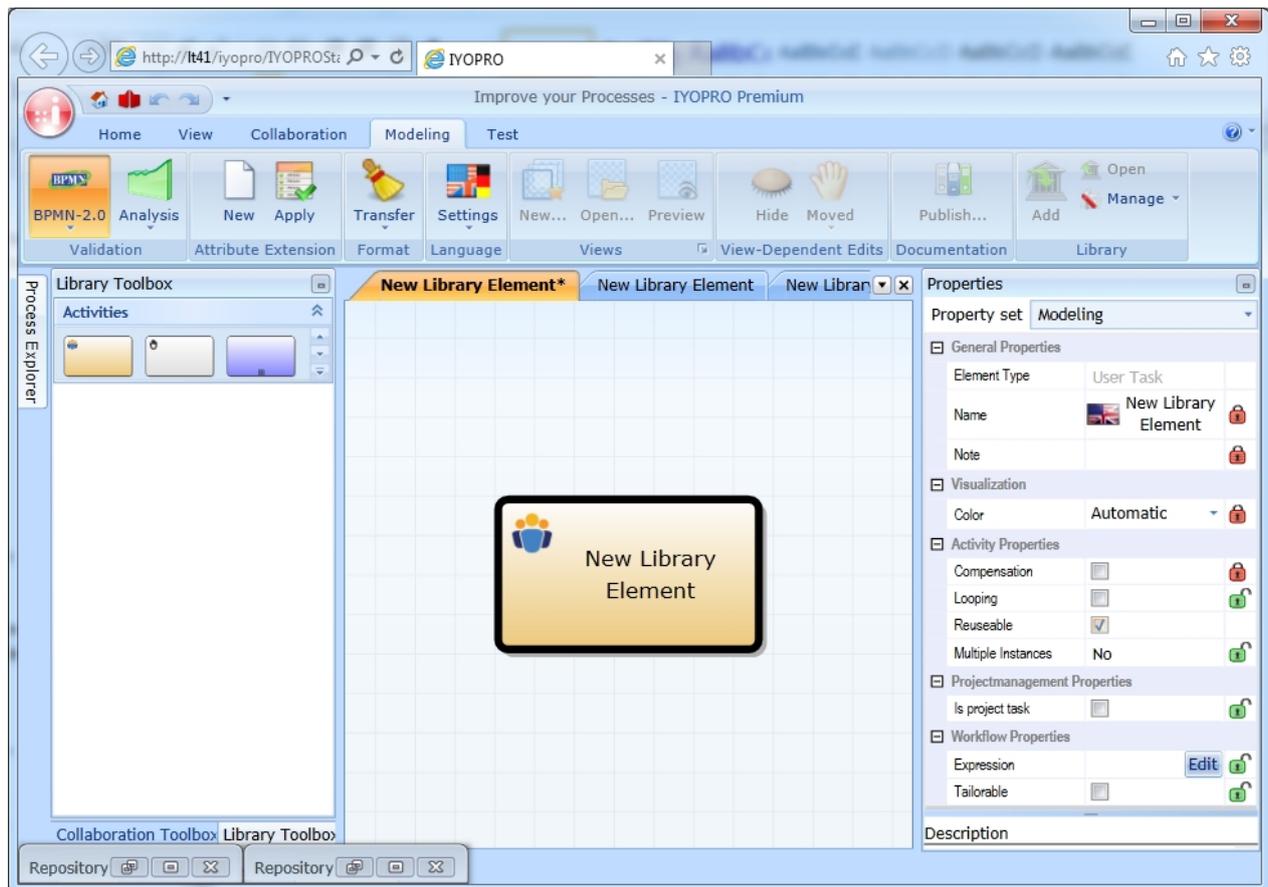
### Create library element from scratch

To create a library element from scratch choose new->Library Element from the application menu. The [library element editor](#) will appear. Drag the desired base activity from the library toolbox onto the editor.



## Editing Library Elements

To edit a library element either [create a new](#) one or open the library element from within the repository by doubleclick.



Library elements are mainly described by predefined properties. The library editor allows you to define, which properties can be edited after the library element has been inserted into a diagram and which not.

## Locking and unlocking properties

To prevent a property from being changed within a process diagram close the lock of that property. The icon becomes a red closed lock.

To allow a property being changed within a process diagram open the lock of that property. The icon becomes a green open lock.

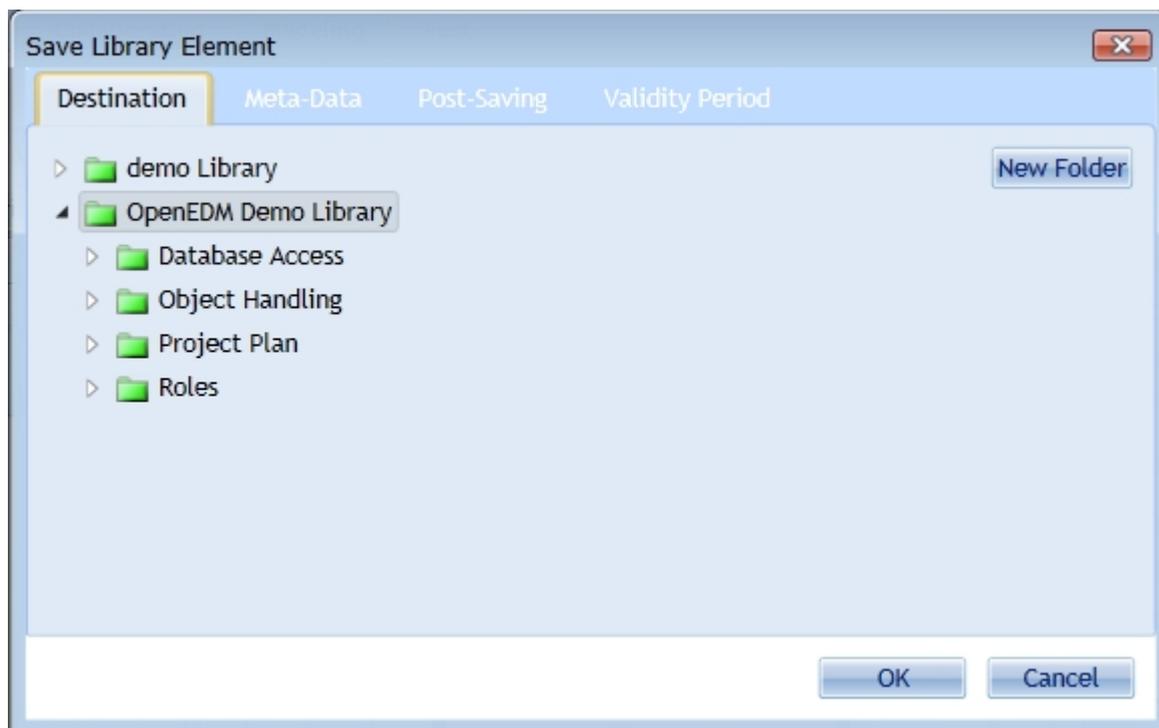
## Saving Library Elements

To save a library element click on the save button of the ribbon within the [library element editor](#).

The Save Library Element dialog appears. It contains out of four tabs which control the save and release process.

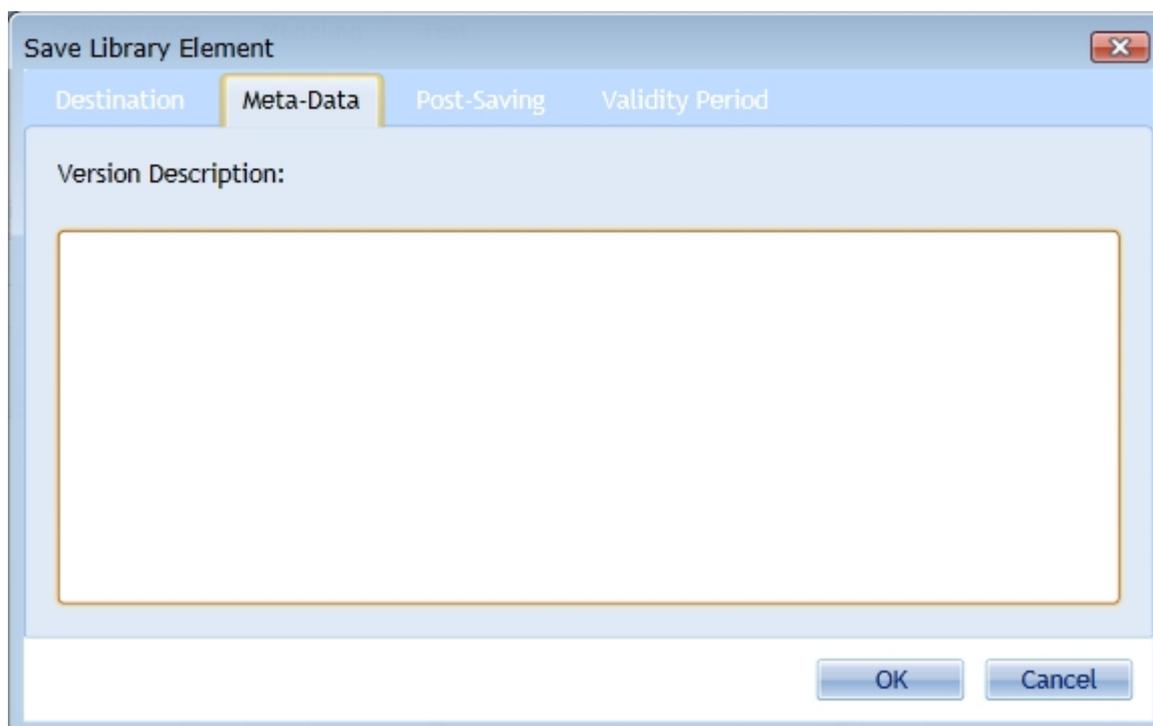
## Library Element Destination

In the first tab specify the library element destination. Navigate through the library folders and specify an appropriate destination. Choose the new folder button to create a new folder if required.



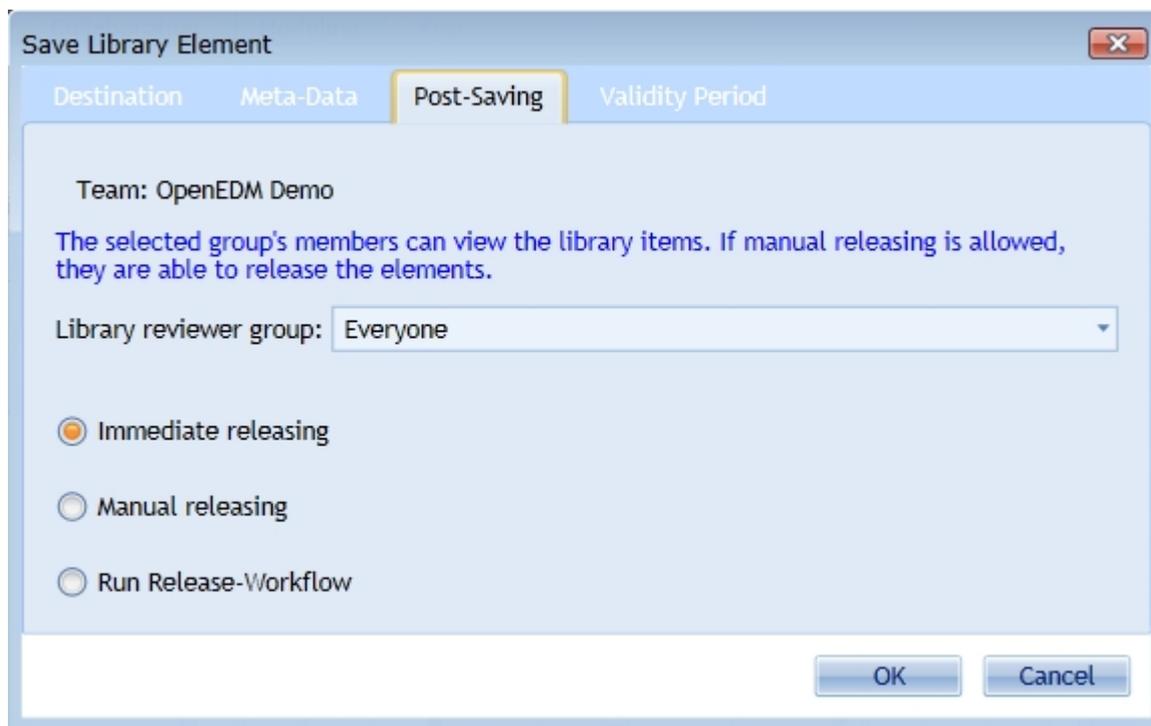
### Editing Version Description

To edit the version description select the Meta-Data tab and type in your description.



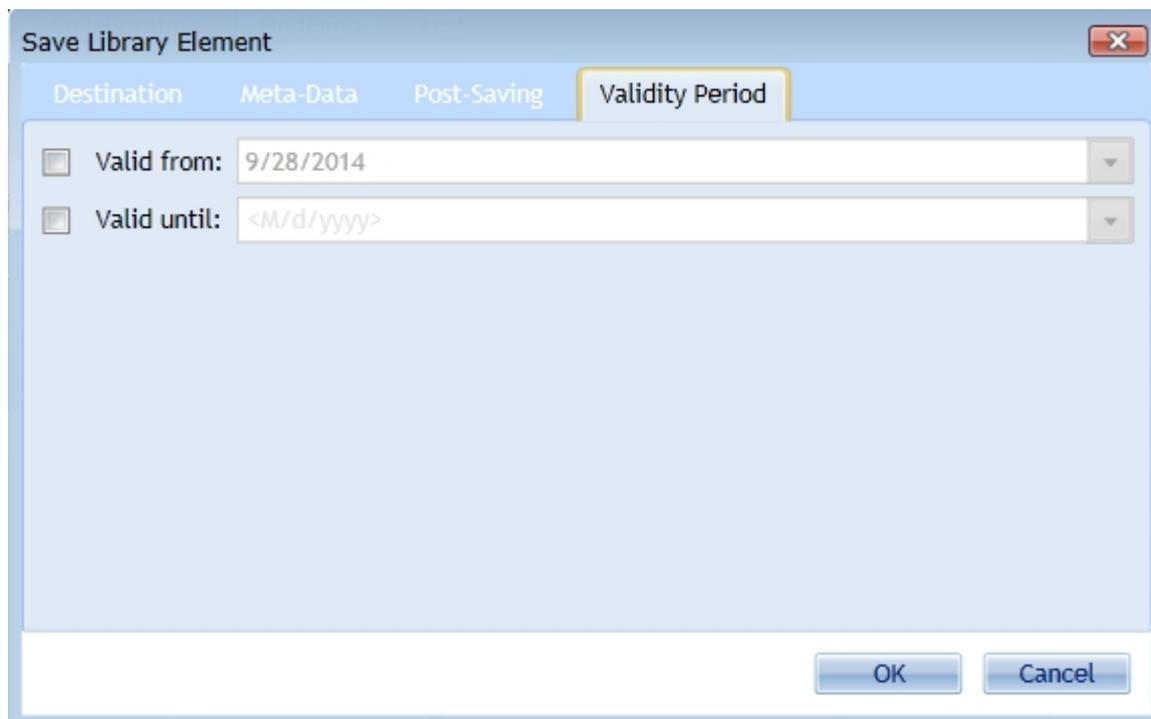
### Deployment and Release

Specify how the deployment process will be archived. This setting is affected by your [team settings](#)



## Validity Period

Specify the lifetime, when the library element availability will be in effect.



If nothing other defined, the library element will be valid from the moment of package release without end time. You can specify other times as necessary.

## Using Library Elements

### Use existing Library element in your Diagram

Open the [Process Explorer](#) and expand the Library node. Navigate to the desired library element and drag it onto your diagram. If you drag the library element over an existing activity and drop it IYOPRO will replace the

existing element with the library element.

You can also navigate to the library element using the repository. Even here you can click and drag the library element onto your diagram.

### Open Library Element out of a diagram

To open a library element within the [library element editor](#) select the library element on your diagram. Navigate to the Modeling tab of the ribbon and click the Open button of the Library group.

### Highlight Library Elements within a diagram

To highlight all library elements within a diagram navigate to the Modeling tab of the ribbon and click the Manage->Highlight button of the Library group.

### Refresh Library Elements within a diagram

If you modify library elements you may wish to refresh substitutions in your diagrams. To do so navigate to the Modeling tab of the ribbon and click the Manage->Refresh button of the Library group. Note that if you open a diagram which contains library elements the elements will always be up do date and do not need to refresh.

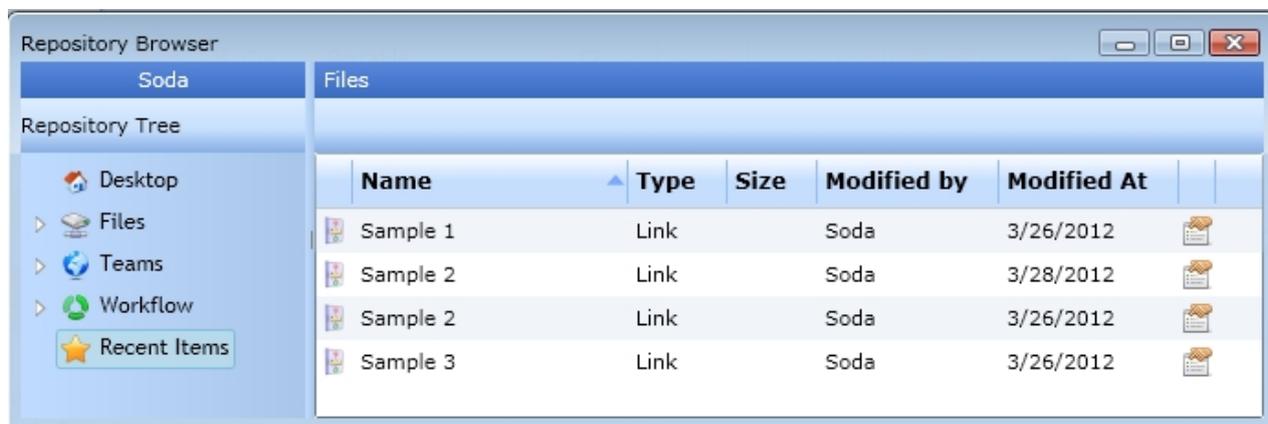
### Find Library Element utilization

To find diagrams, which utilize a specific library element select the library element on your diagram. In the Modeling tab of the ribbon click the Manage->Where Used button of the Library group.

### Find Library Element in Process Explorer

To find a specific library element in the Process Explorer select the library element on your diagram. In the Modeling tab of the ribbon click the Manage->Select in Process Explorer button of the Library group.

## Recent items



In "Recent Items" the objects accessed lately are displayed.

## The Portal

There are two portal versions: the web portal (IYOPRO Web) and the internal portal of the IYOPRO App.

If you are working with the IYOPRO App, you can either find the internal portal as part of the [Repository](#) or you can open one of the two portal versions by clicking on the Portal button in the [ribbon](#) menu:



Which one of the portal versions opens depends on your [preference settings](#). The default is IYOPRO Web.

If you are only working with the portal, simply [open IYOPRO Web](#).

### The app internal portal

The IYOPRO Portal provides information to the following sections:

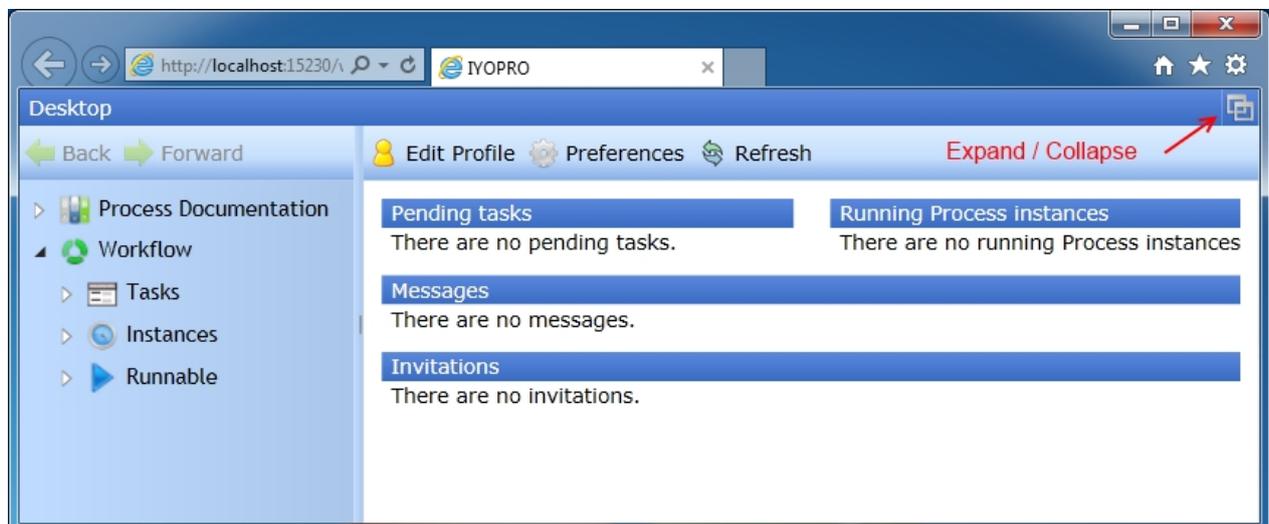
- business process documentation
- workflow automation

To open the portal click on the Portal button in the [ribbon](#) menu or open the [Repository](#).



Note, the Portal button may open the [web portal](#) if you didn't change your [preference settings](#).

Note, that the portal may open as Window within the full featured IYOPRO user interface. To expand / collapse the portal to / from the browser window click on the expand / collapse button in the upper right corner.



The YOPRO Portal consists out of three root nodes in the left navigation pane. These are

- [Desktop](#)  
The desktop contains an overview of your current session as well as options to customize your user profile and your preferences.
- [Process Documentation](#)  
Navigate through this node if you wish to study the released process documentation of your organization.
- [Workflow](#)  
Navigate through this node if you participate in workflows and i.e. need to complete an activity.

## Desktop

The desktop node matches the desktop node of the repository which is explained [here](#).

## Process Documentation

The Process Documentation node of the YOPRO Portal provides access to approved and released business process documentation of your organization. Information, how to publish data to the portal can be found in the [Publishing and Deployment](#) section of the Business Process Modeling chapter.

Beneath the root node a [documentation folder](#) of each team will be shown.

## Overview Page

### Overview

The Overview Page is used to present an overview of the process documentation to the user. These are:

- History  
The history lets the user navigate through different versions of the process model along a time line. View processes as they were in the past, as they are now and as they will be in the future.
- Entry Points  
Root entries for several process documentations
- News  
Information about the latest changes at the process documentation

Date	Type	Name	Released By
4/4/2014	New Document	Pricing	Thomas Roehrig
4/4/2014	New Document	Fixing marketing target	Thomas Roehrig
4/4/2014	New Document	Market & environment analysis	Thomas Roehrig
4/3/2014	New Document	Overview	Thomas Roehrig
4/3/2014	New Document	Process Map	Thomas Roehrig

To navigate through the documentation either expand / collapse / activate the nodes in the navigation menu or double click on the tiles to open the [detail pages](#).

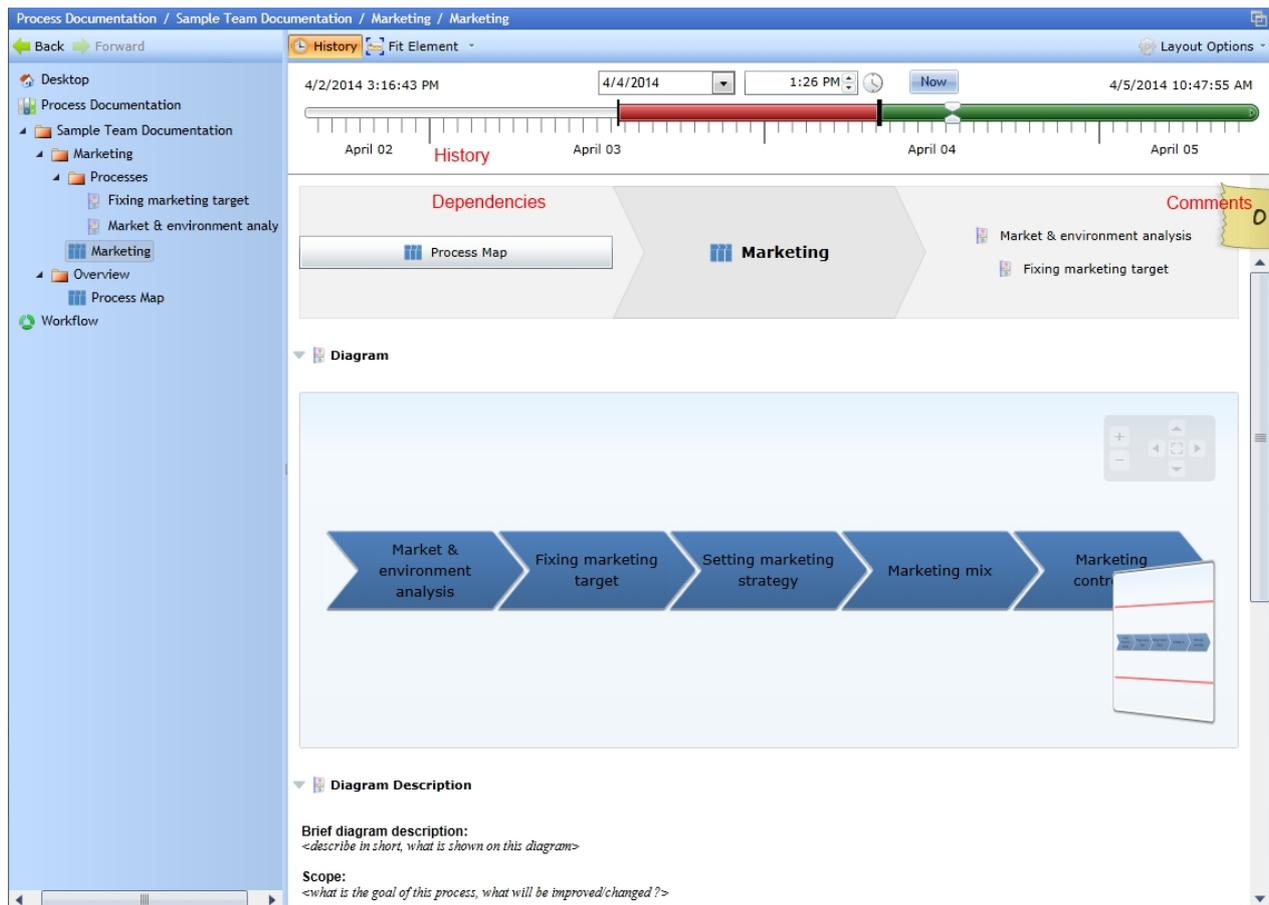
To show the [news](#) pane toggle the "Show News" button.

## Detail Page

## Overview

Detail pages are used to present details of a process to the user. These are:

- History  
The history lets the user navigate through different versions of the process model along a time line. View processes as they where in the past, as they are now and as they will be in the future.
- Dependencies  
The dependencies navigator visualizes parent child relationships between processes and/or landscapes
- Comments  
The comments area allows users to add or view comments about the diagram
- Diagram  
The diagram provides a graphical representation of the process model
- Diagram Description  
The diagram Description provides a textual representation of the process model
- Diagram Elements  
The diagram elements section provides detailed information about the items of the diagram



## Layout options

Layout options allow you to adjust the visualization behavior of detail pages according your needs.

- Open Elements as pop up  
If set (default), the details of an element will be shown in a pop up window when selected.  
If not set, the element will be highlighted and the details will be shown in the element description section below the diagram.

## History

The history control lets you navigate through different versions of the process model along a time line. You can view processes as they where in the past, as they are now and as they will be in the future.

To use the history ensure, that the history button of the details view is activated.



The time line visualizes timestamps, when the validity of a process model has changed, with a vertical black line. Between two of these lines or until the infinite end (a rounded edge) a timespan in a defined color is shown.

The color of a timespan represents the validity. The following colors are used:

- Red  
The process model is no longer in effect.
- Green  
The process model is currently in effect

- Yellow  
The process model will take effect in the future

## Navigation

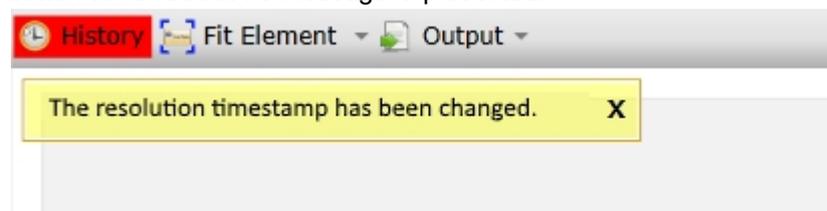
To navigate on the time line you can

- Select a time stamp from the date / time picker control
- Click the Now button to jump to the current time stamp
- Move / Click the slider on the time line
- Pan the time line by dragging the time line with the left mouse button
- Change the time line scale with the mouse wheel

As you can see the process documentation will update immediately if you have changed the resolving time.

## Outdated resolution timestamps

In case IYOPRO automatically opens a diagram with an outdated timestamp the user is informed about it and the history button turns red. If a user changes the viewed timestamp to an outdated date the history button turns red but no message is presented.



## Dependencies

The dependencies control let you navigate to process models, which uses the current model and which are used by the current model.

The current model is shown in the center of the control.

Elements, which use the current model are listed on the left. Elements that are used by the current model are listed on the right side.



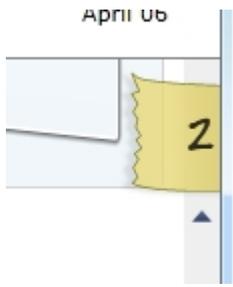
## Navigation

To navigate through the dependencies simply click on the desired element.

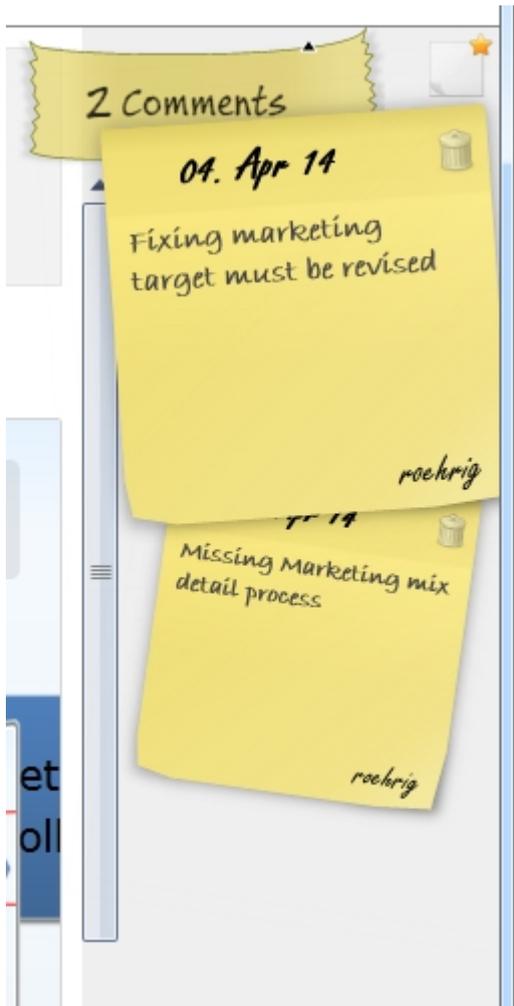
## Comments

Comments allow the reader of the process documentation to leave comments on diagrams and activities.

How many comments have already been written can be seen at the comment strip on the right side.



To show all comments click on the strip and the comment panel will appear. To hide the comments click on the strip again.



### Add comment

To add a comment click on the add button at the top of the comment panel. A new comment will appear. Type in your comment. It will be saved automatically.

### Delete comment

To delete a comment click on the trash on the comment sheet.



Important notice:

Note, that comments will not be deleted. They will be terminated what means, that they will be visible if the current [history](#) time stamp is set appropriate.

Changes within 10 minutes at the same note will be treated as single edit.

## Diagram

The diagram represents a graphical description of the process model.

### Diagram



## Zoom, Pan, Fit

To zoom, pan and fit the view use the [view manipulation functions](#) which are also available in the IYOPRO modeler.

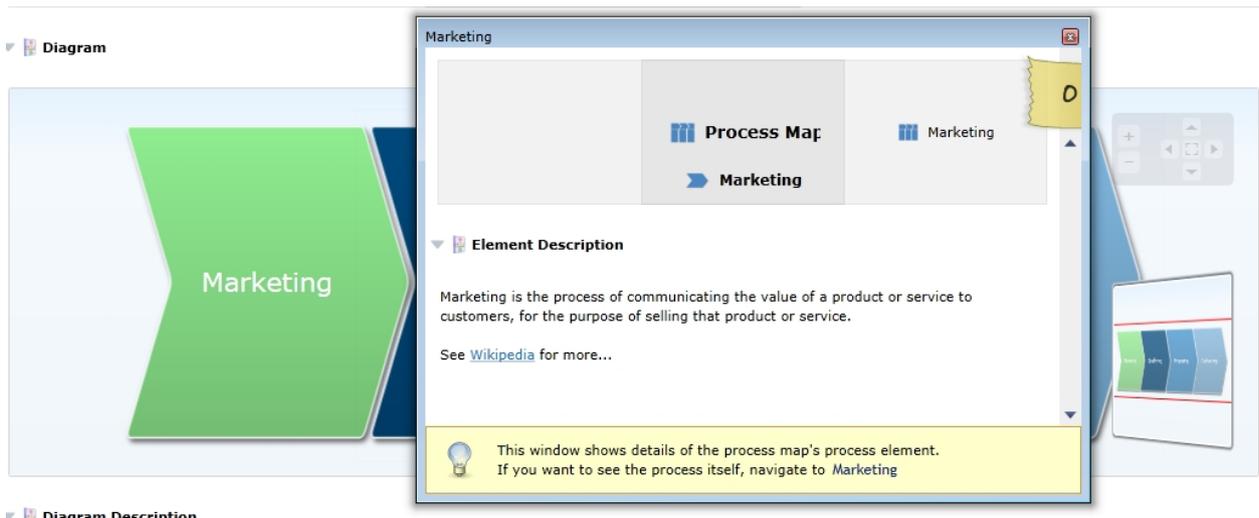
## Navigation

To navigate to process models, which are linked to the shapes, double click on the shape.

## Element details

Element details are listed in the elements table underneath the diagram description.

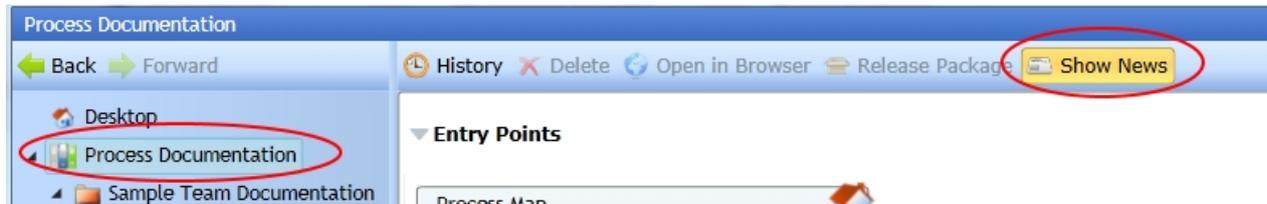
Based on your current [layout options](#) there are two different ways to visualize element details, either as popup or within the page.



## News

If you select the Process Documentation root node you can enable the news pane by checking the Show

News button.



The news pane is located at the bottom of the window.

Date	Type	Name	Released By
4/4/2014	New Document	Pricing	Jeff Jones
4/4/2014	New Document	Fixing marketing target	Jeff Jones
4/4/2014	New Document	Market & environment analysis	Jeff Jones
4/3/2014	New Document	Overview	Jeff Jones
4/3/2014	New Document	Process Map	Jeff Jones

The news pane will show all

- diagrams which have been added to the process documentation
- subscribed diagrams that have been changed
- comments on subscribed diagrams that have been created / changed

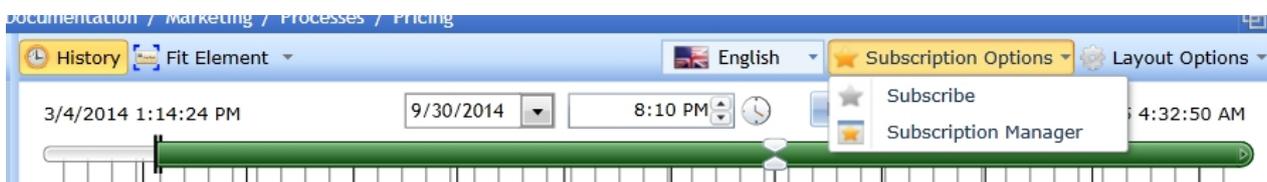
## Subscription

To get informed, if a process model has been changed every portal user may subscribe individual process models.

Once subscribed the user receives notes about changes within the Process Documentation root node.

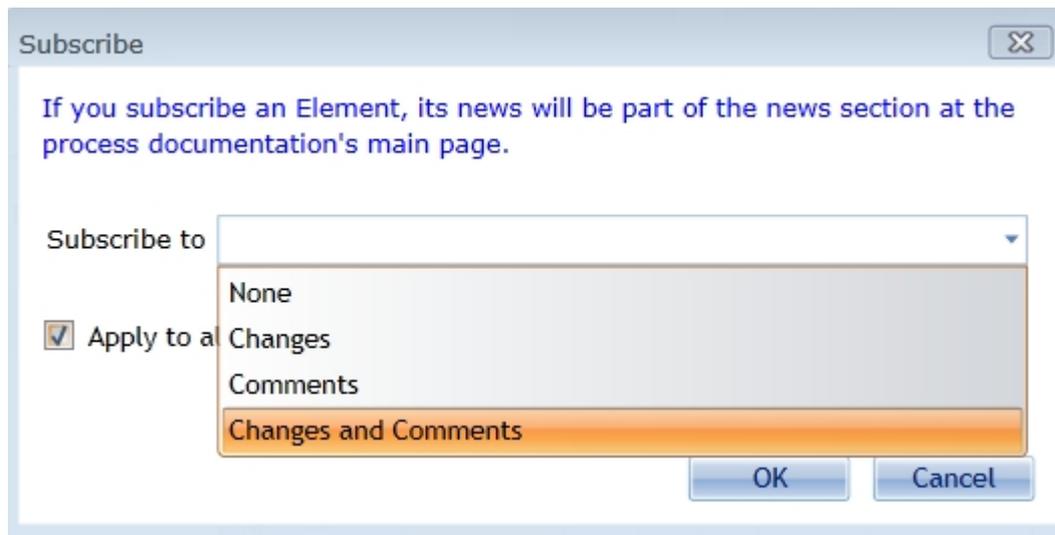
## Subscribe

To subscribe a process model locate the desired process within the process documentation. Choose Subscription Options->Subscribe to subscribe the process model.



In the upcoming dialog choose the level of details of your subscription:

- Changes  
Subscribe to process model changes
- Comments  
Subscribe to Comments that have been created or changed



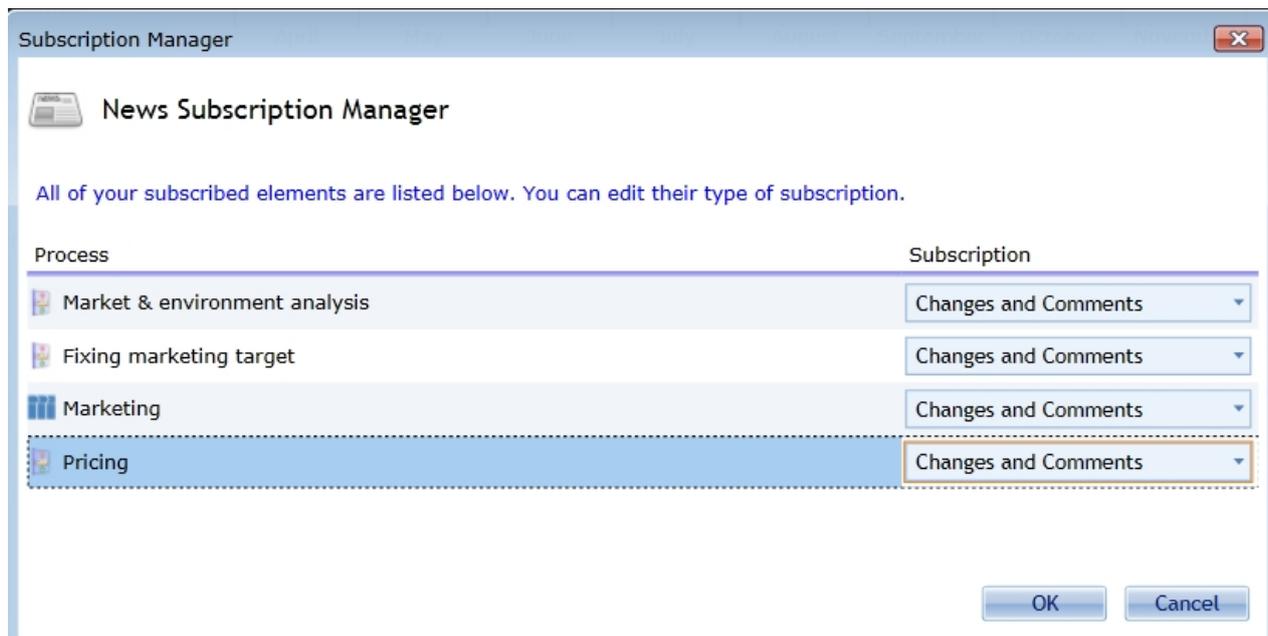
Toggle the "Apply to all sub elements" button if you not wish to subscribe to sub processes as well.

## Unsubscribe

To unsubscribe a previously subscribed process model choose Subscription Options->Unsubscribe and follow the instructions.

## Subscription Manager

The Subscription Manager is used to manage all subscriptions and adjust subscription settings. To open the subscription manager choose "Subscription Options->Subscription Manager"



## Output

Published diagrams and their views can be exported to provide process related information outside of the portal.



## Export diagram

To export a diagram choose the output options and select the favored export format. Diagrams are exported as they are displayed so the currently chosen language and the currently selected view will be used in the export.

There are two available formats:

- Print

This export is similar to the print functionality in the [application menu](#).

- Word Export

This export is similar to the [Word export](#) in the modeling environment.

## Workflow

The Workflow node of the IYOPRO Portal provides access to the workflows of your organization.

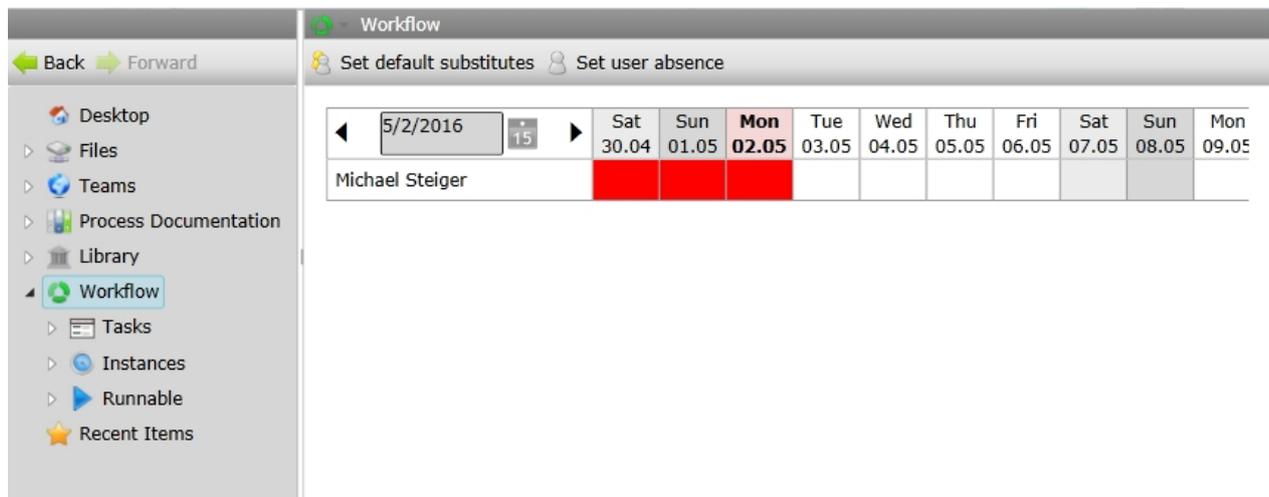
Beneath the root node the following sub nodes exist:

- [Tasks](#)  
This module provides information about open and / or finished tasks of the current user
- [Instances](#)  
This module provides information about process instances, the current user is involved in.
- [Runnable](#)  
This module provides all runnable workflows which may be utilized by the current user.

Repository Tree		Refresh				
		Name	Team	Created at	Created by	Workflow Id
Workflow	Tasks	ECR	Thomas Roehrig			10751
	Instances	ECR Develop sample	Thomas Roehrig			11187
	Runnable	Max Freigabe	Max Mustermann GmbH			10041
	Recent Items	Send Newsletter	Thomas Roehrig			10013

The Workflow node itself provides an overview of currently defined absence times of other users and provides the functionalities to

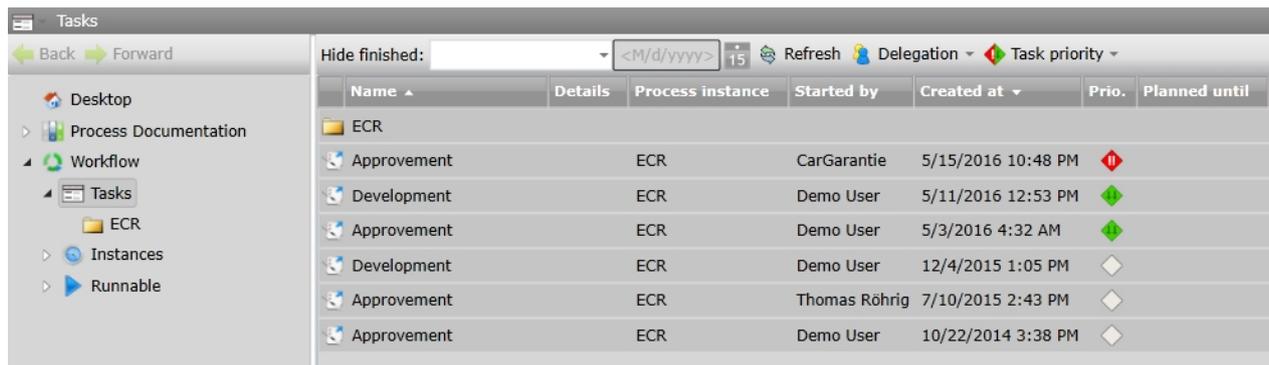
- [Set substitutes](#)
- [Set user absence](#)



## Tasks

The Tasks node provides access to open and / or finished tasks

If the Tasks node is activated, all tasks of the user are shown. Each task is associated to a process instance. For each process instance in addition a folder is shown beneath the tasks node as well as in the task list. If you navigate to such a folder, only tasks of that process instance type are shown. This makes it easy to work on same incidents.



## Task Availability

A various number of user accounts can participate in a workflow task but only one account at a time can be responsible. Therefore workflow tasks are displayed differently depending on their availability for the currently signed on user account.



(Task has no current responsible account or the current account is responsible)



(Another account is responsible for the task)

## Show / Hide finished tasks

In the toolbar of the dialog the "Hide finished" selection allows several useful and predefined options, when to hide finished tasks. These are:

- Immediate  
A task will be hidden immediately after it's finished
- Today  
Only tasks that have been finished today will be shown
- Yesterday  
Only tasks that have been finished since yesterday will be shown
- Last Week

- Only tasks that have been finished since last week will be shown
- Last Month  
Only tasks that have been finished since last month will be shown
- Last Year  
Only tasks that have been finished since last year will be shown
- Since...  
Only tasks that have been finished since a given date will be shown

## Delegate tasks

Click on the button 'Delegation' to delegate tasks and manage already delegated tasks.

## Set task priorities

Select one or more entries of the task list and click the button "Task priority" to define the urgency of those tasks. Tasks can be sorted by their priority.

## Configure the task list

The structure of the task list can be changed. Columns can be added or hidden and the order of displayed columns can be changed.

- To modify the displayed columns right click any column head and choose the needed column set.
- To change the order of displayed columns drag the column head and drop it at the desired position.

## Open a task outside IYOPRO

You may open a task outside IYOPRO. You have the following possibilities:

- Open the task in the web portal and click on the button "Undock"
- Open the task in the app portal
- Use the URL `<IYOPRO Client>/Form.aspx?taskid=<task guid>`  
Example: <https://www.iyopro.com/iyopro/Form.aspx?taskid=93908093-c8d1-465c-b260-cdaa425a48bc>  
(see also topic [Start Workflow outside IYOPRO](#))

## Substitutes

### Introduction

Some workflow tasks have to be finished in a certain time. If the actual responsible user is absent the task has to be done by another user to ensure the prompt handling. Therefore users can define substitutes in IYOPRO.

### Conceptual Overview

Users can individually define substitutes for their workflow tasks. Those substitutes will be able to complete tasks during the absence of the actual responsible person.

- [Set default substitutes](#)

Use this to define a standard substitute for all the tasks of a user in a certain team or a specific task

- [Set user absent](#)

Use this to set a user absent for a defined period of time

The substitute options can be found in the Repository and IYOPRO Portal beneath the [Workflow node](#).

### Set substitutes

### Define task substitutes

Workflow tasks can have specific substitutes. Those task substitutes can be determined in the related task forms. Press the button 'Configure substitutes' to open the configuration.

The following settings can be made:

- Filter  
Enter the name of a user to filter the results of available substitutes
- Green arrow  
Use this button to switch the status of selected users and groups from available to configured
- Red arrow  
Pressing the red arrow button will remove the selected users and groups from the configured substitutes back to available substitutes

## Define standard substitutes

Standard or default substitutes can be configured for every team. These determined substitutes will be responsible for every task while the actual user is absent. These standard substitutes will only be informed about tasks that do not have specific task substitutes.

Standard substitutes can either be configured for oneself or for someone else.

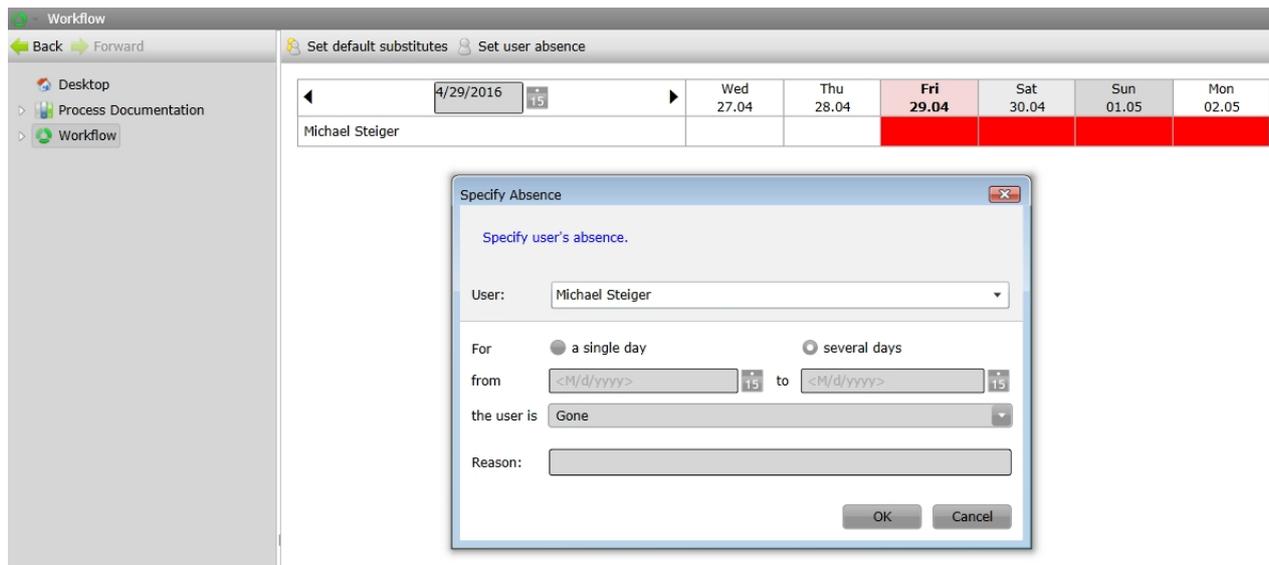
If you want to configure the substitutes for someone else, navigate to the [User node](#). There you can select a user and configure his or her substitutes. Of course, you can also select your own user.

Alternatively, to configure standard substitutes for yourself, you can navigate to the [Workflow node](#) or to a Workflow task.

The mechanism to set standard substitutes is equal to task substitutes which is described above.

## Set user absent

Navigate to the [Workflow node](#) in the Repository or IYOPRO Portal to set a user absent. Click on 'Set user absence' to open the absence settings.



The following absence settings can be made:

- User  
Choose the user who's absence status is supposed to be changed
- Absence duration  
Choose the date and duration of the absence status
- Status  
Decide whether the selected user is absent or present during the selected time
- Reason  
Use this to give a reason for the absence

Press 'OK' to save the absence entry. It will be displayed in the absence table which is shown in the figure above.

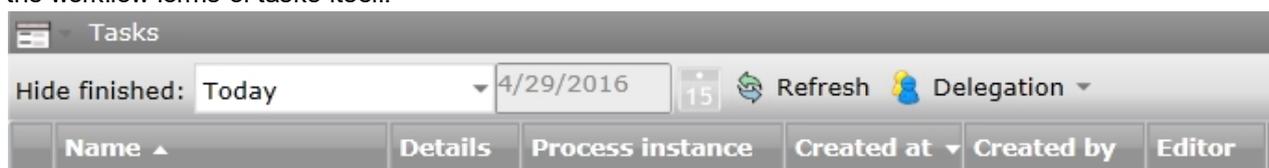
## Delegation

In some cases the actual responsible will not be able to handle a workflow task because of certain circumstances. This might happen because of limited time, missing knowledge or because the task was simply addressed to the wrong person. Therefore IYOPRO provides the delegation functionality.

Delegating a task means to hand over the responsibility for one or more workflow tasks to another person.

- Delegate task  
Use this to hand over the responsibility for the selected task to another user
- Revert task delegation  
Use this to decline the responsibility of a task that has been delegated to you and give it back to the sender
- Show delegated tasks  
This provides an overview of all delegated tasks of the user

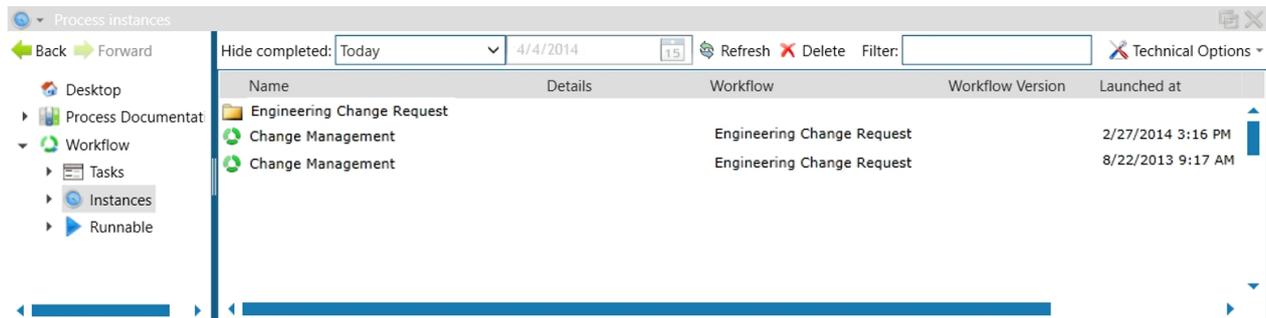
The delegation options can be found in the Repository and IYOPRO Portal beneath the [Tasks node](#) and on the workflow forms of tasks itself.



## Instances

The Instances node provides access to running or finished process instances

If the Instances node is activated, all instances in which the user is involved are displayed. Each instance is associated to a workflow. For each workflow in addition a folder is shown beneath the instances node as well as in the instances list. If you navigate to such a folder, only instances of that workflow are shown. This makes it easy to work on same incidents.



## Show / Hide finished instances

In the toolbar of the dialog the "Hide finished" selection allows several useful and predefined options, when to hide finished instances. These are:

- Immediate  
An instance will be hidden immediately after it's finished
- Today  
Only instances that have been finished today will be shown
- Yesterday  
Only instances that have been finished since yesterday will be shown
- Last Week  
Only instances that have been finished since last week will be shown
- Last Month  
Only instances that have been finished since last month will be shown
- Last Year  
Only instances that have been finished since last year will be shown
- Since...  
Only instances that have been finished since a given date will be shown

## Delete process instances

If you wish to delete one or more process instances select them and click on "Delete" This will delete all process instance data, associated process instance attributes, files and tasks. Deleting a running process instance will stop all activities immediately.

## Technical Options

### Download log

You can download the full process instance runtime informations for one or more process instances out of IYOPRO as Excel® report.

### Show Queues

You can [monitor the workflow queues](#) with the workflow tasks and scheduled workflows.

## Configure the instance list

The structure of the instance list can be changed. Columns can be added or hidden and the order of displayed columns can be changed.

- To modify the displayed columns right click any column head and choose the needed column set.
- To change the order of displayed columns drag the column head and drop it at the desired position.

## Monitor process instances

To monitor process instances follow [this guide](#).

Excel is a registered trademark of Microsoft Corporation in the United States and/or other countries.

## Runnable

The Runnable node provides access to runnable workflows.

If the runnable node is activated, a list of all workflows (Production Version) that the user could start is displayed. Each workflow is associated to a team. For each team, a folder is also displayed below the runnable node as well as in the workflow list. If you navigate to such a folder, only workflows of that team are shown. This facilitates work on the same incidents.



The screenshot shows a web interface for managing workflows. On the left is a navigation pane with a tree view containing 'Desktop', 'Process Documentation', 'Workflow', 'Tasks' (with a sub-item 'ECR'), 'Instances', and 'Runnable'. The 'Runnable' node is selected. The main area displays a table of workflows under a 'Sample Team' folder. The table has columns for Name, Team, Created at, Created by, Workflow Id, and Tags. A 'Refresh' button is located above the table.

Name	Team	Created at	Created by	Workflow Id	Tags
Sample Team					
Anonymous Web Form	Sample Team	2/15/2013	Frank Schusdziarra	55290	
Engineering Change Request	Sample Team	9/25/2014	Thomas Röhrig	173065	
On Premise Installation Tracker	Sample Team	12/19/2013	Thomas Röhrig	99048	
Send Newsletter	Sample Team	5/10/2012	Thomas Röhrig	37606	
Subscribe Newsletter	Sample Team	5/10/2012	Thomas Röhrig	37604	
Wartungseinsatz	Sample Team	3/8/2013	Thomas Röhrig	56898	
Word Document Approval	Sample Team	9/29/2014	Thomas Röhrig	173625	

Double click on the appropriate workflow to start.

Then either a web version of your [start form](#) will appear or the IYOPRO Start Workflow Dialog, depending on the settings in the workflow definition.

If a start form has been declared, a process instance will be created when the form is submitted.

## Configure the workflow list

The structure of the workflow list can be changed. Columns can be added or hidden and the order of displayed columns can be changed.

- To modify the displayed columns right click any column head and choose the needed column set.
- To change the order of displayed columns drag the column head and drop it at the desired position.

## Start Workflow Dialog

The Start Workflow Dialog is used for all types of workflows, which must be manually started.

It shows the name and some organizational information as well as a preview image and a description.

Temperature Monitor



Temperature Monitor



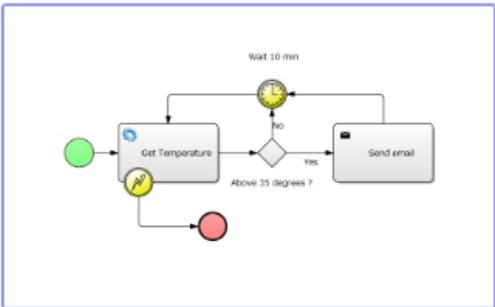
This process is used to monitor the temperature of a heating enging. If the temperature exceeds 35 degree a warning mail will be send.

Type: Workflow

Size:

Created: Friday, September 07, 2012 2:20:25 PM by Thomas Roehrig

Modified: Friday, September 07, 2012 2:20:25 PM by Thomas Roehrig



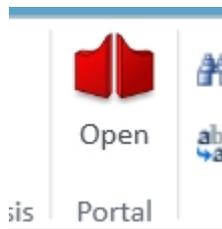
To start such a workflow hit the "Start" button.

## The web portal (IYOPRO Web)

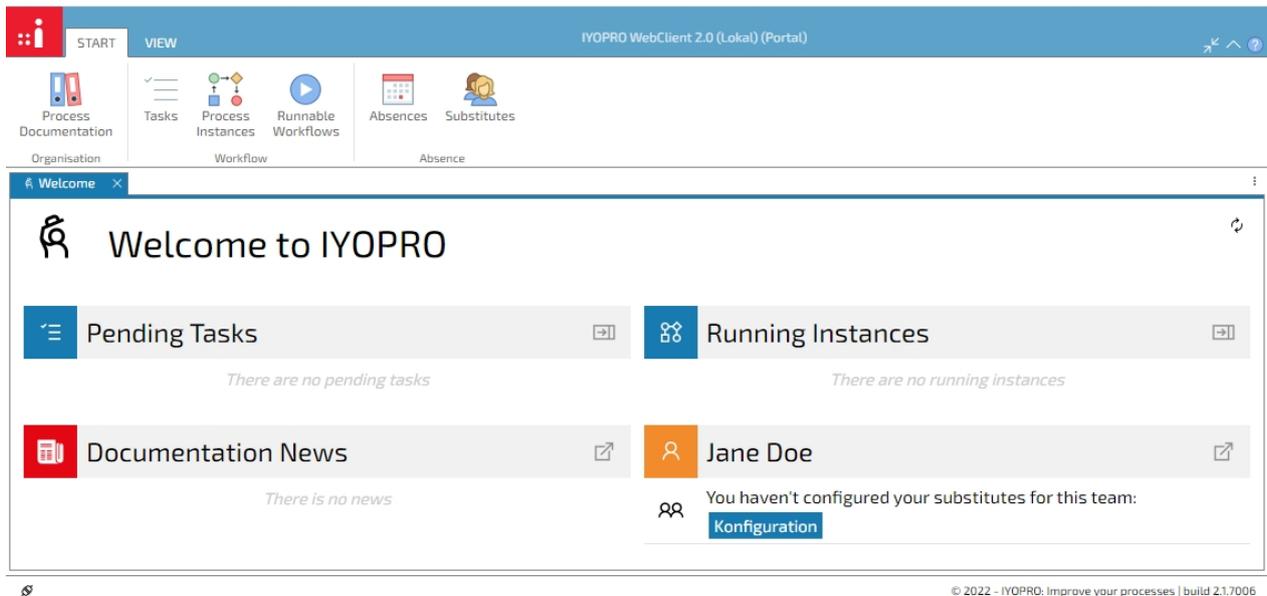
The IYOPRO Portal provides information to the following sections:

- business process documentation
- workflow automation

To open the portal click on the Portal button in the [ribbon](#) menu of IYOPRO App or [open IYOPRO Web](#) directly.



Note, the Portal button may open the [internal portal](#) if you changed your [preference settings](#).

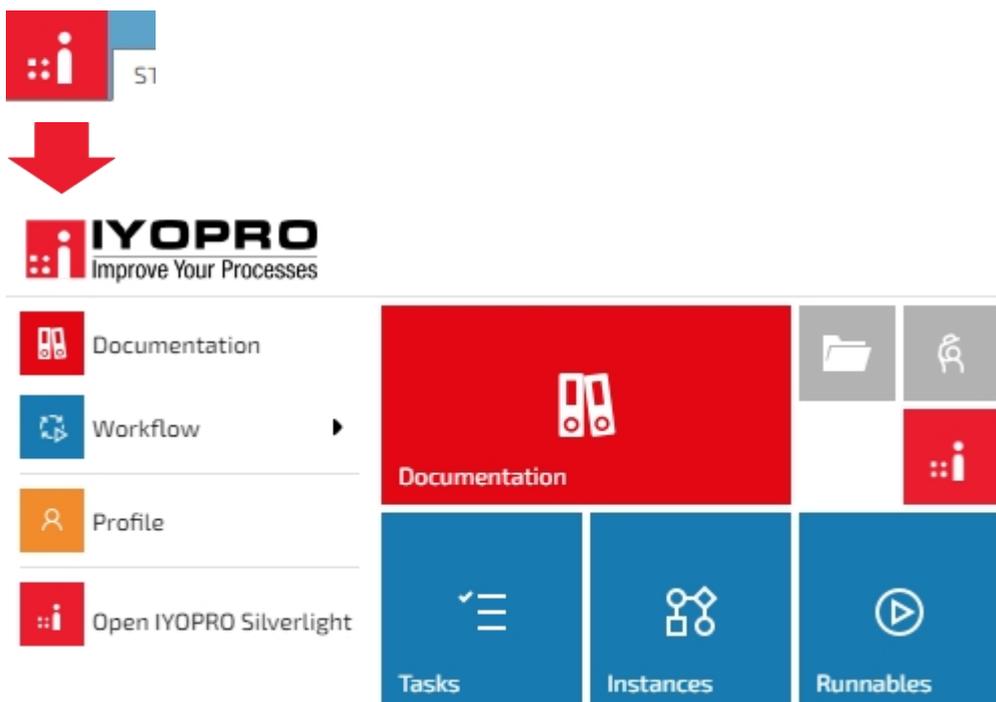


The IYOPRO Portal consists out of three main sections. These are

- [Welcome page / Dashboard](#)  
The welcome page provides an overview of the current status in your workflow and documentation area as well as additional information related to you and your teams.
- [Process Documentation](#)  
Navigate through this section if you wish to study the released process documentation of your organization.
- [Workflow](#)  
Use this section if you participate in workflows and i.e. need to complete an activity.

In addition, you can

- Customize your UI using the options under the "View" tab
- [Edit your user profile](#)
- View [files stored in the Repository](#)



## Dashboard

 Welcome to IYOPRO 

 Pending Tasks  <i>There are no pending tasks</i>	 Running Instances  <i>There are no running instances</i>
 Documentation News  <i>There is no news</i>	 Jane Doe  You haven't configured your substitutes for this team: <a href="#">Konfiguration</a>

This screen informs you about current activities.

- Pending Tasks - Overview of your current tasks
- Running Instances - Overview of your monitored running process instances
- Documentation News - Overview of messages regarding the documentation such as comments or new or altered documents
- Your User - Pending invitations to other teams, absences or information about teams for which you haven't configured substitutes

Note: You can click on the headers and open the respective component.

## Process Documentation

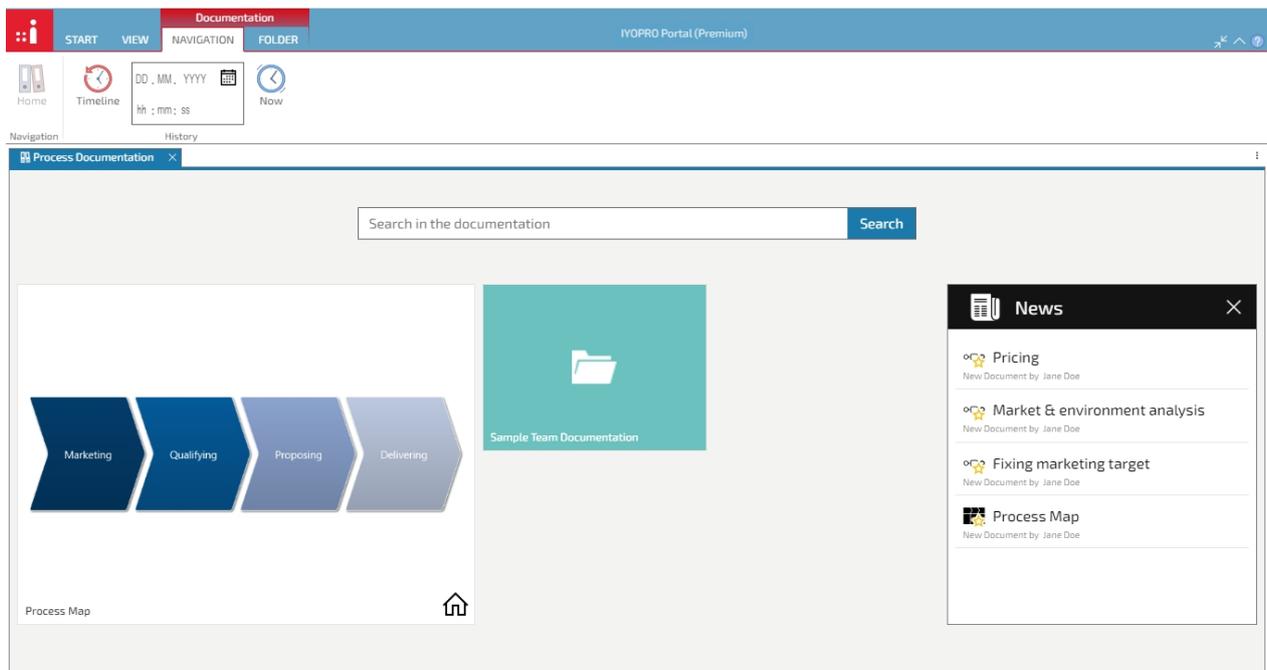
The "Process Documentation" button of the IYOPRO Web Start menu provides access to approved and released business process documentation of your organization. Information, how to publish data to the portal can be found in the [Publishing and Deployment](#) section of the Business Process Modeling chapter.

### Overview Page

#### Overview

The Overview Page is used to present an overview of the process documentation to the user. These are:

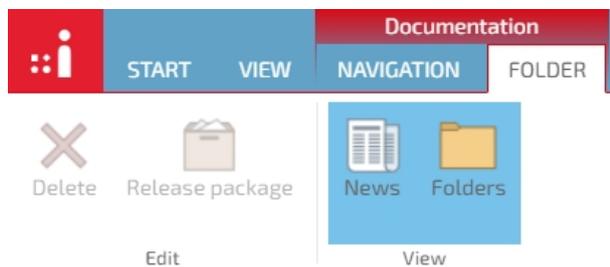
- History  
The history lets the user navigate through different versions of the process model along a time line. View processes as they were in the past, as they are now and as they will be in the future.
- Entry Points  
Root entries for several process documentations
- News  
Information about the latest changes at the process documentation



To navigate through the documentation, double click on the tiles. You can either navigate through the folders or through the entry points to open the [detail pages](#).

To show the [news](#) pane toggle the "News" button in the Folder menu. There you can also activate or deactivate the display of folders.

In addition, you have the option to [manually release a package](#) or to delete selected files / folders.



## Detail Page

### Overview

Detail pages are used to present details of a process to the user. These are:

- **History**  
The history lets the user navigate through different versions of the process model along a timeline. View processes as they were in the past, as they are now and as they will be in the future
- **Dependencies**  
The dependencies navigator visualizes parent child relationships between processes and/or landscapes
- **Comments**  
The comments area allows users to add or view comments about the diagram
- **Diagram**  
The diagram provides a graphical representation of the process model
- **Properties**  
The properties section provides detailed information about the diagram and the elements of the diagram. This includes, for example, the diagram description as a textual representation of the process model
- **Subscription**  
Subscribe process models that are of interest to you so that you get informed about changes to these

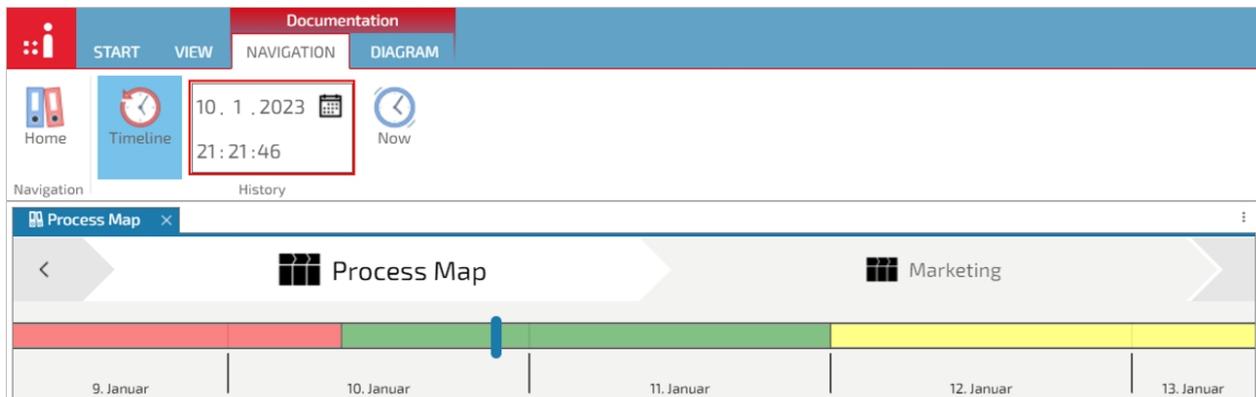
## process models



## History

The history control lets you navigate through different versions of the process model along a timeline. You can view processes as they were in the past, as they are now and as they will be in the future.

To use the history, ensure that the "Timeline" button in the Navigation menu is activated.



If the validity of a process model has changed, the timeline visualizes this timestamp with a vertical black line. Between two of these lines or until the infinite end, a timespan is shown in a defined color.

The color of a timespan represents the validity. The following colors are used:

- Red  
The process model is no longer in effect
- Green  
The process model is currently in effect
- Yellow  
The process model will take effect in the future

## Navigation

To navigate on the timeline you can

- select a timestamp from the date picker control
- click the "Now" button to jump to the current timestamp
- click somewhere on the timeline to change to that date / time
- pan the timeline by dragging the timeline with the left mouse button
- change the timeline scale with the mouse wheel

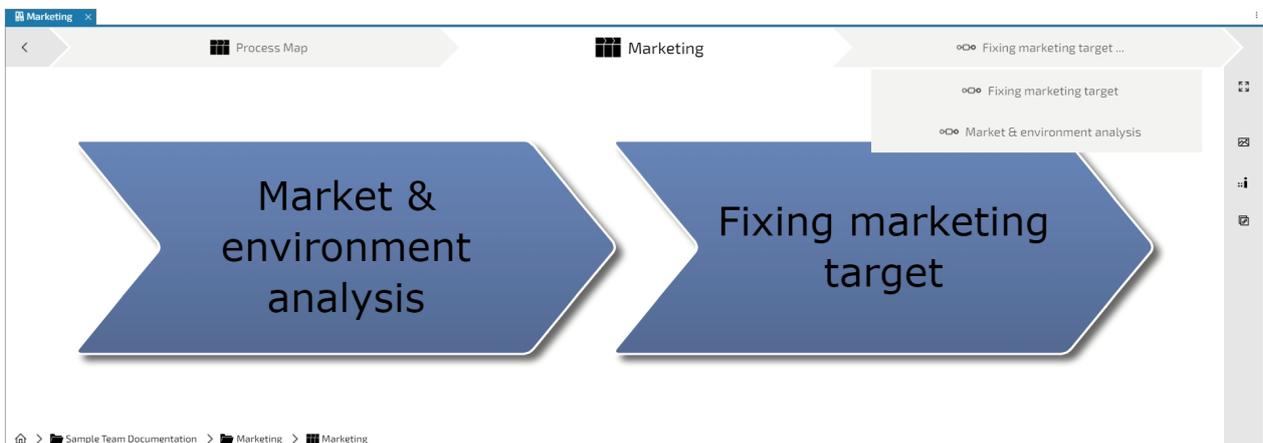
As you can see, the process documentation is updated immediately after you change the resolving time. Also, if you view a date / time other than now, the date control will be outlined in red.

## Dependencies

The dependencies control lets you navigate to process models which use the current model and which are used by the current model.

The current model is shown in the center of the control.

Elements which use the current model are listed on the left. Elements that are used by the current model are listed on the right side.



## Navigation

To navigate through the dependencies simply click on the desired element in the control or double click on the element in the diagram.

You may also select the element in the diagram with a single click and choose between simply opening the referenced diagram or opening it in a new tab in the toolbar that appears.



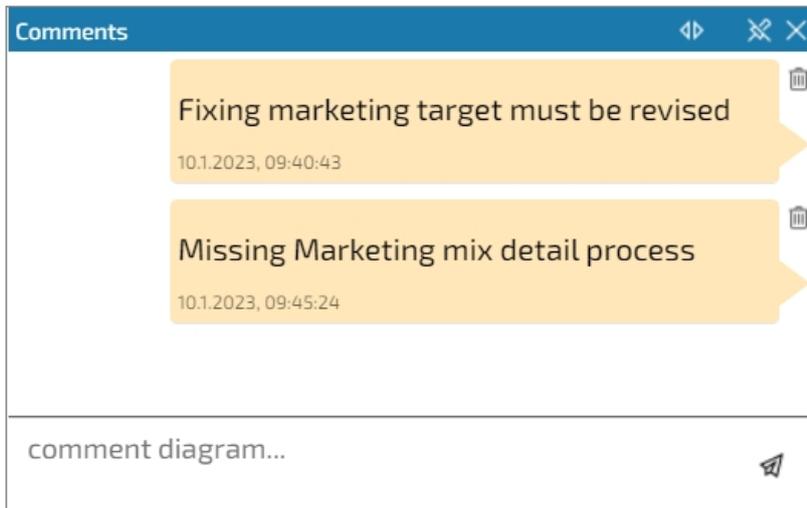
In the breadcrumb bar at the bottom left of the diagram window you can see the folder path. If you want to navigate to one of the elements, you can simply click on it in the bar.

If [views](#) are maintained, they are also displayed in this bar and can be selected in the same way.

## Comments

The Comments panel allows the reader of the process documentation to leave comments on diagrams and activities.

To view comments, click on the "Comments" button in the Diagram menu and the Comments panel will appear. To hide the comments, click on the "x" or unpin the panel.



### Add comment

To add a comment, just write your comment into the text area at the bottom of the Comments panel and click on the small paper plane to the right. The comment then appears in the area above. To leave comments not on the diagram but on individual elements, simply select the desired element in the diagram.

### Delete comment

To delete a comment, click on the trash bin next to the comment.



Important notice:

Note that comments are not deleted, they are terminated. This means that they will be visible if the current [history](#) timestamp is set accordingly.

### Diagram

The diagram represents a graphical description of the process model.



### Zoom, Pan, Fit

You can zoom in and out on the displayed process diagram by using your mouse wheel or zoom to fit the process in the diagram window by using the button  in the grey bar as shown above. To pan your diagram, hold down the right mouse button and drag to the desired position.

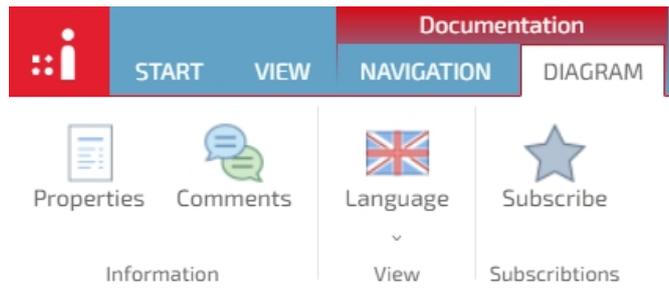
## Navigation

To navigate to process models, which are linked to the shapes, double click on the shape. You may also select the element in the diagram with a single click and use the toolbar that appears. Find more information about navigation inside the process documentation on the page [Dependencies](#).

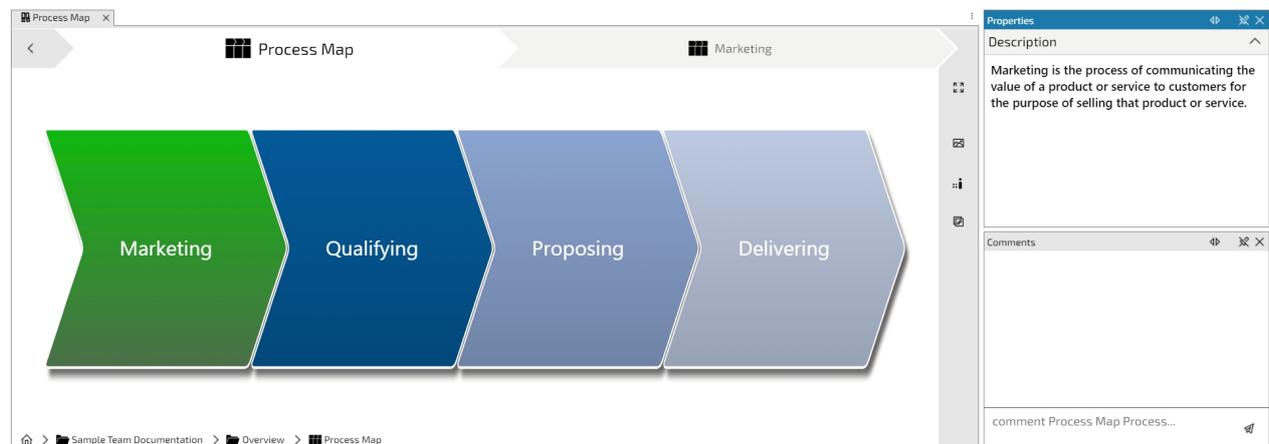
## Diagram / Element details

Diagram or element details are listed in the Properties panel, which by default is to the right of the diagram window.

If the panel is closed, you can open it by clicking on the "Properties" button in the Diagram menu.



The details listed include the description, [responsibilities](#), [extended attributes](#) and tags of the diagram or selected element, as far as they have been maintained. If you select an element, its color changes to green. This way you can differentiate whether you are looking at the details of the diagram or an element and which one it is.



## Views

In the breadcrumb bar at the bottom left of the diagram window you can see the folder path of your open diagram.

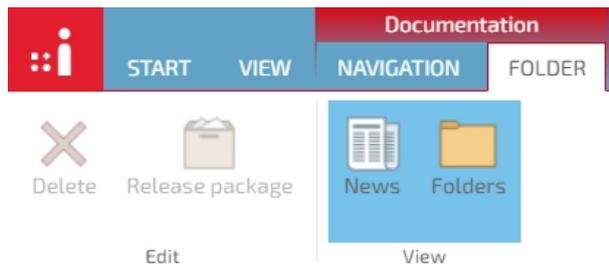
If [views](#) are maintained, the view options are displayed at the end of this bar / folder path. You can switch between the views by clicking on the name of the desired view.



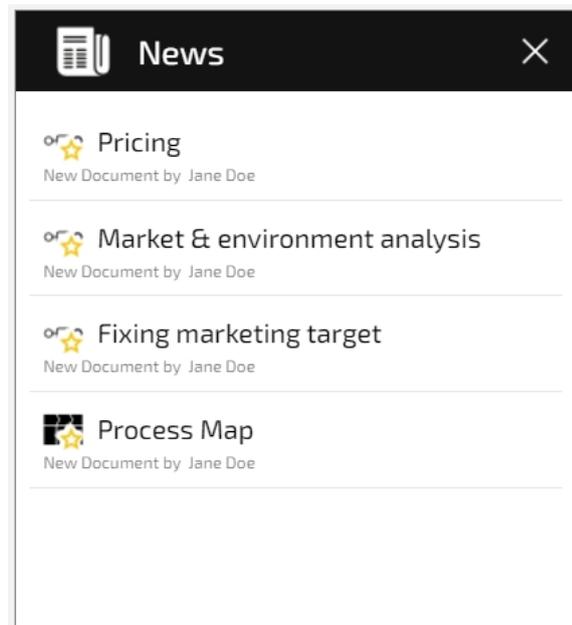
## News

You are able to view the documentation news either on the overview page or while navigating through folders.

To activate the news pane, select the "News" button in the Folder menu. If you want to close the news pane, either click on the "News" button again or click on the "x" in the news pane.



The news pane is located on the right side of the window.



The news pane will show all

- diagrams which have been added to the process documentation
- subscribed diagrams that have been changed
- comments on subscribed diagrams that have been created / changed

## Subscription

Subscribe process models that are of interest to you so that you get informed about changes to these process models.

Every portal user may subscribe individual process models.

Once subscribed the user receives notes about changes in the [news section](#).

## Subscribe

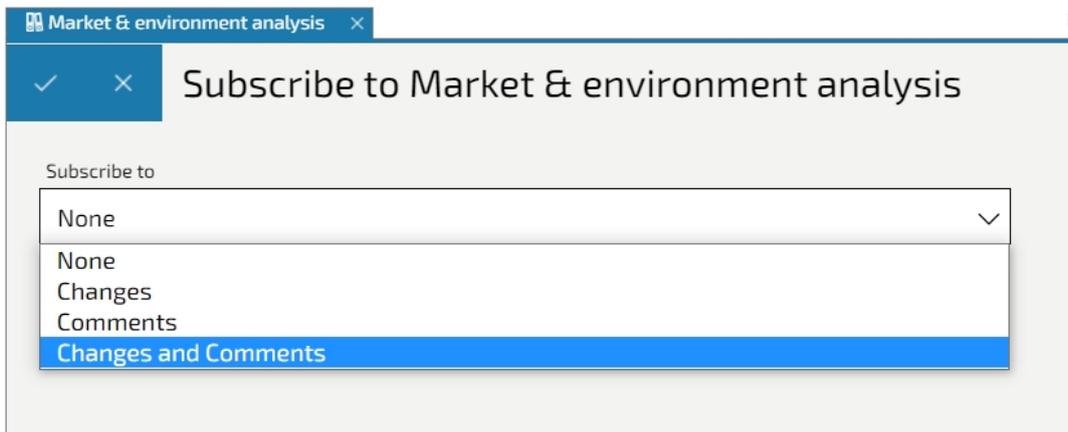
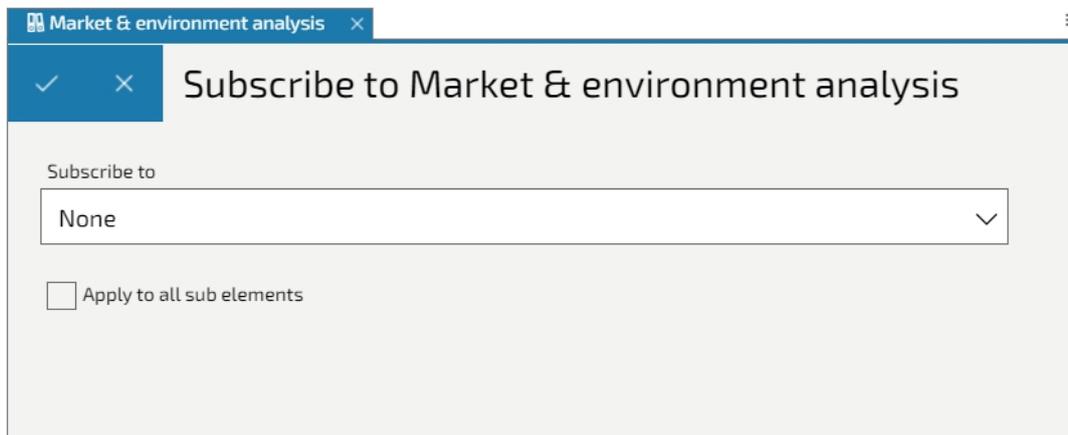
To subscribe a process model locate the desired process within the process documentation. Click on the "Subscribe" button in the Diagram menu to subscribe the process model.



In the upcoming dialog, choose the level of details of your subscription:

- Changes  
Subscribe to process model changes
- Comments  
Subscribe to Comments that have been created or changed

Select the "Apply to all sub elements" button if you wish to subscribe to sub processes as well.



## Unsubscribe

To unsubscribe a previously subscribed process model, follow the same steps as subscribing to a process, but select "None".

## Output

Published diagrams and their views can be exported to provide process related information outside of the portal.

To export a diagram choose an output option from the grey bar to the right of the diagram.



Diagrams are exported as they are displayed so the currently chosen language and the currently selected view will be used in the export.

You have the following options:

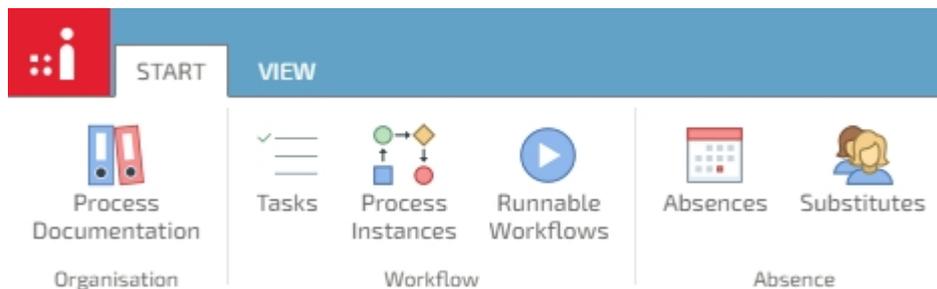
- **Open as Image**  
This option opens the diagram as an image in a new browser tab. You can then save it etc.
- **Open in IYOPRO App**  
This option opens the diagram in the IYOPRO App
- **Copy Link to Clipboard**  
With this option you can copy the link to the diagram and e.g. distribute it

## Workflow

The Workflow part of the IYOPRO Web Start menu provides access to the workflows of your organization.

You have the following choices:

- [Tasks](#)  
This module provides information about open and / or finished tasks of the current user
- [Instances](#)  
This module provides information about process instances, the current user is involved in.
- [Runnable](#)  
This module provides all runnable workflows which may be utilized by the current user.



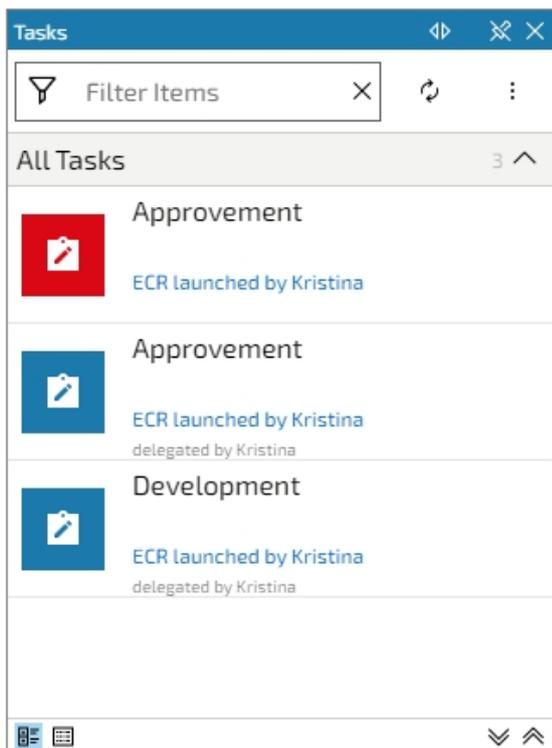
In addition, you have the option of defining absences and getting an overview of the currently defined absence times of other users as well as configuring who should replace you in your absence:

- [Set substitutes](#)
- [Set user absence](#)

## Tasks

The Tasks list provides access to open and / or finished tasks.

If the Tasks list is activated, all tasks of the user are shown. The small number in the collapsible header indicates how many tasks there are.

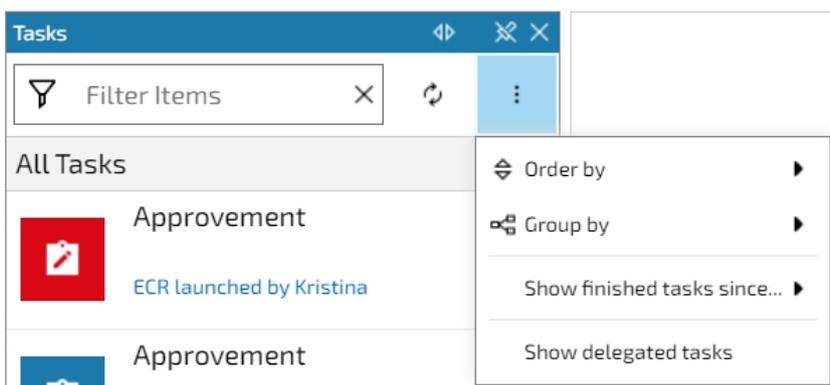


## Task Availability

A various number of user accounts can participate in a workflow task but only one account at a time can be responsible. Therefore workflow tasks are displayed differently depending on their availability for the currently signed on user account.

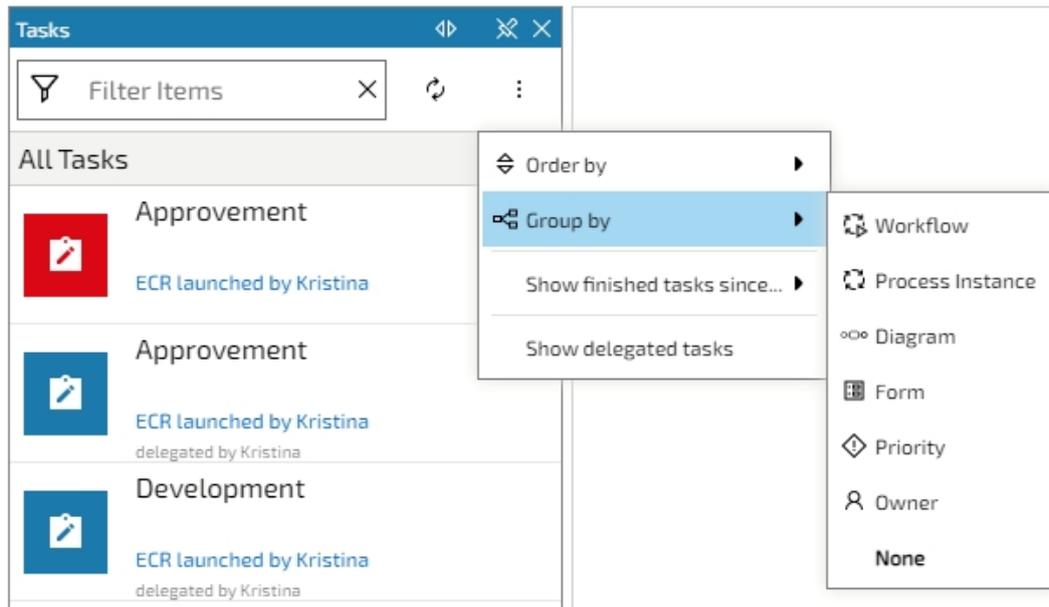
You can find an overview [here](#).

## Task list menu



## Order and group by

You can modify the view of your task list by changing the order of the tasks or grouping the tasks. This makes it easy to work on same incidents, for example.



### Show / Hide finished tasks

The "Show finished tasks since" selection allows several useful and predefined options, when to show finished tasks. These are:

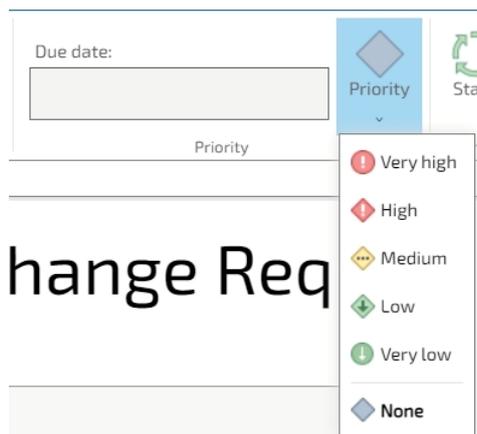
- Never  
A task will be hidden immediately after it's finished
- Today  
Only tasks that have been finished today will be shown
- Yesterday  
Only tasks that have been finished since yesterday will be shown
- Last Week  
Only tasks that have been finished since last week will be shown
- Last Month  
Only tasks that have been finished since last month will be shown
- Last Year  
Only tasks that have been finished since last year will be shown
- Always  
Finished tasks will always be shown
- Date selection  
Only tasks that have been finished since a given date will be shown

### Delegate tasks

Click on the button "Show delegated tasks" to view the tasks that you have delegated. Find more information about this topic in the chapter "[Delegation](#)".

### Set task priorities

Open a task and click on the button "Priority" in the task menu to define the urgency of that task.



The priority is then displayed in the Tasks list. If you have the list view active, the [task icon](#) will be colored according to the priority. Tasks can be sorted by their priority.

## Open a task outside YOPRO

You may open a task outside YOPRO. You have the following possibilities:

- Open the task in the web portal and click on the button "Undock"
- Open the task in the app portal
- Use the URL `<YOPRO Client>/Form.aspx?taskid=<task guid>`  
 Example: <https://www.iyopro.com/iyopro/Form.aspx?taskid=93908093-c8d1-465c-b260-cdaa425a48bc>  
 (see also topic [Start Workflow outside YOPRO](#))

## Substitutes

### Introduction

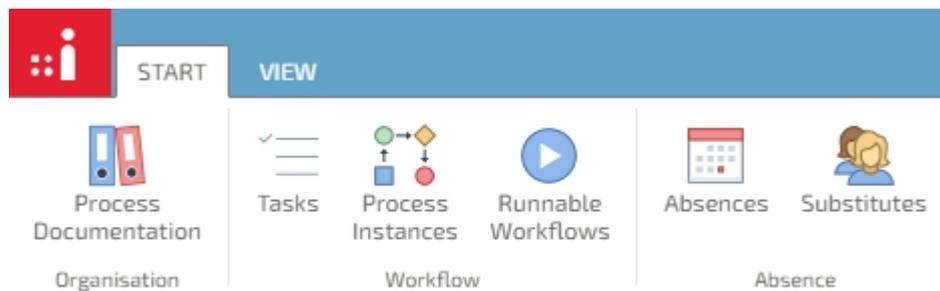
Some workflow tasks have to be finished in a certain time. If the actual responsible user is absent the task has to be done by another user to ensure the prompt handling. Therefore users can define substitutes in YOPRO.

### Conceptual Overview

Users can individually define substitutes for their workflow tasks. Those substitutes will be able to complete tasks during the absence of the actual responsible person.

- [Set substitutes](#)

Define a standard substitute for all the tasks of a user in a certain team or a specific task. You can either configure team substitutes by clicking on the "Substitutes" button in the Start menu



or configure team and task substitutes by clicking on the "Configure Substitutes" button in the Workflow Task menu that appears when you have a task open.



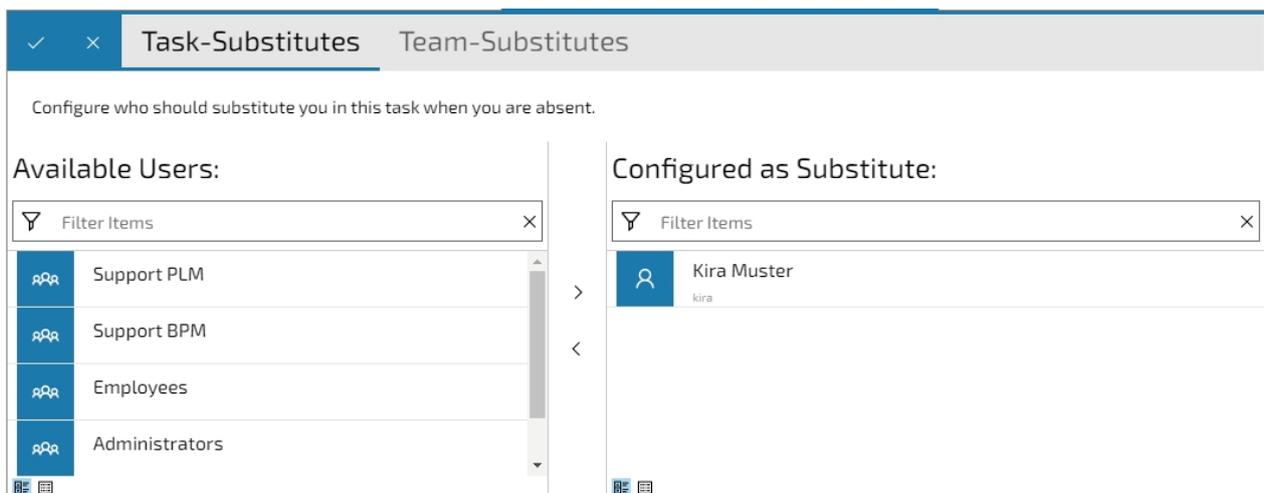
- [Set user absent](#)

Click on the "Absences" button to set a user absent for a defined period of time

## Set substitutes

### Define task substitutes

Workflow tasks can have specific substitutes. Those task substitutes can be determined in the related task forms. Press the button 'Configure Substitutes' to open the configuration.



You have the following usage options:

- Filter  
Enter the name of a user or group to filter the list of available substitutes
- Move to the right  
Either press the arrow to the right or double click on the selected entry to switch the status of selected users and groups from available to configured
- Red arrow  
Either press the arrow to the left or double click on the selected entry to remove the selected users and groups from the configured substitutes back to available substitutes

### Define standard substitutes

Standard or default substitutes can be configured for every team. These specified substitutes will be responsible for every task while the actual user is absent. These standard substitutes will only be informed about tasks that do not have specific task substitutes.

Standard substitutes can either be configured for oneself or for someone else (IYOPRO App).

If you want to configure the substitutes for someone else, navigate to the [User node](#) in IYOPRO App. There you can select a user and configure his or her substitutes. Of course, you can also select your own user.

Alternatively, to configure standard substitutes for yourself, you can either click on the "Substitutes" button in the Start menu or open a workflow task and click on the "Configure Substitutes" button in the Task menu. The mechanism to set standard substitutes is equal to task substitutes which is described above.

## Set user absent

Click on "Absences" in the Start menu to open the absence settings.

The screenshot shows the 'Absences' settings interface. At the top, there is a navigation bar with 'Welcome' and 'Absences' tabs. Below this is a calendar view for December 2022 and January 2023. The user 'Kira Muster' is selected. The status is set to 'Gone' for the selected date (Wed 21.12). The reason field is empty. An 'Add' button is visible at the bottom right.

Press the "+" to add an entry to the absence table.

The following absence settings can be made:

- User  
Choose the user who's absence status is supposed to be changed
- Absence duration  
Choose the date and duration of the absence status by selecting the date(s) in the calendar. If the selected user already has an entry in the absence table, you can also select the date(s) directly in the entry.
- Status  
Decide whether the selected user is absent or present during the selected time
- Reason  
Use this to give a reason for the absence

Press "Add" to save the absence entry. It will be displayed in the absence table.

You can change the displayed dates by pressing on the arrows on the right and left side or by dragging the date header.

## Delegation

In some cases the actual person responsible will not be able to handle a workflow task because of certain circumstances. This might happen because of limited time, missing knowledge or because the task was simply addressed to the wrong person. Therefore IYOPRO provides the delegation functionality.

Delegating a task means to hand over the responsibility for one or more workflow tasks to another person.

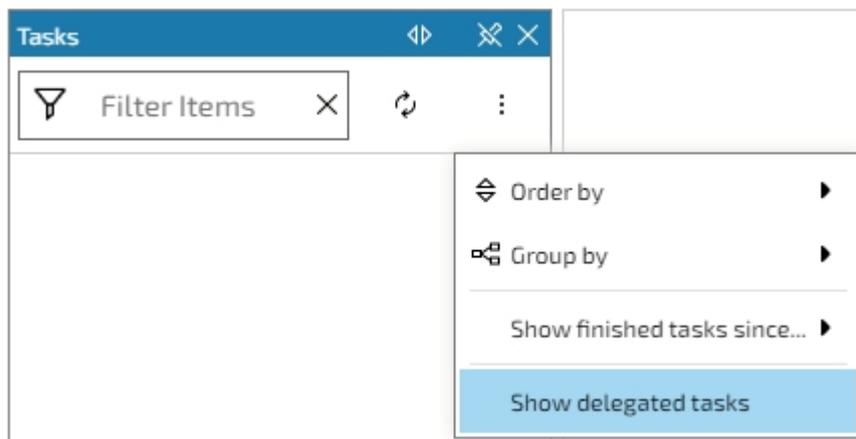
- Delegate task  
Use this to hand over the responsibility for the selected task to another user
- Revert task delegation  
Use this to decline the responsibility of a task that has been delegated to you and give it back to the sender

- Show delegated tasks  
This provides an overview of all delegated tasks of the user

The delegation options can be found in the workflow task menu that appears when you have a task open.



To view the delegated tasks, expand the overflow menu of the task list and select the corresponding option.



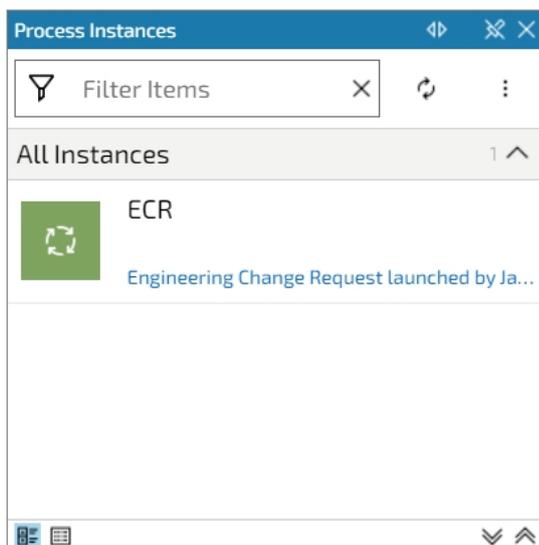
## Instances

The Process Instances list provides access to running or finished process instances.

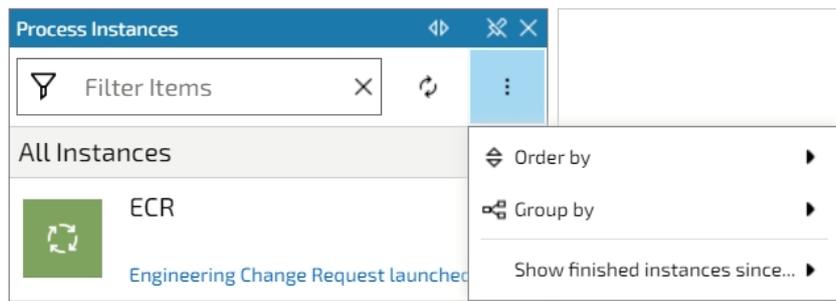
If the Process Instances list is activated, all instances in which the user is involved are displayed. So the user has started the instances himself or he should manage the instances.

The process instances can have different states. You can find an overview [here](#).

The small number in the collapsible header indicates how many process instances there are.

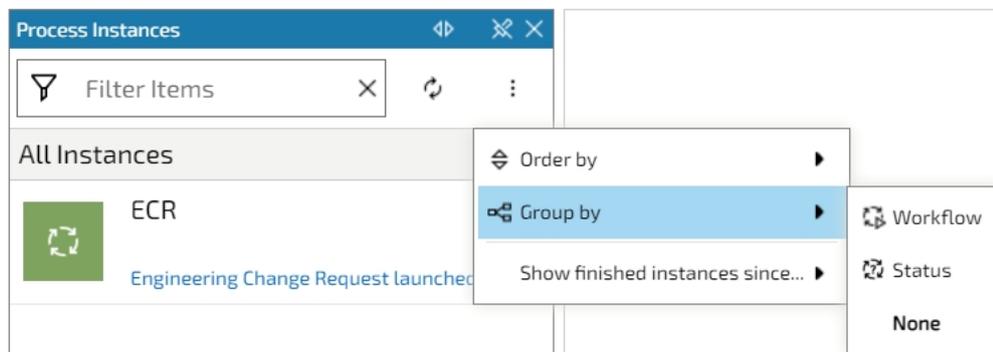


## Process instance list menu



### Order and group by

You can modify the view of your instance list by changing the order of the instances or grouping the instances. This makes it easy to work on same incidents, for example.



### Show / Hide finished instances

The "Show finished instances since" selection allows several useful and predefined options, when to show finished instances. These are:

- Never  
An instance will be hidden immediately after it's finished
- Today  
Only instances that have been finished today will be shown
- Yesterday  
Only instances that have been finished since yesterday will be shown
- Last Week  
Only instances that have been finished since last week will be shown
- Last Month  
Only instances that have been finished since last month will be shown
- Last Year  
Only instances that have been finished since last year will be shown
- Always  
Finished instances will always be shown
- Date selection  
Only instances that have been finished since a given date will be shown

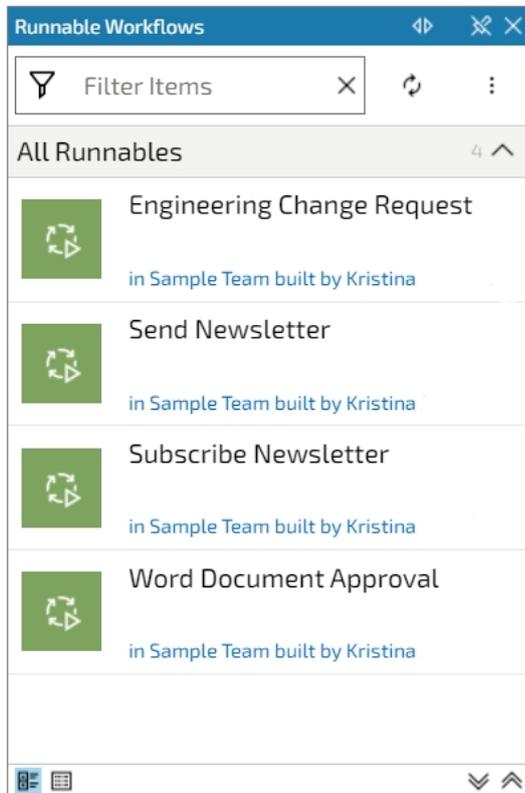
## Monitor process instances

To monitor process instances follow [this guide](#).

## Runnable

The Runnable Workflows list provides access to executable workflows.

If the Runnable Workflows list is activated, a list of all workflows that have an [active version](#) and that the user could start is displayed. The small number in the collapsible header indicates how many runnable workflows there are.



Double click on the appropriate workflow to start.

Then either a [start form](#) will appear or the IYOPRO Start Workflow Dialog, depending on the settings in the workflow definition.

If a start form has been declared, a process instance will be created when the form is submitted.

## Start Workflow Dialog

The Start Workflow Dialog is used for all types of workflows, which must be manually started.

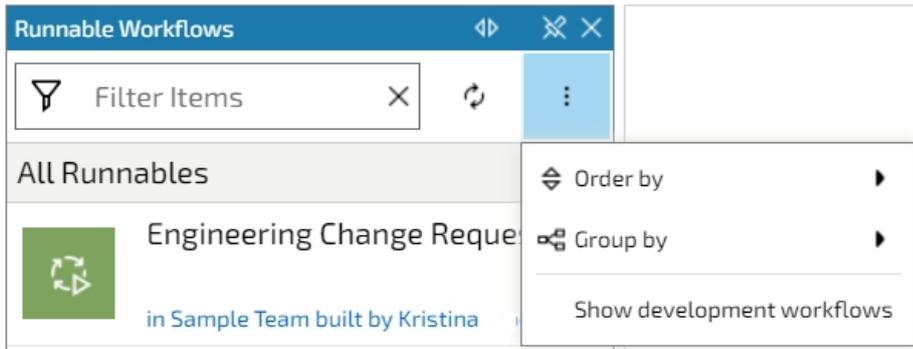
It shows the name of the workflow that you are about to start and a button to run the workflow.

If you want to get more information about that workflow, you can open the help by clicking on the "Help"

button in the menu (). It shows some organizational information as well as a preview image and a description.

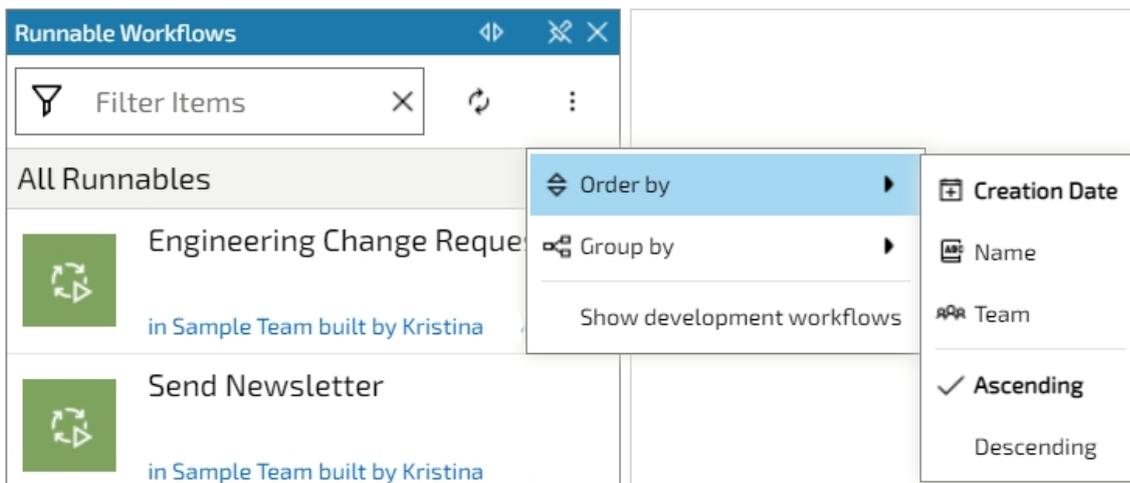
To start such a workflow hit the "Run Workflow" button.

## Runnable workflow list menu



### Order and group by

You can modify the view of your runnable workflow list by changing the order of the workflows or grouping the workflows. This facilitates work on the same incidents.



### Show development workflows

If you have the necessary authorization, you can display (and start) workflows that are still in development. Whether or not you see development workflows in the list depends on what is set in the [Workflow Build Options](#).

## Overview of workflow list icons and their meaning

Here you can find an overview of the different icons that are displayed in the workflow lists (for IYOPRO Web mainly in list view) and their meaning.

### Tasks

App	Web	Meaning
		I completed the task
		The task was otherwise completed
		Open task, I should / can complete this task

		Open task, I should / can complete this task on behalf of an absent person
		Open task, This task was accepted by someone else, I can't edit it (anymore)

### Task priorities

App	Web	Meaning
		No priority
		Very low priority
		Low priority
		Medium priority
		High priority
		Very high priority

In the list view of IYOPRO Web, the background colors of the task icons are displayed according to the colors of the priorities:



- Very low / low priority,



- Medium priority,



- Very high / high priority

### Process Instances

App	Web	Meaning
		Finished
		In execution
		An error has occurred
		Stopped and delayed due to execution conditions (e.g. running long-term)

### Runnable Workflows

App	Web	Meaning
-----	-----	---------

		Active version of a workflow
		Development version (not active version) of a workflow

# Business Process Modeling

---

IYOPRO is one of the fastest and easiest BPMN 2.0 process modellers available.

Note: In this chapter we refer to [The Properties Area](#). In the "Property Set" you may select:

- Modeling
- Simulation

We refer to the setting "Modeling" here.

## Content:

[Single file diagramming](#) - Working with a single diagram

[BPI projects](#) - Arrange one or more diagrams and documents in a BPI project

[Diagram types](#) - Get an overview of the different diagram types

[Validation](#) - Get an impression how IYOPRO will assist you while modeling diagrams

[Properties](#) - Understand the basic concept of properties and property editors

## Single file diagramming

In this chapter the single file diagramming is described. Single file diagramming is usually the quickest way to generate a process documentation.

Content:

[Create a new diagram](#)

[Edit an existing diagram](#)

[Save a diagram](#)

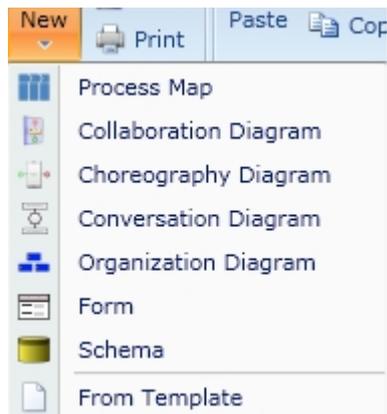
### Create a new diagram

To create a new diagram click on the "New" button in the Home ribbon tab.



In general this will create a new collaboration diagram and you can start diagramming immediately.

If you wish to create another diagram type you may click on the small arrow below the new button and choose one from the list of available diagram types:

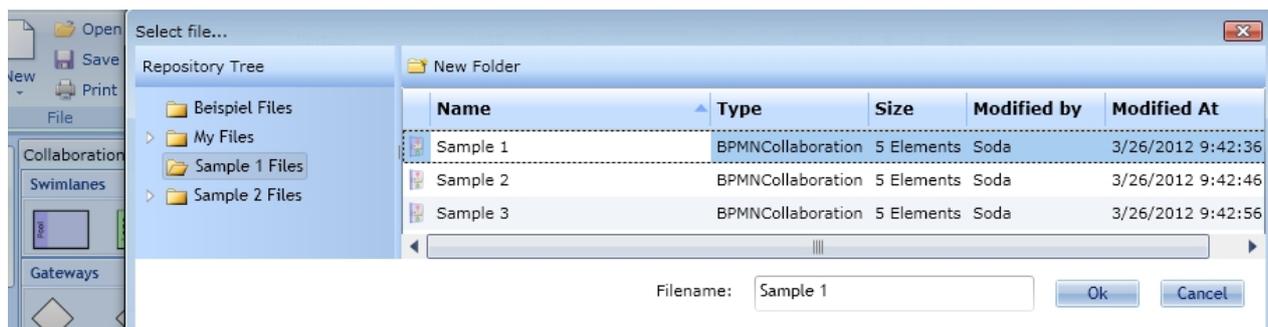


Alternatively you may select "New" from the [Application menu](#) and chose one of the diagram types shown.

To learn more about the different diagram types refer to [Diagram types](#)

## Edit an existing diagram

To open an existing diagram from within the repository click the "Open" button from the main ribbon tab and choose the desired diagram:



The file selection dialog will appear. This allows you to browse through the repository file system and locate the required dataset.

You may either double click this item or select it and hit the OK button to open it.

This operation is also available via the [Application Menu](#).

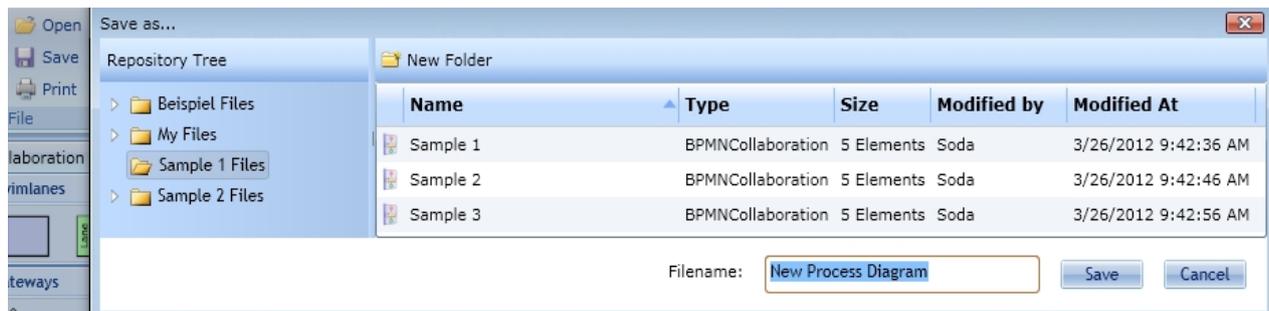
Alternatively you may open the file from within the [Repository Browser](#).

## Save a diagram

To save a diagram simply click on the save button either in the [home tab](#) of the ribbon, the [quick access toolbar](#) or the [application menu](#).

If the file to save has already been saved it will be overwritten. If the file does not exist a "Save as" operation will be performed.

To save a file using a new name (create a copy) use the "Save as" button out of the [application menu](#).



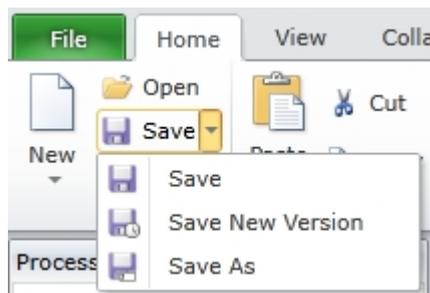
Browse to the location where you wish to save your file. Enter the file name and hit "Save"

## Version of a diagram

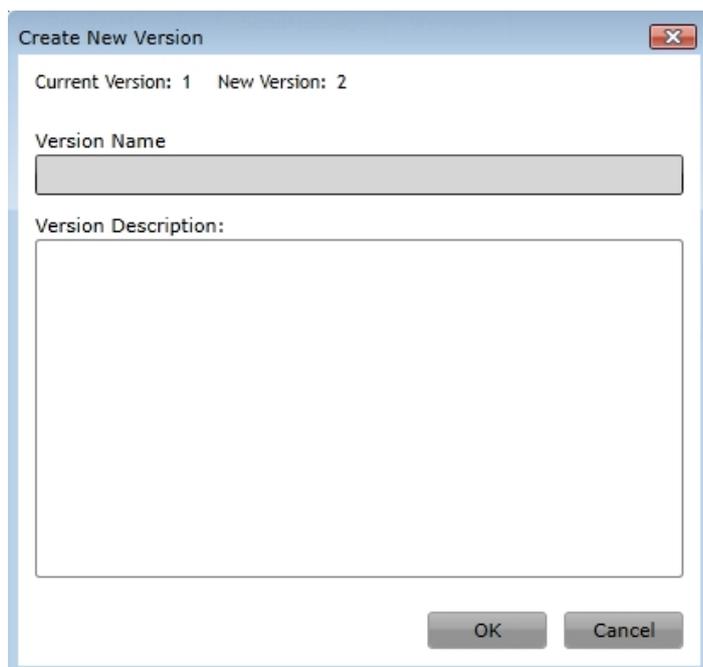
If you don't want to overwrite your original diagram after making changes, you may work with versions of that diagram.

## Save New Version

To save a diagram as a version click on "Save New Version" in the "Save As" menu. You can find it either in the [quick access toolbar](#) or the [application menu](#).

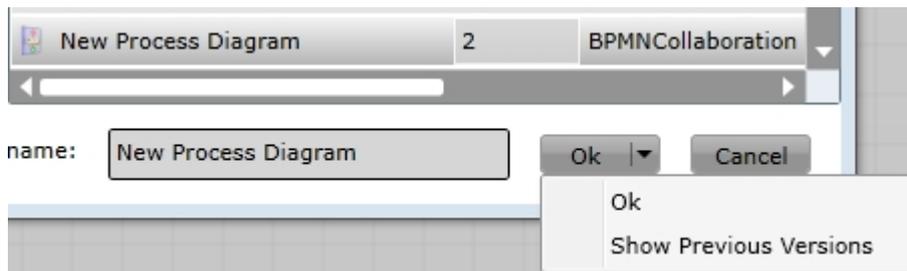


The following dialog will be displayed. You can now enter a name for the new version and a description.



## Open an older version

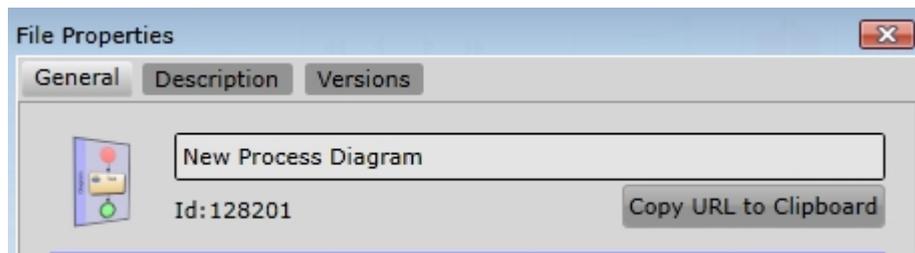
If you want to load a previous version of a diagram, click on "Open" in the [quick access toolbar](#) or the [application menu](#). In the file selection dialog you may now navigate to the diagram. After selecting the diagram, click on the arrow of the splitted OK-Button and then on "Show Previous Versions".



You may then choose from a list of available file versions of the selected object. In addition you'll find a Properties entry in the [application menu](#). The properties dialog contains a "Versions" tab which provides the required information.

### Load a diagram version via URL

Generally, you may load a diagram via URL. To get the URL, open its file properties, e.g. as described above or by right-clicking on the diagram in the [process explorer](#).



Copy the URL to the clipboard. It should be something like this:

```
https://www.iyopro.de/iyopro?load=124218
```

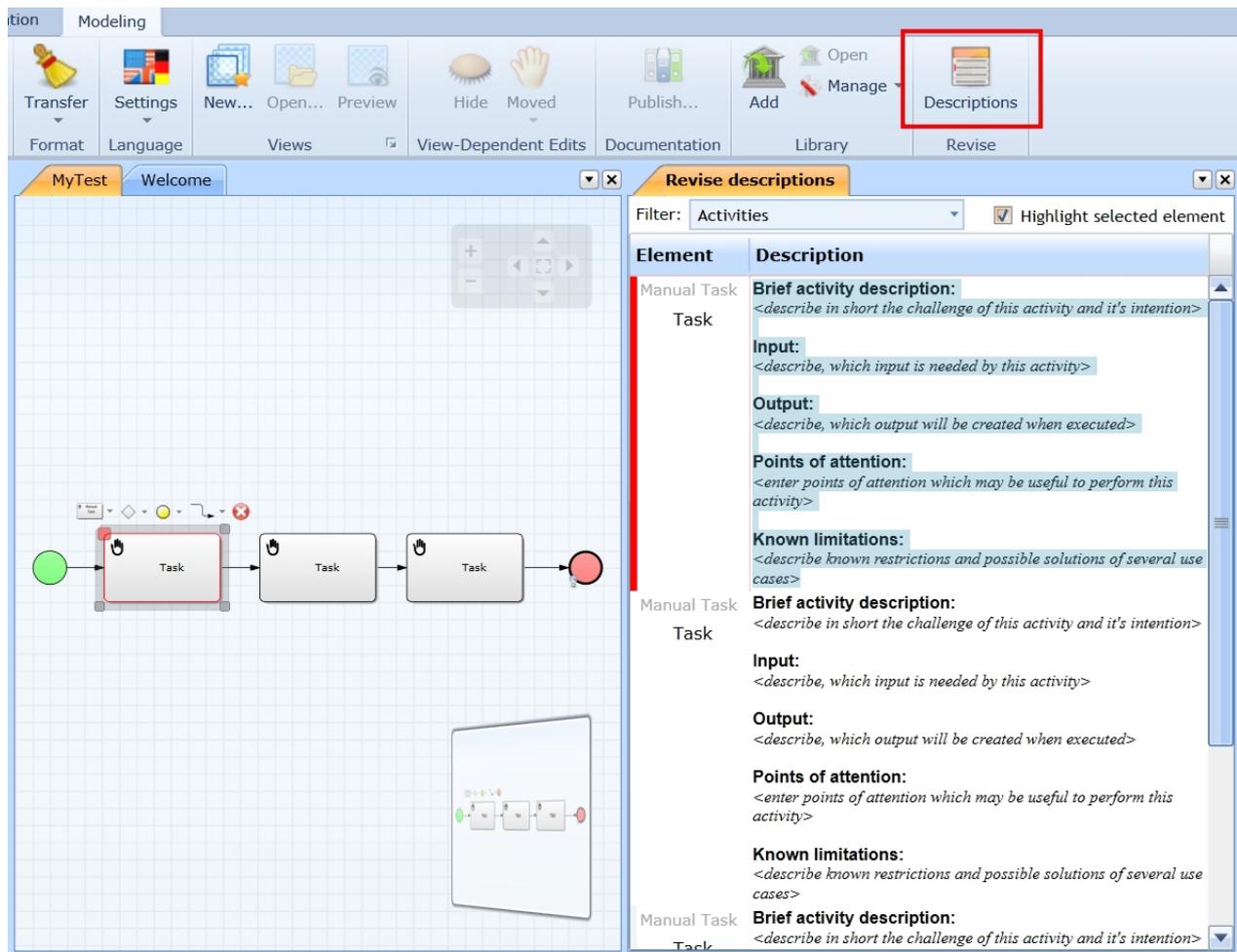
The URL will load the newest version of a diagram. If you want to load a specific version, add the URL parameter "version" or short "v" to the URL. In this parameter, the version number is passed. The result should look like this:

```
https://www.iyopro.de/iyopro?load=124218&version=1
```

### Tabular description editor

Editing descriptions of activities can become a time consuming job if the number of diagram elements grow. To be able to edit diagram descriptions in a more clear way IYOPRO provides the tabular description editor.

To open the description editor use the Descriptions-Button in the Revise-Group of the Modeling tab in the ribbon.



IYOPRO will place the editor and the diagram side by side. You may either click on activities in the diagram or in the editor to activate the description to edit.

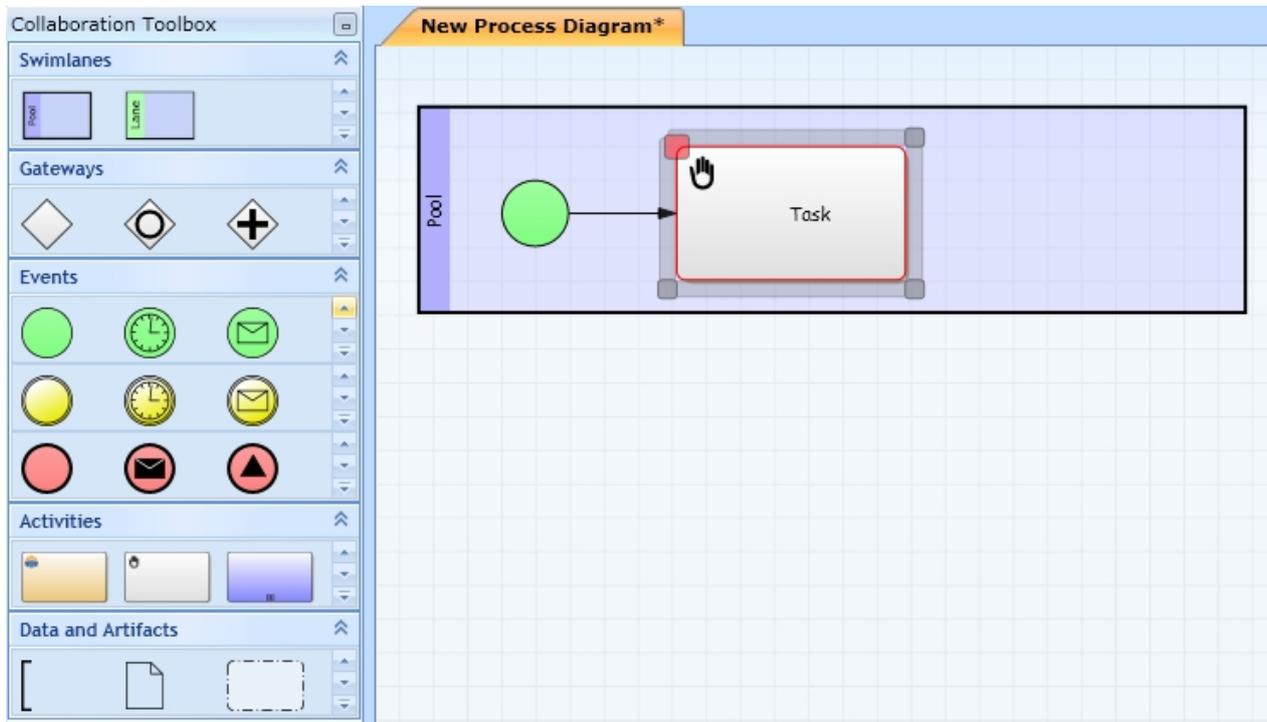
Note, that you not only can edit the description but also the name of the shape within the tabular editor.

## Modeling tips

If a new diagram is shown in a new modeling window the corresponding toolbox will appear in the toolbox area.

The toolbox provides all available stencils to perform diagramming of the chosen diagram type.

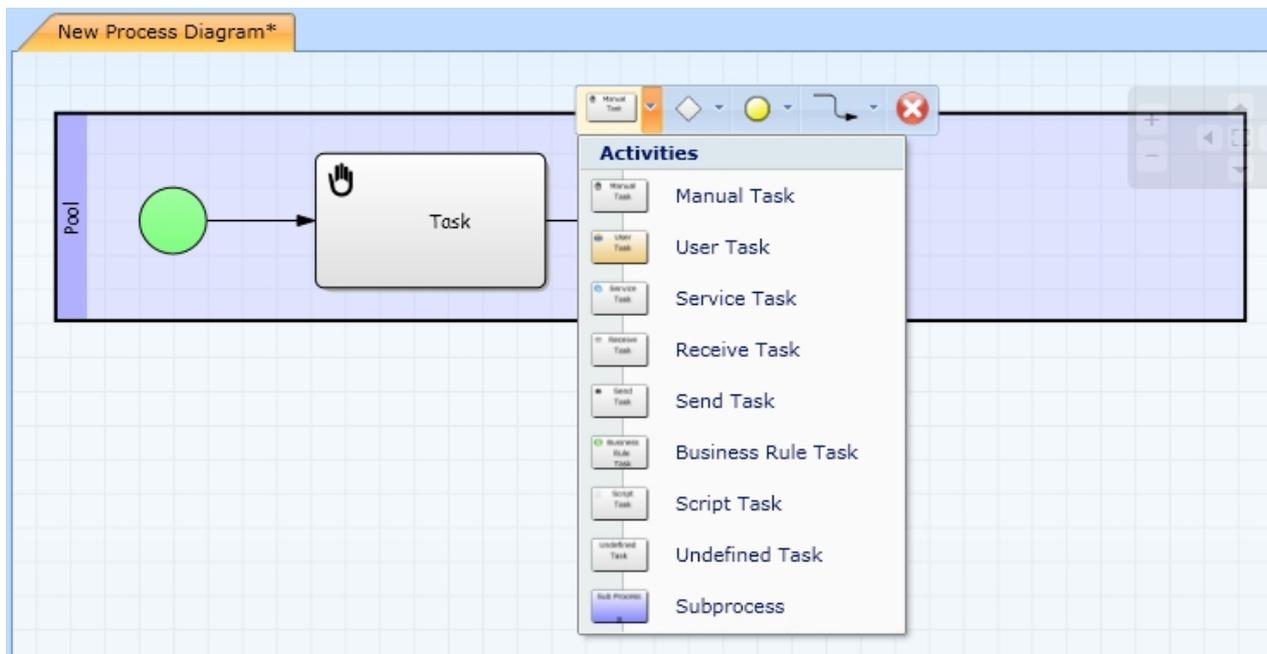
You may now drag symbols from the toolbox onto the work area. If a wrong symbol has been dragged it can either be dropped back on the toolbox or be deleted by pressing Esc on the keyboard.



**Note:**

If you drag a new item from the toolbox over the diagram and the diagram contains a selected item IYOPRO will automatically create a sequence flow (or whatever makes sense) between the selected item and the new one.

If you select an item in the diagram or move the mouse over an element a context sensitive tool-bar will appear. This tool-bar provides commonly used elements in context of the reference object, from which you may drag additional items onto the diagram.



**BPI projects**

BPI is a shortcut of Business Process Improvement. The methodology has first been documented by H. James Harrington in 1991. The systematic approach enables optimizing processes in order to achieve more efficiency and effectiveness

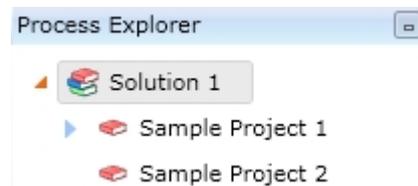
The objective of improving a business process implicates that this intention becomes a project:

- Unique goal
- Time, cost and resource constraints
- Separation from other development plans (business as usual, other projects)
- Specific organizational structure

The "BPI project" feature in YOPRO facilitates process management by offering a structured procedure. Based on BPI Projects you may later create workflows or perform deep BPS simulation & analysis. Like the repository tree the "BPI Project" follows a hierarchy. The hierarchy is displayed in [the process explorer](#).

The core elements of the "BPI project" feature are:

- Solution - the root which groups projects together
- Project - a single BPI project.

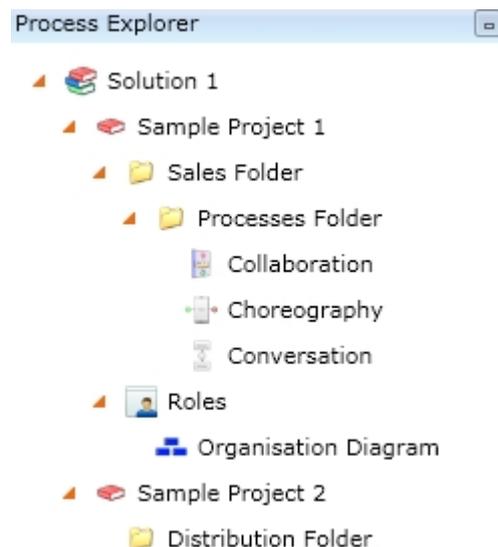


## Solution

The "Solution" is the "bracket", which groups various "Projects" together. If you create a new project, you may choose to create either a new solution or attach the project to an existing solution. ([Create a new BPI project](#)).

## Project

A Project consists out of all documents, which describe the process model, i.e. diagrams, documents, folders etc.

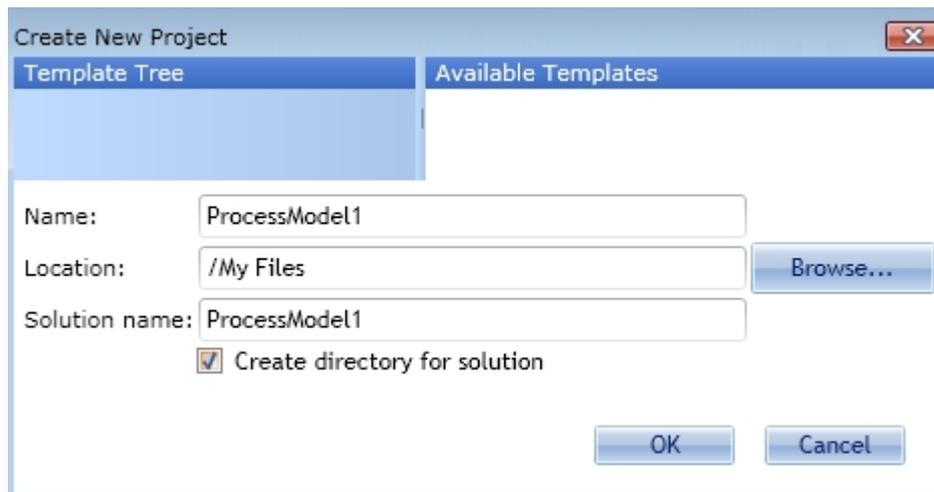


This chapter provides information on the following operations

- [Create a new BPI project](#)
- [Add a new diagram](#)
- [Add an existing diagram](#)
- [Add a folder](#)
- [Remove an item](#)
- [Rename an item](#)

## Create a new BPI project

From the [Application menu](#) click "New" -> "BPI Project" :



### Name

Enter the name of the "Project".

### Location

Browse for the destination folder.

### Solution name

If there is a current solution is active, the new project will be attached to that solution. You may choose to create a new solution, where the project will be attached to.

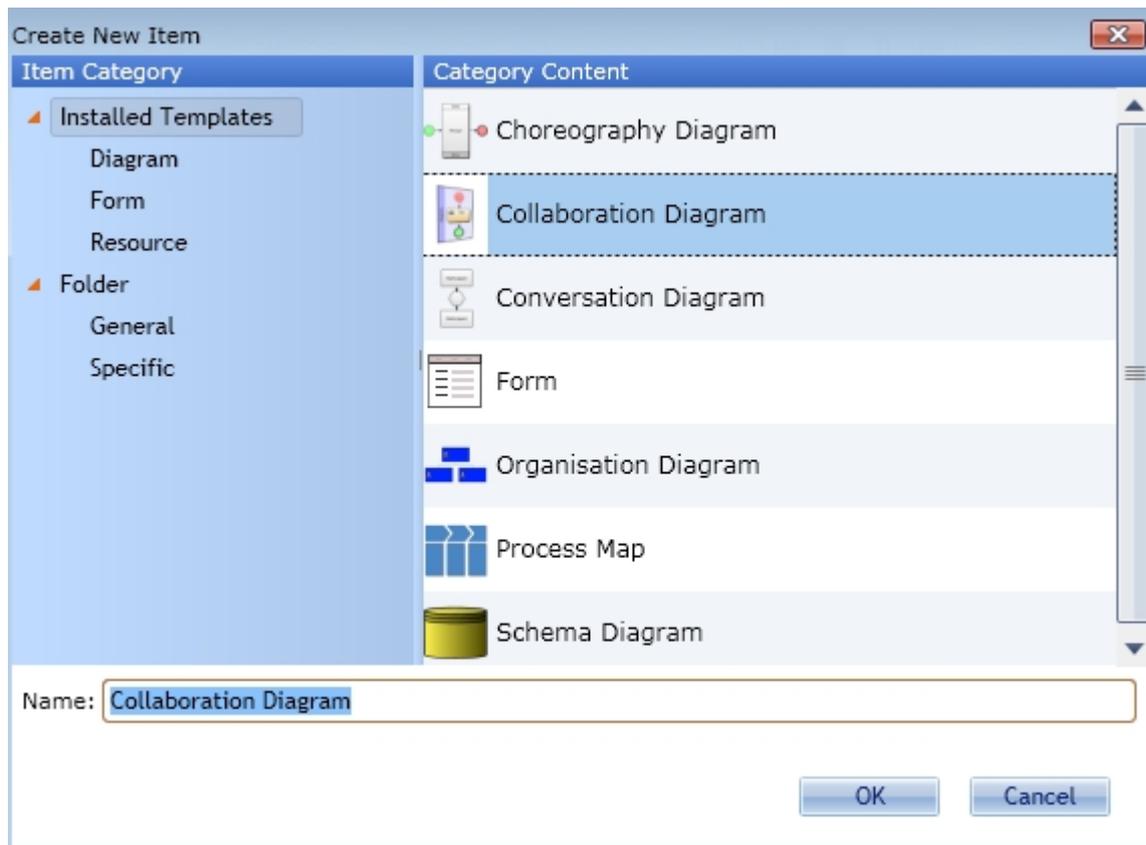
### Create directory for solution

If selected the solution will contain a subdirectory to organize the associated projects.

Click "OK" and the project will be created and shown in the [The Process Explorer](#).

## Add a new diagram

In the [process explorer](#) select a project or project folder. Right click to display its context menu. Click "Add" -> "New Item".

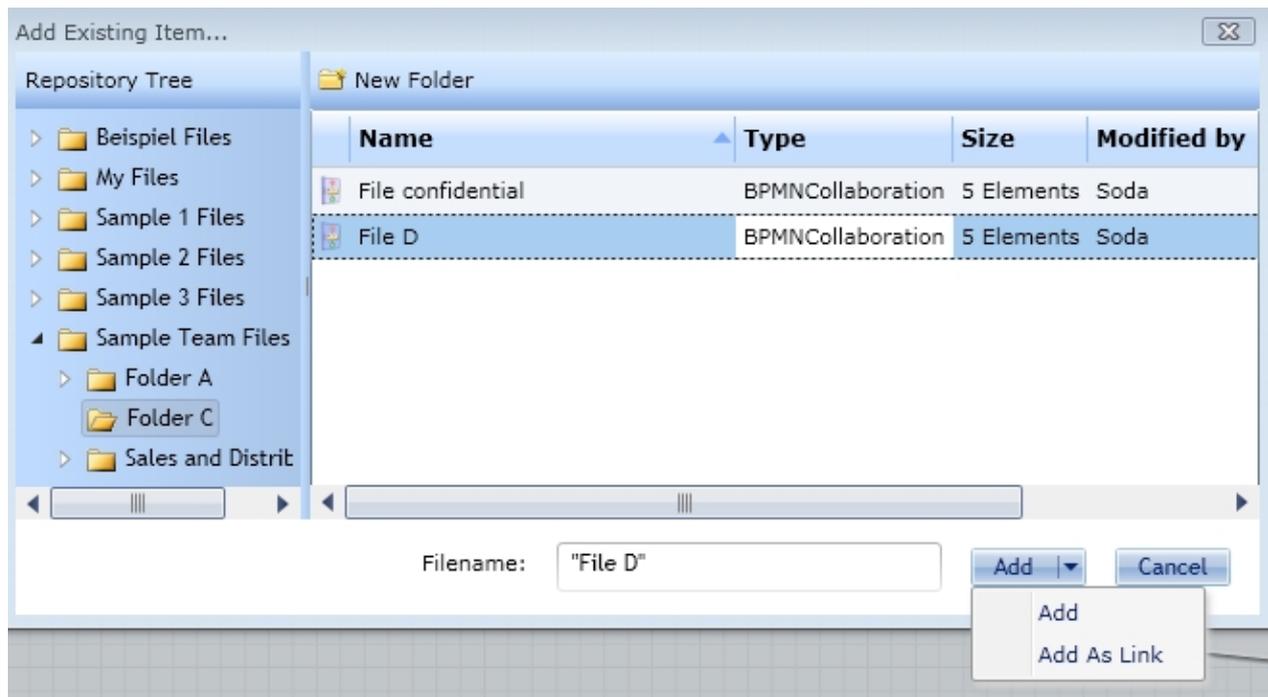


Select the desired diagram type from the templates tree and name the new diagram. Click "OK" to add a new diagram to your project.

### Add an existing diagram

If your repository contains an item which will be required in your project you may choose "Add existing item".

In the [process explorer](#) select a project or project folder. Right click to display its context menu. Click "Add" -> "Existing Item".

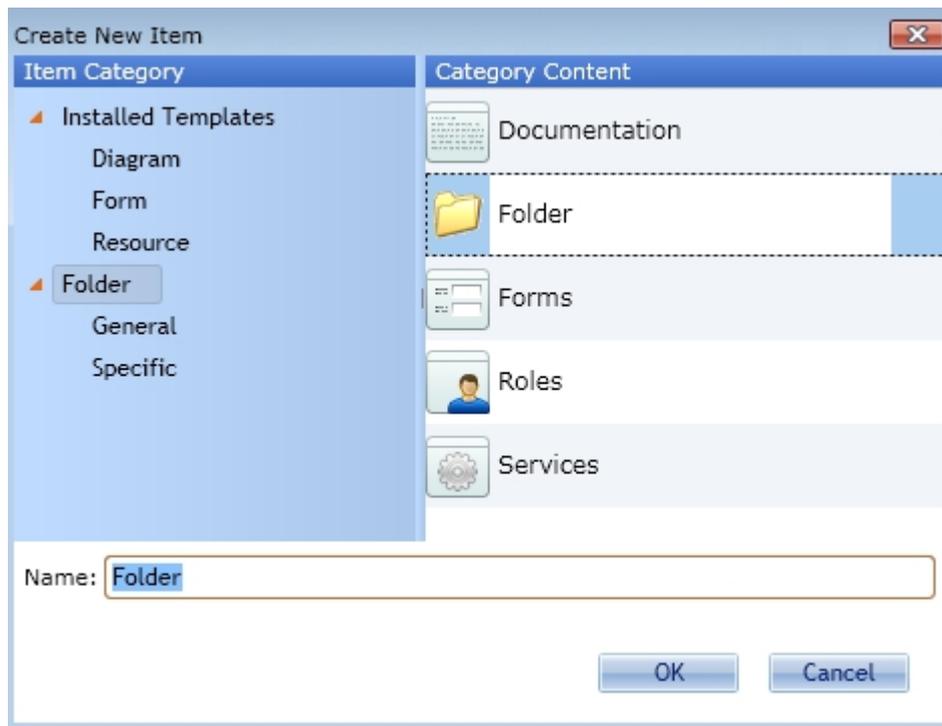


Browse for the file you wish to add and select it. Note the arrow beside the add button. You may perform the following add modes:

1. Add - the existing item will be copied into the project. The source item remains and will not be changed due to editing operations. This is the default.
2. Add As Link - the existing item will be referenced and changed on editing and save operations from within the project.

### Add a folder

In the [process explorer](#) select a project or project folder. Right click to display its context menu. Click "Add" -> "New Folder".



Select the desired folder type and name the new folder. Click "OK" to create the folder.

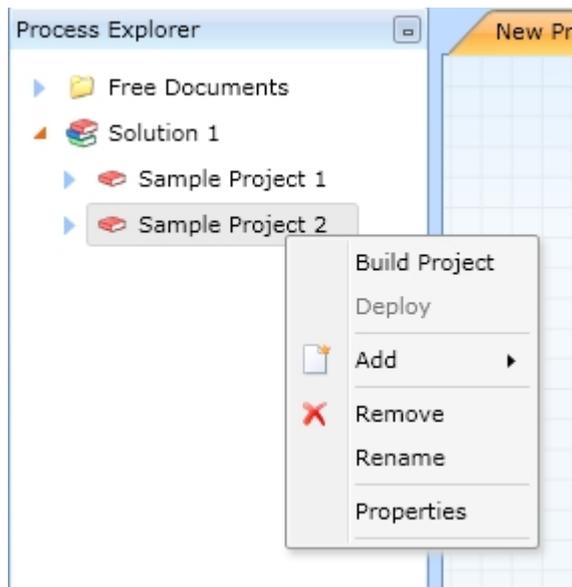
YOPRO provides different folder symbols. These are:

- General Folder
- Documentation
- Forms
- Roles
- Services

This templates we provide in order to make your project even more transparent. There are no restrictions regarding the items, which may be stored in a specific folder type.

## Remove an item

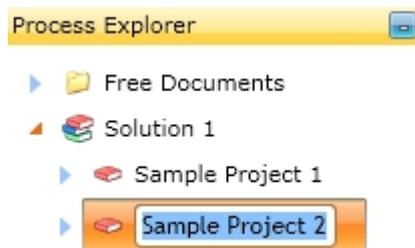
In the [process explorer](#) select an item. Right click to display its context menu. Click "Remove".



Note: Remove an Item removes it from the project structure. It does not delete the item from the file system.

### Rename an item

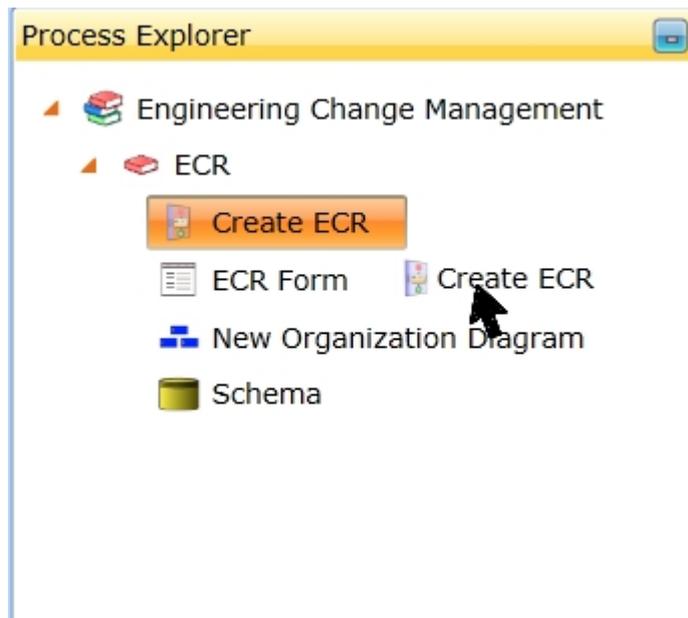
In the [process explorer](#) select an item. Right click to display its context menu. Click "Rename".



Enter the desired name and press return. This operation will rename the associated file in the repository as well.

### Rearrange an item

In the [process explorer](#) select an item and drag it over a folder or a project element to move the item to that location.



## Diagram types

IYOPRO not only provides diagram types according to the BPMN 2.0 standard, but also additional templates to improve the transparency and clarity of your business processes.

### Contents:

[BPMN 2.0 diagrams](#)

[Schema diagrams](#)

[Organisation diagrams](#)

[Process maps](#)

[Forms](#)

### BPMN 2.0 diagrams

## Business Process Model and Notation (BPMN)

"Business Process Model and Notation (BPMN)" is a graphical specification (language), which provides a set of symbols in order to facilitate diagramming business processes. The standard has originally been named "Business Process Modeling Notation". The current version is 2.0. BPMN is maintained by the Object Management Group.

A BPMN Diagram is the depiction of an entire BPMN model or a part of it. To reflect this, the following BPMN 2.0 diagrams are available in IYOPRO:

- [Collaboration diagram](#)
- [Choreography diagram](#)
- [Conversation diagram](#)

In order to construct these diagrams, a set of symbols is used:

### [Shapes](#)

Collaboration diagram

## BPMN definition:

### Collaborations:

A collaboration is the synchronized interaction via communication between two or more processes. The participants of the collaboration are represented by pools (refer to [Pools and Lanes](#)). The processes run within these pools, which are connected by message flows to show the message exchange between the participants.

## Processes

In general a process consists of activities and capital, which are interrelated. A process requires input and has an outcome.

In BPMN a process is graphically represented by a sequence of stencils. These stencils are a limited selection of [Gateways](#), [Events](#), [Activities](#) and Sequence Flows.

The Object Management Group differentiates between:

- Private Business Process
  - Private non-executable (internal) Business Processes
  - Private executable (internal) Business Processes
- Public Processes

In IYOPRO the term "Collaboration diagram" encompasses all standard processes, regardless of whether a messages flow actually is taking place.

In using the overall heading "Collaboration diagram", it should be emphasized that all stencils required for modeling with or without message flow are available.

The diagram type "Collaboration diagram" is shown together with the Collaboration Toolbox.

## Collaboration Toolbox



Simple example of a collaboration diagram: Book ordering

## Choreography diagram

## BPMN definition:

### Choreography diagram:

A Choreography diagram focuses on the interaction between participants. It may consist of [Gateways](#), [Events](#) and [Activities](#). The "Activity" symbol available in this diagram type represents message exchanges between participants

The diagram type "Choreography diagram" is shown together with the Choreography Toolbox.

### Choreography Toolbox



### Conversation diagram

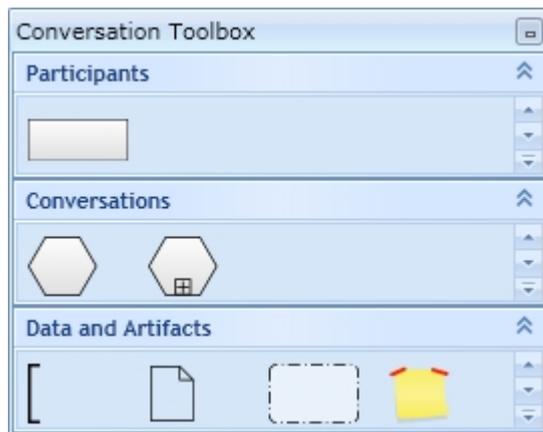
## BPMN definition:

### Conversation diagram:

Conversation diagrams provide an overview of which "conversations" are processed jointly by which participants. A conversation is a logical relation of message exchanges. This logical relation combines different tasks, such as "Ordering" and "Delivering" in the book ordering process.

The diagram type "Conversation diagram" is shown together with the Conversation Toolbox.

### Conversation Toolbox



## Shapes

The five basic categories of elements in BPMN are

1. Swimlanes
  - [Pools and Lanes](#)
2. Flow Objects
  - [Gateways](#)
  - [Events](#)
  - [Activities](#)
1. Artifacts (= [Annotations](#))
2. Connecting Objects
  - [Sequence Flow](#)
  - [Message Flow](#)
  - [Associations](#)
  - [Data Associations](#)
3. Data
  - [Data Objects](#)
  - [Data Inputs](#)
  - [Data Outputs](#)
  - [Data Stores](#)

## Content:

[Pools and Lanes](#)

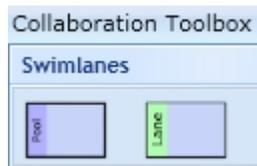
[Gateways](#)

[Events](#)

[Activities](#)

[Annotations](#)

Pools and Lanes

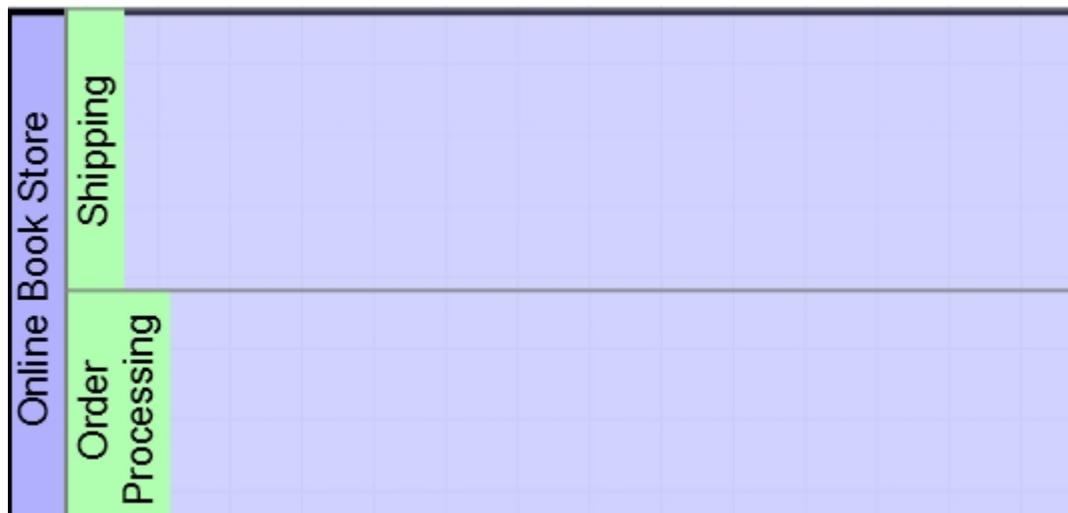


## BPMN definition:

The term "Swim-lane" encompasses "Pools" and "Lanes".

Pools represent the participants in a collaboration. A pool is a kind of container for a process, which is separating it from another pools.

A lane is partitioning a pool into sub-segments, which fill the entire length either vertically or horizontally.



The stencils "Pool" and "Lane" can be dragged as any other shape too from the Collaboration Toolbox onto the working area.

To edit the name of a swim-lane either double click onto the heading, select the swim-lane and press F2 or edit the name in the [Properties Area](#).

The following [Properties](#) may be edited:

Multiple Instances	Determines, if the pool will be repeated
	No Pool is performed once
	Parallel Three vertical lines shown at the bottom-center of the pool indicate a parallel multiple instance. Activities in the pool can follow a any order (can be parallel).
	Sequential Three horizontal lines shown at the bottom-center of the pool indicate a sequential multiple instance. Activities in the pool need to follow a certain order (sequence).
Black Box	Only the name of the pool will be

	displayed in a box
Free Position	The pool is freely movable within the modeling area
Orientation	Horizontal The pool is displayed horizontally Vertical The pool is displayed vertically
Executable	None Not executable. For documentation purposes. Full Fully executable as workflow Simulation Executable for simulation purposes
Expression	Advanced property to create program code in order to enable the execution of the pool as a process
Responsible	Assign participants responsible for this pool.

## Gateways

### BPMN definition:

Gateways are used to split and merge sequence flows in Collaboration- and Choreography diagrams.

Gateway Control Types:

Notation	Element	Description
	Exclusive gateway	Split: Only one of the possible outgoing paths can be taken. Join: Wait until the token arrives  Depiction without or with "X" is possible, but the usage should be consistent.
	Inclusive Gateway	Split: More than one of the possible outgoing paths can be taken. Join: Wait, until all possible incoming tokens arrived.
	Parallel Gateway	Split: All of the possible outgoing paths are taken. Join: Wait until tokens arrived from all incoming path's
	Event based exclusive gateway	Starts a process, if one of the following events occur.
	Event based parallel gateway	Starts a process, if all of the following events occur.
	Event based gateway	The outgoing path is determined by an event / a receive task

	Complex gateway	A set of any rules.
---	-----------------	---------------------

Concept of a token: How do diagrams behave?

The Object Management Group (OMG) has invented the token as a theoretical concept in order to define, how processes do behave when executed.

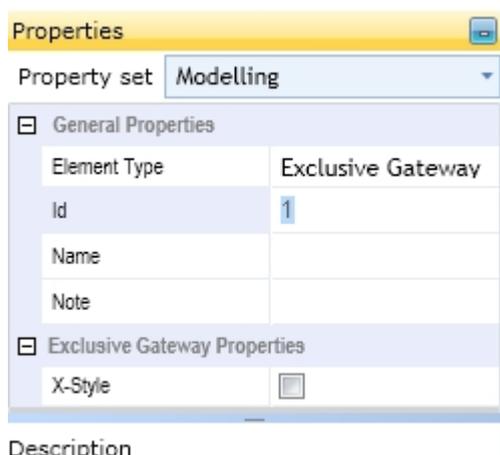
The token passes via the sequence flows through the elements.

As described for the gateway control types above, they are interacting with a token. A token passing through a Parallel Gateway that splits a sequence flow will be duplicated. One token for each sequence flow that is outgoing. At the joining Parallel Gateway the tokens of all incoming sequence flows will be collected. Only if all required tokens have arrived, they will be combined to one token again and the process can proceed.

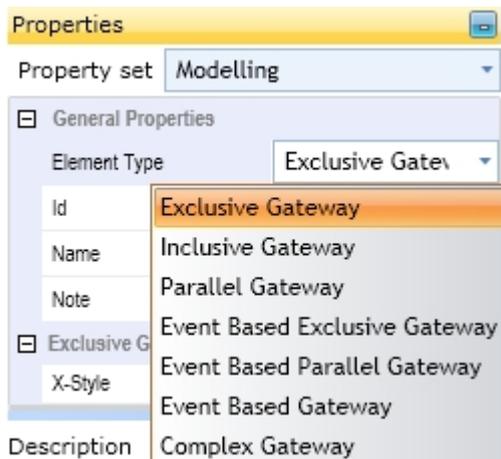
Gateways available in the collaboration and choreography toolbox (refer to: [Collaboration diagram](#) and [Choreography diagram](#)):



[The Properties Area](#) allows to edit the gateway properties:



The following [Properties](#) may be edited:  
 Element Type You may change the gateway type.



- Id** Id number, which is allocated by the system
- Name** The name of the element. This name is displayed as text underneath the element.
- Note** A short information about the element
- X-Style** Determines, if the "X" in the gateway symbol is displayed

## Events

### BPMN definition:

An "Event" occurs in Collaboration and in Choreography diagrams. Events affect the process flow. There are three types of events: Start, Intermediate, and End events.

Events can be

- "catching": by receiving a trigger, the event is considered to be occurred
- "throwing": activated by the process, the event will send a trigger

Event types:

	Catching		Throwing	
Flow Dimension	Start	Intermediate		End
Type Dimension	Event indicates start of process	wait for	trigger	Event indicates end of process
General events				
Timer event				
Message event				
Signal event				

Compensation event				
Error event				
Escalation event				
Conditional event				
Multiple event				
Parallel Multiple event				
Cancel event				
Link event				
Terminate event				

Interrupting and non-interrupting events:

Subprocesses and activities may have an event attached to their boundary. In this case the event may act interrupting as well as non-interrupting. The boundary of non-interrupting events are dashed. The markers are similar.

Note: Not all types of events can be attached

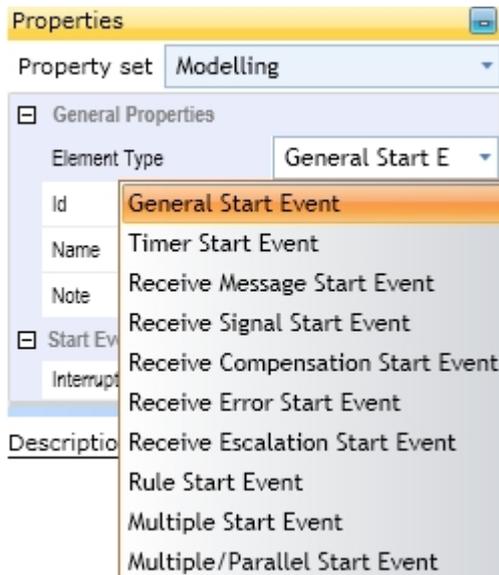
The options "interrupting" and "non-interrupting!" are available in the [The Properties Area](#). For details, refer to the bottom of this page.

Events available in the collaboration and choreography toolbox (refer to: [Collaboration diagram](#) and [Choreography diagram](#)):



The following [Properties](#) may be specified:

Element Type You may change the Event type.



- Id** Id number, which is allocated by the system
- Name** The name of the element. This name is displayed as text underneath the element.
- Note** A short information about the element
- Interrupting** Determines, if "interrupting" takes effect

## Activities

### BPMN definition:

An activity is the work actually performed within a process.

Activity types are:

- **Task**  
A task is a unit of work, which cannot be broken down to a finer level of detail. Different types of tasks are identified within BPMN in order to specify their behavior. For details refer to table below.
- **Sub-Process**  
A Sub-Process consists of activities. It can be broken into a finer level of detail.
- **Call activity**  
A Call activity is a reusable Task

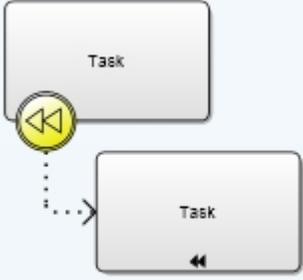
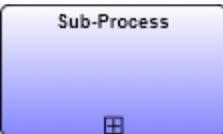
An Activity is a generic term for work that company performs in a Process. An Activity can be atomic or non-atomic (compound). The types of Activities that are a part of a Process Model are: Sub-Process and Task, which are rounded rectangles. Activities are used in both standard Processes and in Choreographies

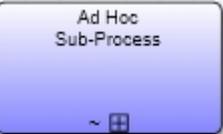
Symbol	Element	Description
	Task	

 User Task	User task	This task requires interactive user interaction
 Manual Task	Manual Task	A Manual Task is a Task that is expected to be performed without the aid of any business process execution engine or any application. An example of this could be a telephone technician installing a telephone at a customer location
 Service Task	Service Task	A Service Task will call a function (i.e. a web service)
 Receive Task	Receive Task	A Receive Task is a simple Task that is designed to wait for a Message to arrive from an external Participant (relative to the Process). Once the Message has been received, the Task is completed.
 Send Task	Send Task	A Send Task is a simple Task that is designed to send a Message to an external Participant (relative to the Process). Once the Message has been sent, the Task is completed
 Business Rule Task	Business Rule Task	A Business Rule Task provides a mechanism for the Process to provide input to a Business Rules Engine and to get the output of calculations that the Business Rules Engine might provide.
 Script Task	Script Task	A Script Task is executed by a business process engine. The modeler or implementer defines a script in a language that the engine can interpret. When the Task is ready to start, the engine will execute the script. When the script is completed, the Task will also be completed
 Undefined Task	Undefined Task	A Task which is not further specified is called Abstract Task

Activity markers available for all task types

Symbol	Marker	Description	Property
 Compensation	Compensation	Compensation is concerned with undoing steps that were already successfully completed, because their results and possibly side effects are no longer desired and need to be reversed. If an Activity is still active, it cannot be compensated, but rather needs to be canceled. Cancellation in turn can result in compensation of already successfully completed portions of an active Activity, in case of a Sub-Process.	<input type="checkbox"/> Activity P <input type="checkbox"/> Compens <input type="checkbox"/> Looping <input type="checkbox"/> Reuseabl <input type="checkbox"/> Multiple I

			
	<p>Looping</p>	<p>The attributes of tasks will determine if they are repeated or performed once. There are two types of loops:</p> <ul style="list-style-type: none"> <li>• Standard</li> </ul> <p>A small looping indicator will be displayed at the bottom-center of the activity.</p> <p>After each run, it will be checked, if the loop (termination) conditions are reached. The number of runs is not determined.</p> <ul style="list-style-type: none"> <li>• Multi-Instance <ul style="list-style-type: none"> <li>○ Parallel</li> </ul> </li> </ul> <p>Three vertical lines shown at the bottom-center of the pool indicate a parallel multiple instance. Activities in the pool can follow a any order (can be parallel).</p> <ul style="list-style-type: none"> <li>○ Sequential</li> </ul> <p>Three horizontal lines shown at the bottom-center of the pool indicate a sequential multiple instance. Activities in the pool need to follow a certain order (sequence).</p>	<ul style="list-style-type: none"> <li>Activity P <ul style="list-style-type: none"> <li>Compens</li> <li>Looping</li> <li>Reuseabl</li> <li>Multiple In</li> </ul> </li> <li>Activity P <ul style="list-style-type: none"> <li>Compens</li> <li>Looping</li> <li>Reuseabl</li> <li>Multiple In</li> </ul> </li> <li>Workflow <ul style="list-style-type: none"> <li>Expressio</li> <li>Form</li> </ul> </li> <li>Activity P <ul style="list-style-type: none"> <li>Compens</li> <li>Looping</li> <li>Reuseabl</li> <li>Multiple In</li> </ul> </li> <li>Workflow <ul style="list-style-type: none"> <li>Expressio</li> <li>Form</li> </ul> </li> </ul>
	<p>Sub-Process (collapsed view)</p>	<p>A Sub-Process consists of activities. It can be broken into a finer level of detail. )</p> <p>The Sub-Process can be in a collapsed view that hides its details or it can be in an expanded view that shows its details within the view of the Process in which it is contained. In the collapsed form, the Sub-Process object uses a marker to distinguish it as a Sub-Process, rather than a Task.</p>	

	Sub-process (expanded view)		
	Transaction	<p>A transaction is a Sub-Process that is supported by a special protocol that insures that all parties involved have complete agreement that the activity should be completed or cancelled. The attributes of the activity will determine if the activity is a transaction. A double-lined boundary indicates that the Sub-Process is a Transaction.</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Subprocess</li> <li>Ad Hoc</li> <li>Expanded</li> <li>Subprocess</li> <li>Type</li> <li><input type="checkbox"/> Workflow</li> <li>Expression</li> </ul>
	Event Sub-Process	<p>An Event Sub-Process is a specialized Sub-Process that is used within a Process (or Sub-Process). A Sub-Process is defined as an Event Sub-Process when its triggeredByEvent attribute is set to true.</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Subprocess</li> <li>Ad Hoc</li> <li>Expanded</li> <li>Subprocess</li> <li>Type</li> <li><input type="checkbox"/> Workflow</li> <li>Expression</li> </ul>
	Ad Hoc Sub-Process	<p>The Activities of an Ad Hoc Sub-Process are controlled by the process performers, not by sequence flows.</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Subprocess</li> <li>Ad Hoc</li> <li>Expanded</li> <li>Subprocess</li> <li>Type</li> </ul>
  	<p>Call Activity</p>  <p>Call to a process</p>	<p>A Call Activity is a wrapper for a globally defined Task or Process reused in the current Process. A call to a Process is marked with a symbol</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Activity P</li> <li>Compens</li> <li>Looping</li> <li>Reuseabl</li> <li>Multiple In</li> <li><input type="checkbox"/> Activity P</li> <li>Compens</li> <li>Looping</li> <li>Reuseabl</li> <li>Multiple In</li> <li><input type="checkbox"/> Subprocess</li> <li>Ad Hoc</li> <li>Expanded</li> <li>Subprocess</li> <li>Type</li> </ul>

Activities available in the choreography toolbox (refer to: [Choreography diagram](#)):

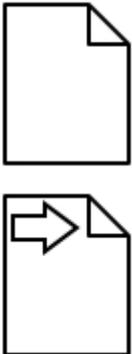
Notation	Element	Description	IYOPRO
Task Type	Task		
	Choreography task	A Choreography Task is an atomic Activity in a Choreography. It represents a set of one or more Message exchanges. Each Choreography Task involves two Participants. The name of the Choreography Task and each of the Participants are all displayed in the different bands that make up the shape's graphical notation. There are two or more Participant Bands and one Task Name Band.	Drag from toolbox
	Choreography Sub-process	A Sub-Process is a compound Activity that is included within a Process or Choreography. It is compound in that it can be broken down into a finer level of detail (a Process or Choreography) through a set of sub-Activities.	Drag from toolbox

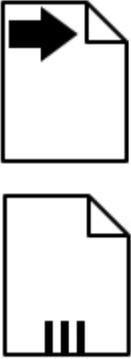
## Data

### BPMN definition:

Data is represented with the two elements:

1. Data Objects
2. Data Stores

Symbol	Element	Description
	Data Object	<p>General data object</p> <p>Data Input</p> <p>Data Output</p> <p>List Object</p>

		
	Data Store	Used to store Data permanent (i.E. a Database, a disk...)

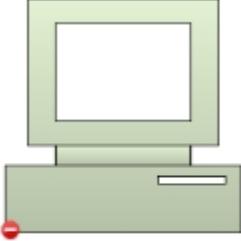
### Annotations and Artefacts

Annotations and Artifacts are used to provide additional information about the Process.

There are two standardized Artifacts, but modelers or modeling tools are free to add as many Artifacts as necessary. There could be additional BPMN efforts to standardize a larger set of Artifacts for general use or for vertical markets. The current set of Artifacts includes:

1. Group
2. Text Annotation

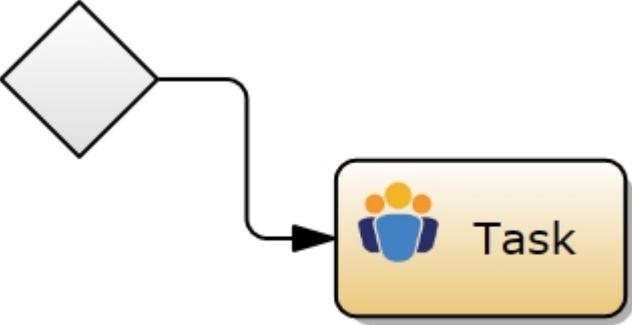
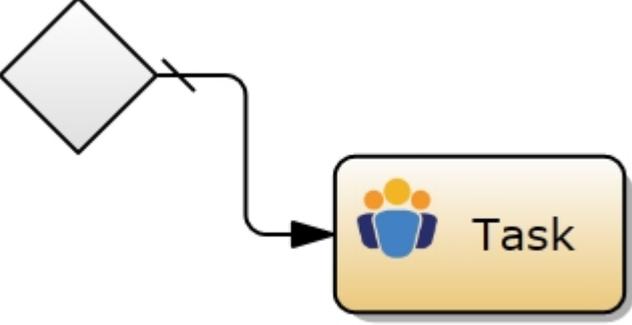
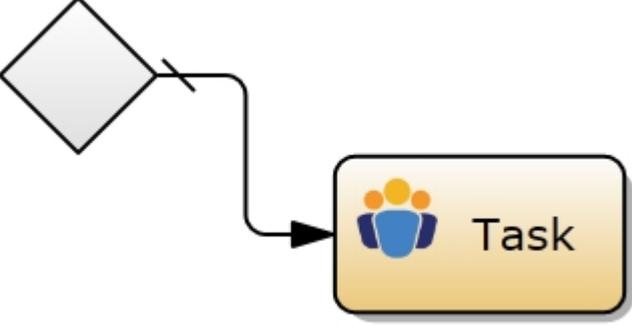
Symbol	Element	Description
	Annotation	Use the Annotation to attach Comments to Diagram Elements
	Group	Used to group elements together

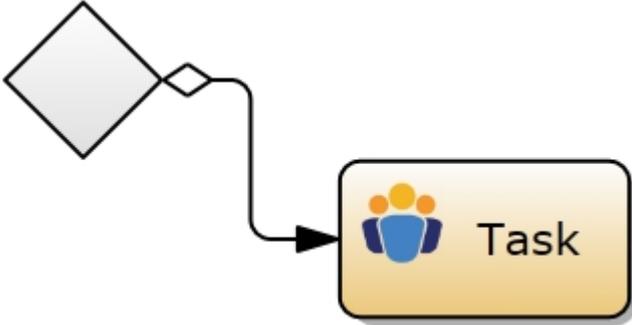
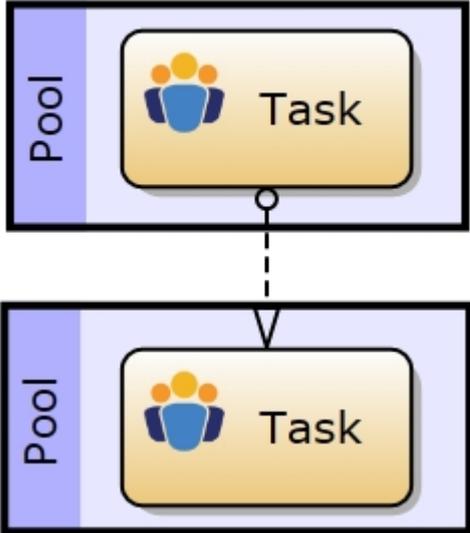
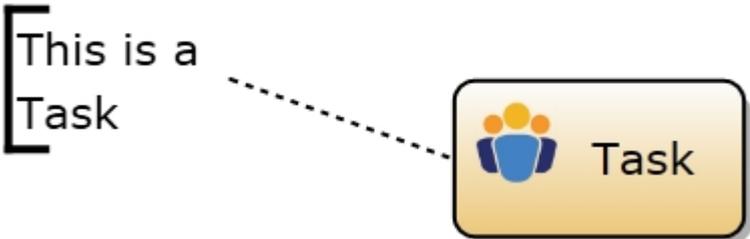
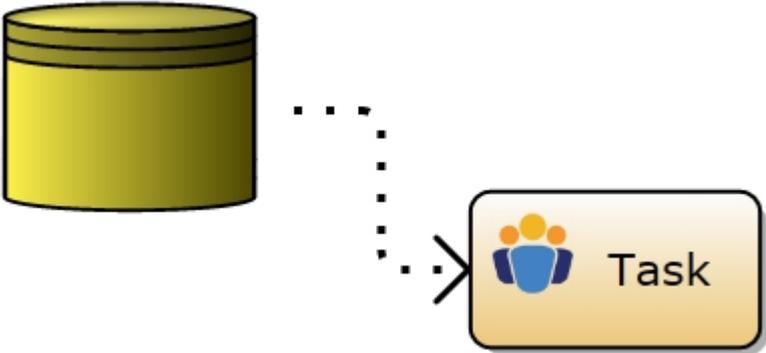
	IT-System	Use this Shape to attach an IT-System to Activities
---	-----------	---

### Flow Objects

There are four ways of connecting the Flow Objects to each other or other information.

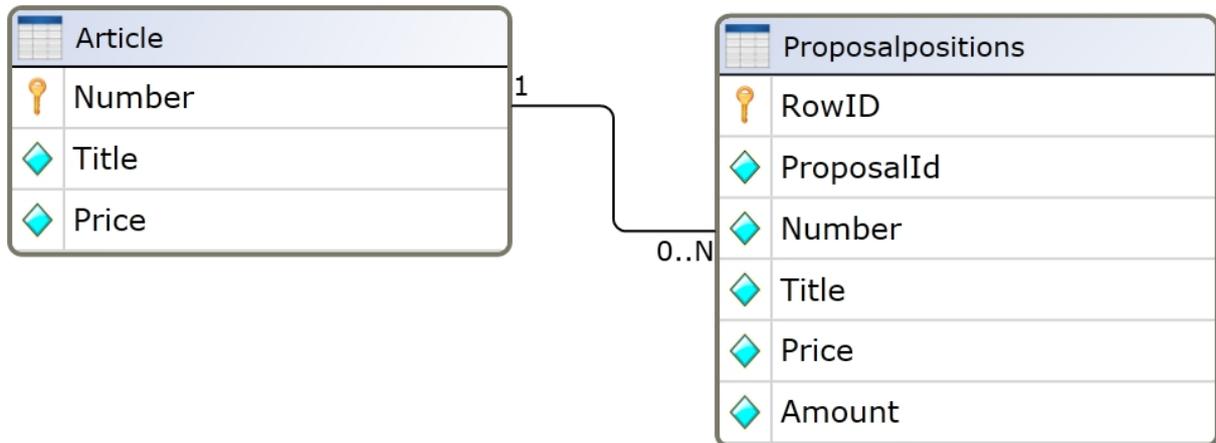
1. Sequence Flows
2. Message Flows
3. Associations
4. Data Associations

Symbol	Element	Description
	Sequence Flow	Normal Sequence flow
		Default Sequence flow
		Conditional Sequence flow

		
	<p>Message Flow</p>	<p>Represents the flow of a</p>
	<p>Association</p>	<p>I.E. Annotation Association</p>
	<p>Data Association</p>	<p>Used to associate a data</p>

Schema diagram  
 Schema diagram

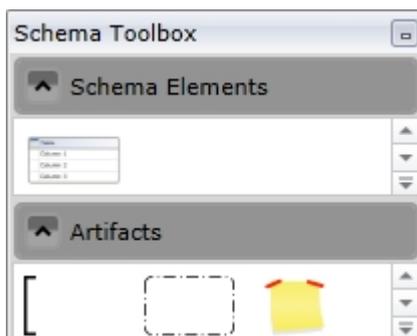
Schema diagrams provide a way to model the required database schema used behind [data stores](#) in a workflow szenario.



Sample Schema Diagram

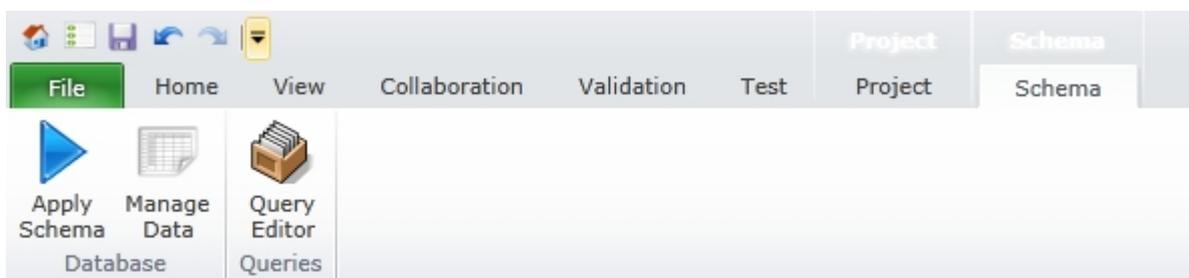
The diagram type "Schema diagram" is displayed together with the Schema Toolbox.

### Schema Toolbox



### Schema Context Tab

When editing a schema diagram you'll notice the schema contextual ribbon tab. The schema contextual ribbon tab provides additional options to your schema diagram.



### Apply Schema

To create a real database from your schema definition you have to apply your schema. If you press the apply schema button, your schema will either be created or updated to your new schema definition.

If your schema is not valid, you'll receive error messages in the [output window](#).

## Manage Data

To edit the data in the underlying database table follow this [link](#).

## Query Editor

To edit Queries for the Workflow Execution follow this [link](#).

## Schema table

A Schema Diagram consists out of one or more Schema Tables.

## Create Schema Table

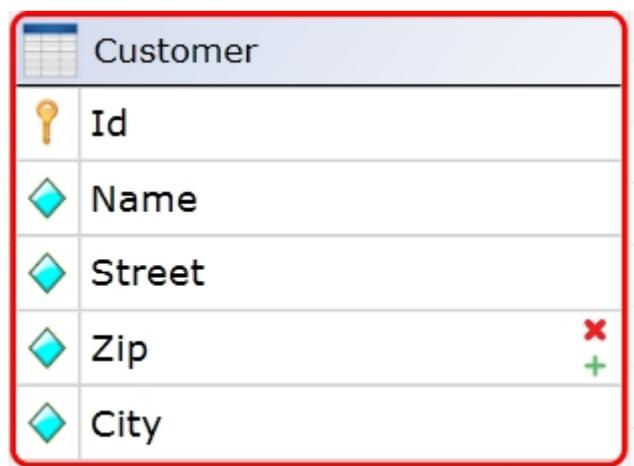
To create a Schema Table drag a Schema Table from the Schema Toolbox into the Work area.

## Edit Schema Table Name

Edit the Table Name either in the [Properties Area](#) or double click the Heading of the Schema.

## Add/Remove Columns

To add/remove columns move the mouse over a column of a selected schema table and click on the appearing +/- sign to add or remove a column.



Customer	
🔑	Id
🔹	Name
🔹	Street
🔹	Zip
🔹	City

## Edit Column Properties

To Edit Column Properties select the requested Column. The [Properties Area](#) contains a section "Column Properties" where you may adjust your settings.

Properties	
Property set: Modelling	
[-] General Properties	
Element Type	Schema Table
Id	2
Name	Customer
Note	
[-] Workflow Properties	
Variable Name	
[-] Column Properties	
Size	0
Data Type	String
Name	Zip
Primary Key	<input type="checkbox"/>
[-] Column Workflow Properties	
DataSet Key	<input type="checkbox"/>
Variable Name	

The following Settings are available

**Size** If "Data Type" is "String" this represents the size of the underlying database column. Zero (0) indicates a Huge Column size, that is limited by the Database specification only.

**Data Type** The Possible Data Types are  
 Variant - unspecified  
 String - A character based column  
 Integer - may contain integer numbers  
 Float - may contain floating point numbers  
 DateTime - may contain dates  
 Currency - may contain currency values  
 Guid - may contain Globally Unique Identifiers  
 Object - may contain an Object (i.e. List, Dictionary...)



**Important notice:**  
 Because the Variant datatype is unspecified the data **MUST** be in the format of a python string. That means, strings **MUST** be enclosed in quotation marks.

**Name** The name of the column

**Identity** Only available if Data Type is Number

or Guid - specifies that this column is an Identity column. Identity column values are automatically generated by the database.

**Primary Key** Primary Key columns identify the database row unique. Each table requires at least one primary key column. This option is often used in conjunction with the Identity setting.

## Manage Data

### Manage Data

To edit the data in the underlying database table of a schema table select the schema table and click on "Manage Data" in the [Schema Context Tab](#). The appearing dialog allows you to

1. Export Data  
Export the content of your table into a csv formatted file
2. Import data  
Import data from a csv formatted file into your table
3. Edit a row  
Doubleclick onto a row to edit the row content
4. Delete a row  
Select the rows to delete and press the Delete button

Type	CK01	CK02	CK03	CK04	CK05	CK06	CK07
"Roman"	0	0	1	1	0	0	1
","	1	0	0	1	1	1	0
"m"	1	0	1	0	0	1	0
"Skater"	1	0	0	1	1	0	1
""	0	0	0	0	0	0	0

## Query Editor

### Query Editor

To utilize Schema definitions within workflows you may wish to define queries to access data. Queries provide an easy way to load and save data from within a data store in a workflow scenario.

To define and edit queries select a schema definition and click on "Query Editor" in the [Schema Context Tab](#). The appearing dialog allows you to

1. Create a new Query or select an existing one
2. Select the columns of your result set  
These columns will be retrieved from or stored/updated to the database when utilizing the query

definition

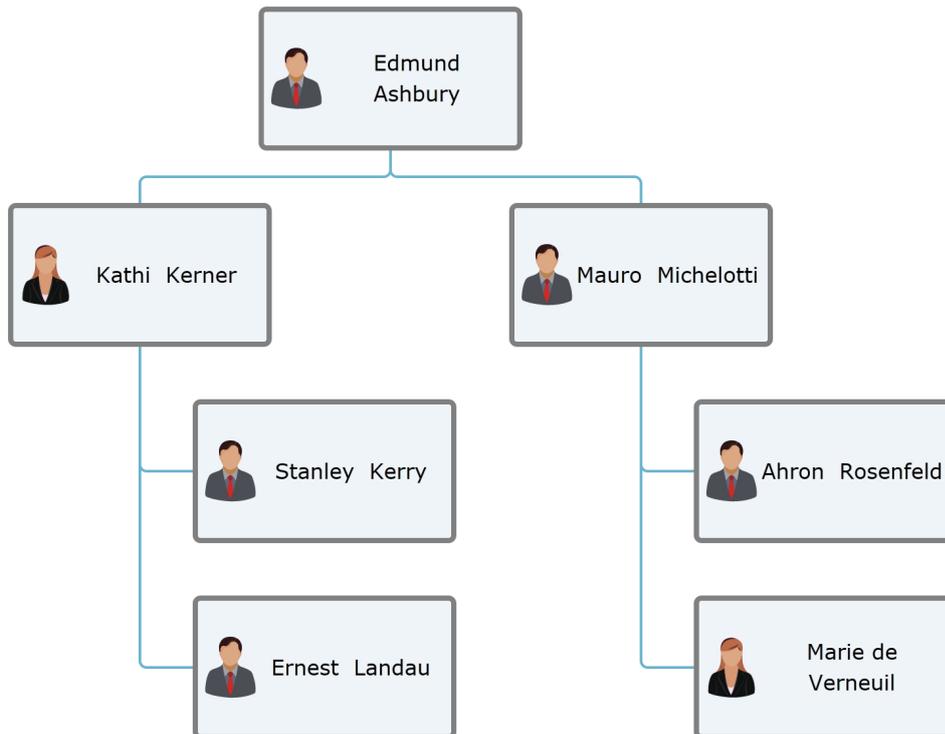
3. Select the columns of your query set  
These column(s) are used to identify the rows which will be retrieved or saved from/to the database.  
Usually you specify the primary key columns here.



## Organization diagram

## Organization diagram

Organization diagrams provide a way to model the dependencies of users, groups and roles used for workflow execution. The usage of roles provide a way to separate the workflow definition from the real human resources in order to reuse the workflow definition for different purposes.

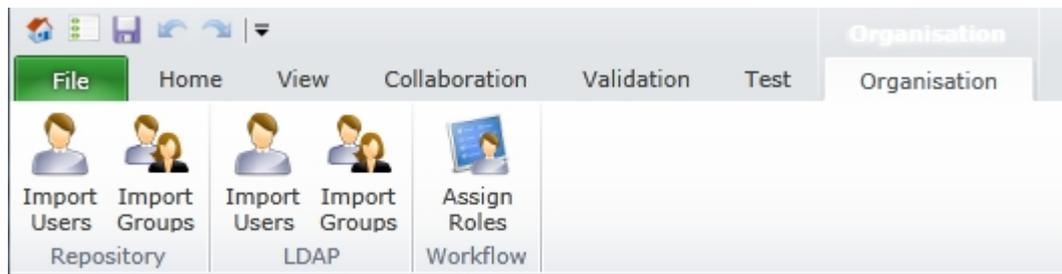


Sample Organization Diagram

The diagram type "Schema diagram" does not provide a Toolbox.

### Organisation Context Tab

When editing an organisation diagram you'll notice the organisation contextual ribbon tab. The organisation contextual ribbon tab provides additional options to your organisation diagram.



### Import Users from Repository

To import users from the repository follow this [link](#).

### Import Groups from Repository

To import groups from the repository follow this [link](#).

### Import Users from LDAP (OOB only)

To import users from the active directory follow this [link](#).

### Import Groups from LDAP (OOB only)

To import groups from the active directory follow this [link](#).

### Assign Roles

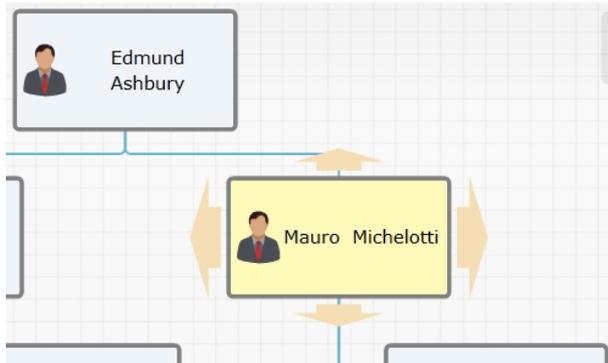
To assign roles to Organisation elements follow this [link](#).

## Human resource

An Organization Diagram consists out of one or more Resources.

### Human Resource

A Human Resource represents a person in general. You may specify it's name, email address, phone etc. In order to add a Human Resource move the mouse over an existing item and click on one of the appearing arrows. The arrow indicates the direction, where the new resource will be inserted.



### Resource Properties

To edit Resource Properties select the requested resource and adjust the requested settings in the [Properties Area](#).

Properties	
Property set: Modelling	
Organisation Properties	
Roles	
Repository Resource Properties	
Id	0
Layout Properties	
Alignment	TipOver
General Properties	
Element Type	Human
Id	6
Note	
Name	Kathi Kerner
Human Resource Properties	
E-Mail	Kerner@mycpmpar
First Name	Kathi
Last Name	Kerner
Mobile Phone	
Phone	
Position	Leader Product Mar
Sex	Female

## Repository users and usergroups

### Repository user and usergroups

To import users and usergroups from the repository select the organization diagram. The [Organisation Context Tab](#) will appear which contains the following features:

#### Import Users from Repository

In order to import users already defined in the repository select the root resource in your organization diagram where to subordinate the new repository users. Then click on the "Import Users" button of the repository group. You may select one or more users from a list of available users. The displayed users can be filtered by their names. Click OK to append the users.

#### Import Groups from Repository

In order to import groups already defined in the repository select the root resource in your organization diagram where to subordinate the new repository groups. Then click on the "Import Groups" button of the repository group. You may select one or more groups from a list of available groups. The displayed groups can be filtered by their names. Click OK to append the groups and their containing users.

## LDAP import

### LDAP user and usergroups (OOB only)

Note: Importing structural data from your active directory requires using IYOPRO in OOB Mode. Otherwise IYOPRO is not allowed to access these information

To import users and usergroups from the active directory select the organization diagram. The [Organisation Context Tab](#) will appear which contains the following features:

#### Import Users from LDAP

In order to import users out of you active directory select the root resource in your organization diagram where to subordinate the new users. Then click on the "Import Users" button of the LDAP group. You may select one or more users from a list of available users. The displayed users can be filtered by their names. Click OK to append the users.

#### Import Groups from LDAP

In order to import groups out of you active directory select the root resource in your organization diagram where to subordinate the new groups. Then click on the "Import Groups" button of the LDAP group. You may select one or more groups from a list of available groups. The displayed groups can be filtered by their names. Click OK to append the groups and their containing users.

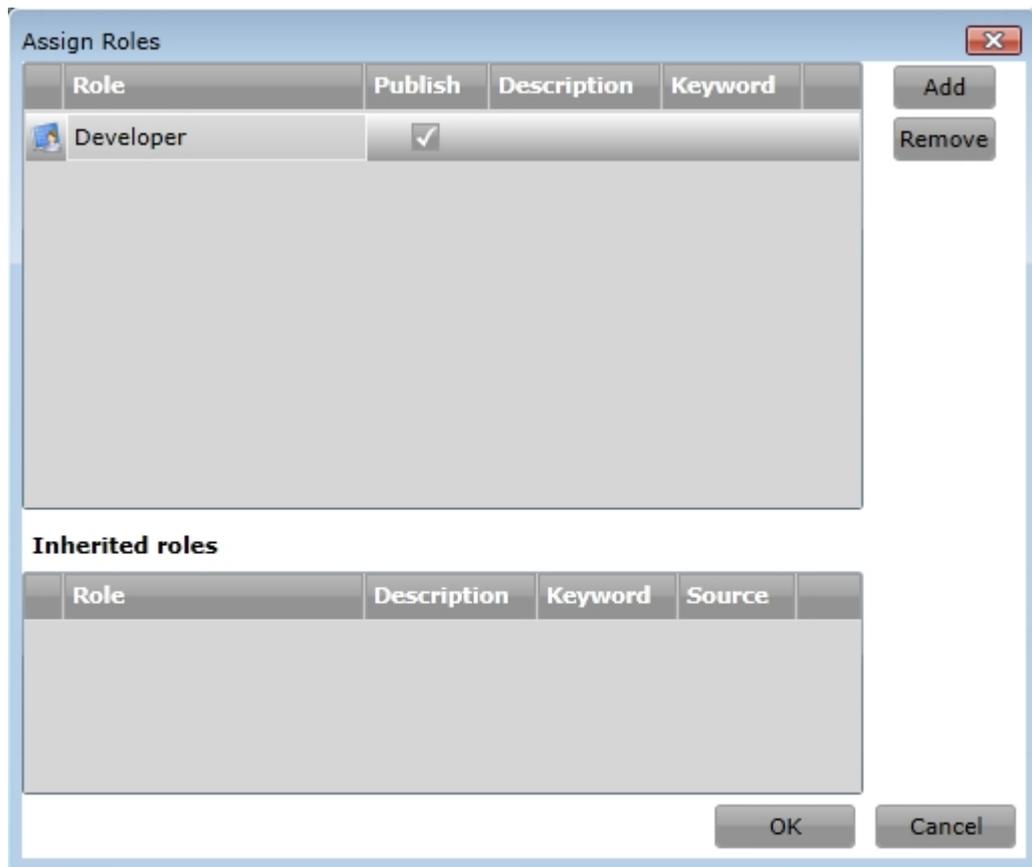
## Assigning roles

### Prerequisites

To assign roles to users select the organization diagram. The [Organisation Context Tab](#) will appear from which you may select the following feature:

#### Assign roles

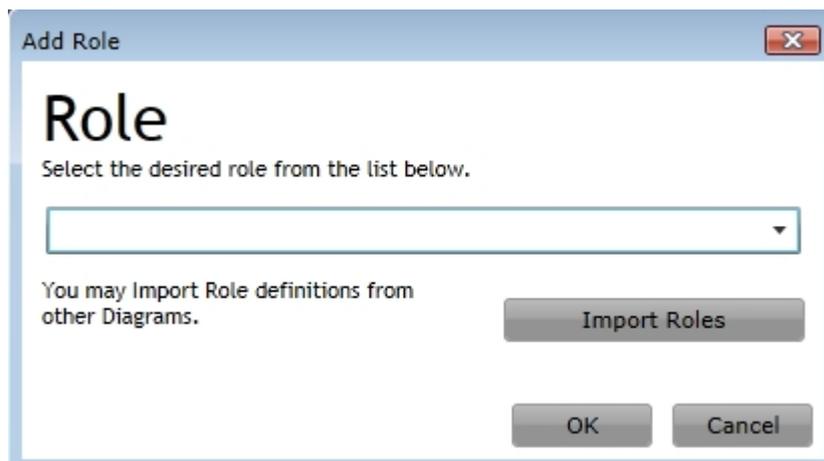
In order to assign roles to structure members of your organization diagram select the resource where to assign a role. Then click on the "Assign Roles" button of the Workflow group. In the appearing Assign Roles dialog you may add or remove one or more roles from the selected element.



If you wish to specify, that not only the selected diagram element but also all subordinates of the selected element will be included in the role, ensure that the Publish switch is checked.  
The roles already inherited by the selected element are shown in the lower part of the dialog.

## Add Role

To add a role press the Add Button. In the Add Role dialog select a role from the list of available roles.



If the desired role does not exist you may either select <new> from the list of available roles or click on "Import Roles" to import role definitions from other diagrams.

If you click on <new>, you may define a new role which consists of a name, a short description and a keyword.

The keyword is a value that can be set additionally. It can e.g. be used as an identifier by an external program.

## Edit Role

To edit a role, double-click on the role to be changed. In the dialog that opens, you can now edit the description and the keyword of the selected role.

## Remove Role

To remove a role from the selected element, select the role to remove and press the Remove button.

## Process map

### Process Map

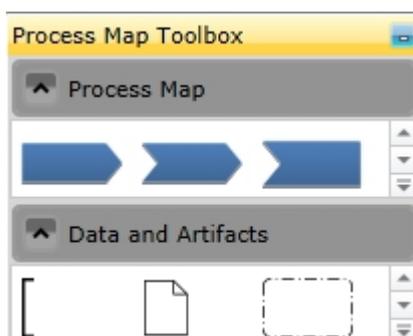
Process Maps visually describe the flow of activities and processes without the need of formal correctness. The only goal they archive is to give an overview over the process landscape and to provide a start point where to navigate from.



Sample Process Map

The diagram type "Process Map" is displayed together with the Process Map Toolbox.

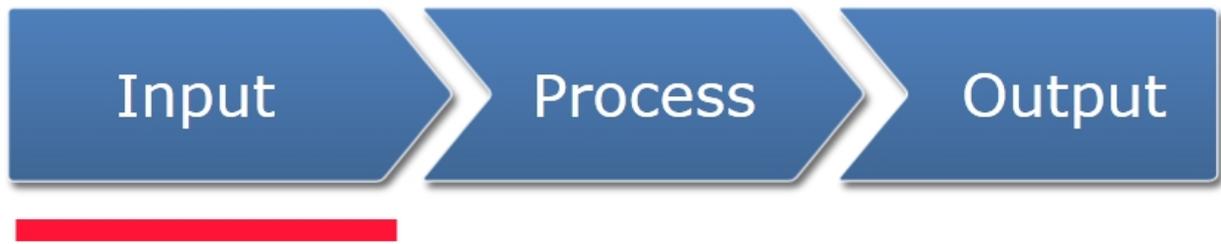
## Process Map Toolbox



## Send element

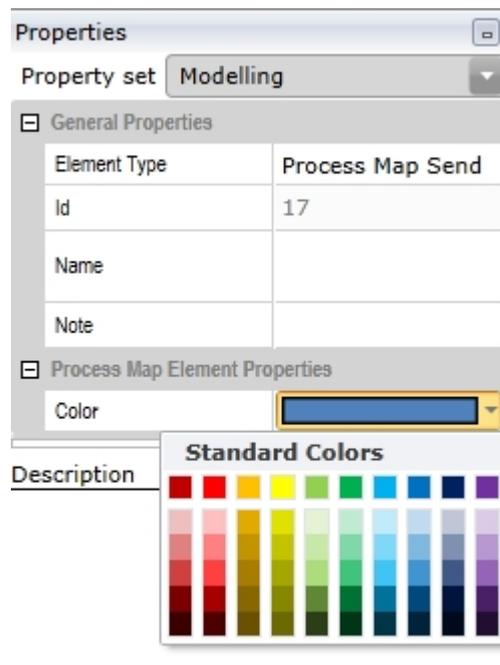
### Send Element

The send element is used to visualize input information for other elements (i.e. activities / processes)



It acts as sender of information to other elements whereas pure senders do not receive information (or it is not wanted to show possible input information).

Senders utilize the general properties. In addition you may specify the senders diagram color.



Process element

## Process Element

The process element is used to visualize an activity which receives an input and generates an output.



It acts as receiver as well as sender of information.

Processes utilize the general properties. In addition you may specify the senders diagram color.

The following [Properties](#) may be edited:

Element Type	You may change the element type.
Id	Id number, which is allocated by the system
Name	The name of the element. This name is displayed as text underneath the element.
Note	A short information about the element
Color	Specify the color of the process in the diagram
Expanded	Toggle this button to show an assigned subdiagram within the boundaries of the process element.
Process	Use this to link another diagram to this process. You may then navigate through the toolbar of this element to this diagram.

[Receive element](#)

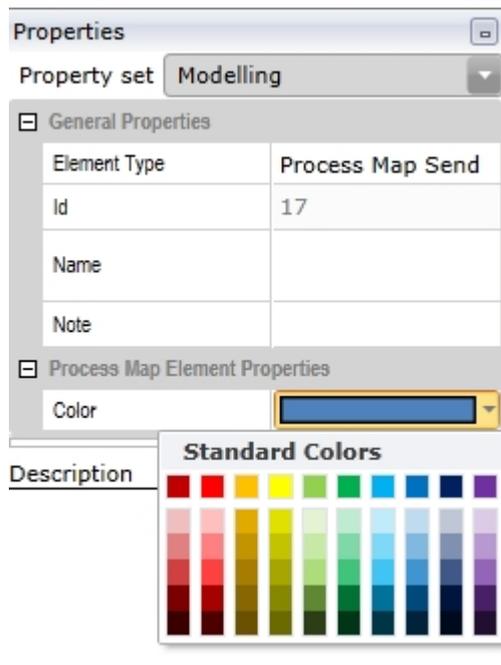
## Receive Element

The receive element is used to visualize output information



It acts as sink of information from other elements whereas pure receivers do not provide information (or it is not wanted to show possible output information).

Receivers utilize the general properties. In addition you may specify the receivers diagram color.



## Forms

## Forms

Forms are very important if you wish to create human workflows with YOPRO.

**Engineering Change Request (ECR)**

ECR Number  Title\*  Status

Requestor\*  Request Date\*  Decision By  Decision Date

**Reason for Change**

Safety  Financial  Manufacturing  Other

Description\*

Benefits\*

**Estimated Change Procedure**

Fast Track  Priority  1  2  3  4 Date Required

Details of Engineering Change

Drawing  Specification  Quality  BOM / Product Master

Developer  Develop Start  Approver  Approved At

Sample Form

Forms can be created out of the following elements:

- [Heading](#)
- [Text](#)
- [RichText](#)
- [Textbox](#)
- [Textarea](#)
- [Dropdown](#)
- [Checkbox](#)
- [Radiobutton](#)
- [Datepicker](#)
- [DateRangePicker](#)
- [Fileupload](#)
- [Hyperlink](#)
- [Button](#)
- [Group](#)
- [Page Break](#)
- [Datagrid](#)
- [HTML Control](#)
- [QR-Code](#)

## General Element Properties

All form elements provide a "General Properties" section in the [properties](#) dialog

Properties	
Property set	Modeling
^ General Properties	
Width	100 %
Max Width	0
Disabled	<input type="checkbox"/> <a href="#">Edit</a>
Id	6
Label	Enter your Text here
Visibility	<input type="checkbox"/> <a href="#">Edit</a>

The following settings are available

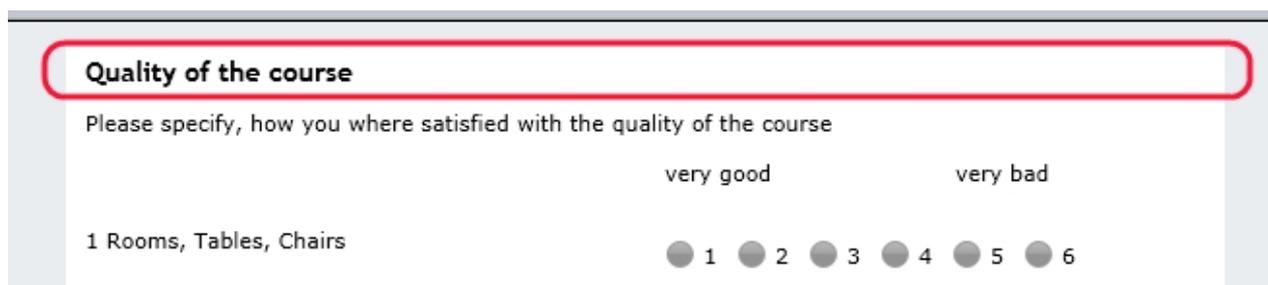
Width	Specify the width of the element on the form. You may choose between 25,50,75 and 100%
Max Width	Specify the maximum width of the element on the form. Width is based on this.
Id	Id number, which is allocated by the system
Label	The label of the element which usually represents the element in the form
Disabled	You may specify a <a href="#">condition</a> , when the element is disabled (i.e. cannot receive input) or not.
Visibility	You may specify a <a href="#">condition</a> , when the element is visible or not. This is usually used, if data can only be entered in case of an input of another field.

## Heading

### Heading

A Heading represents a headline

In order to add a Heading drag the shape from the Form Toolbox onto the Form and drop it at the desired location.



The image shows a sample form with a heading. The heading is "Quality of the course" and is highlighted with a red rounded rectangle. Below the heading is the text "Please specify, how you where satisfied with the quality of the course". There are two labels, "very good" and "very bad", positioned above a row of six radio buttons numbered 1 to 6. The text "1 Rooms, Tables, Chairs" is located below the radio buttons.

Sample Heading

## Properties

This Element utilizes beside the [General Element Properties](#) the following properties:

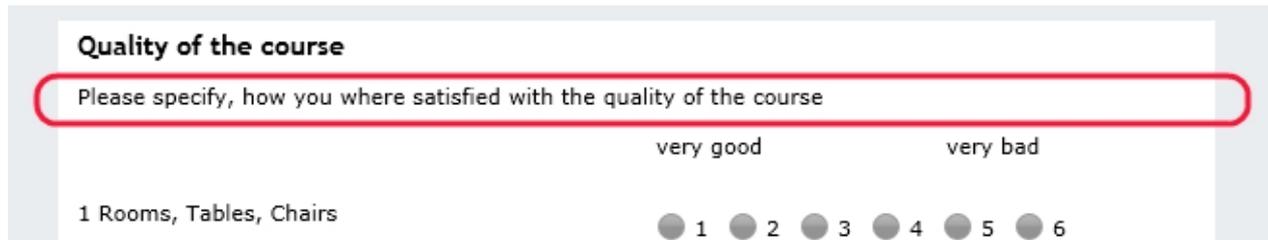
none.

Text

## Text

A Text represents a simple text.

In order to add a Text drag the shape from the Form Toolbox onto the Form and drop it at the desired location.



The image shows a sample form titled "Quality of the course". It features a text input field with the placeholder text "Please specify, how you where satisfied with the quality of the course". Below the input field, there are two labels: "very good" and "very bad". At the bottom, there is a row of six radio buttons labeled 1 through 6. The text "1 Rooms, Tables, Chairs" is visible on the left side of the form.

Sample Text

## Properties

This Element utilizes beside the [General Element Properties](#) the following properties:

none.

## Comparison of Text and RichText

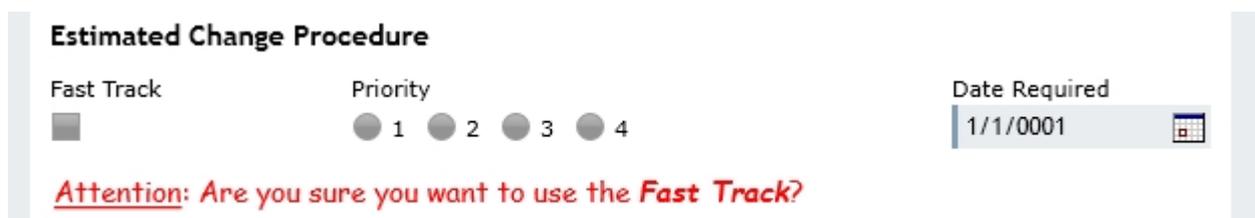
With RichText you can insert a variety of visual characteristics into your text and thus style your text according to your ideas. With Text you just write a simple text. So on the one hand, you have more design options with RichText, on the other hand, adding text with the Text element is faster, since there are no stylistic customization options. The Text element thus has the advantage that the displayed text looks the same everywhere and there is no further effort for you. Another advantage is a saving in performance. We therefore recommend using Text where RichText is not required.

RichText

## RichText

A RichText represents more than a simple text. It supports text formatting as well as different fonts, font sizes and colored text.

In order to add a RichText, drag the shape from the Form Toolbox onto the Form and drop it at the desired location.



The image shows a sample RichText form titled "Estimated Change Procedure". It contains a "Fast Track" checkbox, a "Priority" section with four radio buttons labeled 1, 2, 3, and 4, and a "Date Required" field with the value "1/1/0001" and a calendar icon. Below these fields, there is a red warning message: "Attention: Are you sure you want to use the Fast Track?"

Sample RichText

## Properties

This Element utilizes beside the [General Element Properties](#) the following properties:

none.

## Comparison of Text and RichText

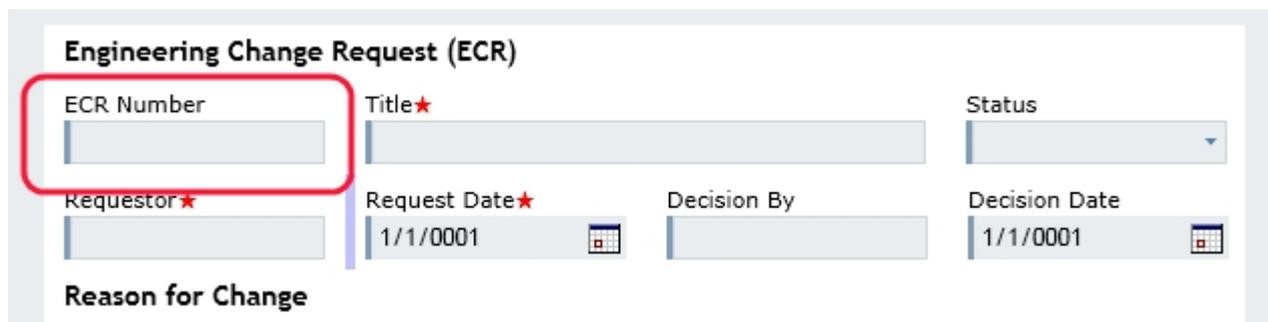
With RichText you can insert a variety of visual characteristics into your text and thus style your text according to your ideas. With Text you just write a simple text. So on the one hand, you have more design options with RichText, on the other hand, adding text with the Text element is faster, since there are no stylistic customization options. The Text element thus has the advantage that the displayed text looks the same everywhere and there is no further effort for you. Another advantage is a saving in performance. We therefore recommend using Text where RichText is not required.

### Textbox

## Textbox

A Textbox is used to input a single line of text.

In order to add a Textbox drag the shape from the Form Toolbox onto the Form and drop it at the desired location.



The image shows a sample form titled "Engineering Change Request (ECR)". The form contains several input fields: "ECR Number" (highlighted with a red box), "Requestor" (with a red star), "Title" (with a red star), "Request Date" (with a red star and a calendar icon), "Decision By", "Status" (a dropdown menu), and "Decision Date" (with a calendar icon). Below these fields is a section labeled "Reason for Change".

Sample Textbox

## Properties

This Element utilizes beside the [General Element Properties](#) the following properties:

Properties	
Property set: Modeling	
[-] General Properties	
Width	25 %
Disabled	<input type="checkbox"/> Edit
Id	7
Label	Requestor
Visibility	<input type="checkbox"/> Edit
[-] Input Properties	
Hint	
Required	<input checked="" type="checkbox"/>
Required Mode	If enabled
Variable	RequestBy
[-] Text Input Properties	
Verify	None
[-] Text Box Properties	
Max Size	100
[-] Workflow Properties	
Action	<input type="checkbox"/> Edit
OnDrop	<input type="checkbox"/> Edit

## Input Parameters

- Hint** A hint for the user, what to input. Hints are usually shown as tooltips if the user moves the mouse over the element or as error message if the user inputs invalid data
- Required** Specify, if the input is required or not. A form may only be submitted if all required inputs have been made.
- VariableName** The name of the variable, where the input will be stored in or where the preset value will be taken from.
- Verify** An input verification, which will control the input. You may specify
- None  
No input verification
  - Integer  
Only integer values may be specified
  - Float  
Only floating-point numbers may be specified
  - Currency  
Only a currency may be specified
  - Email  
Only an Email address may be specified

MaxSize	The maximum number of characters which may be entered
Action	You may specify an expression which will be executed, if the user presses return in the Textbox
OnDrop	You may specify an expression which will be executed, if the user drops text into the Textbox

## Textarea

### Textarea

A Textarea is used to input a multiple lines of text.

In order to add a Textarea drag the shape from the Form Toolbox onto the Form and drop it at the desired location.

The image shows a sample form titled "Engineering Change Request (ECR)". The form contains several input fields: "ECR Number", "Title" (marked with a red star), "Status" (a dropdown menu), "Requestor" (marked with a red star), "Request Date" (with a calendar icon and the value "1/1/0001"), "Decision By", and "Decision Date" (with a calendar icon and the value "1/1/0001"). Below these fields is a section titled "Reason for Change" with four radio button options: "Safety", "Financial", "Manufacturing", and "Other". At the bottom of the form is a large "Description" field (marked with a red star), which is highlighted with a red rectangular border. The form is set against a light gray background.

Sample Textarea

### Properties

This Element utilizes beside the [General Element Properties](#) the following properties:

Properties	
Property set: Modelling	
General Properties	
Width	100 %
Disabled	<input type="checkbox"/> Edit
Id	11
Label	Description
Visibility	<input type="checkbox"/> Edit
Input Properties	
Hint	
Required	<input checked="" type="checkbox"/>
VariableName	Description
Text Input Properties	
Verify	None
Text Area Properties	
Rows	5

## Input Parameters

Hint	A hint for the user, what to input. Hints are usually shown as tooltips if the user moves the mouse over the element or as error message if the user inputs invalid data
Required	Specify, if the input is required or not. A form may only be submitted if all required inputs have been made.
VariableName	The name of the variable, where the input will be stored in or where the preset value will be taken from.
Verify	An input verification, which will control the input. You may specify <ul style="list-style-type: none"> <li>● None No input verification</li> <li>● Integer Only integer values may be specified</li> <li>● Float Only floating-point numbers may be specified</li> <li>● Currency Only a currency may be specified</li> <li>● Email Only an Email address may be specified</li> </ul>
Rows	The number of rows visible in the form.

## Dropdown

### Dropdown

A Dropdown is used to select a value from a list.

In order to add a Dropdown drag the shape from the Form Toolbox onto the Form and drop it at the desired location.

The image shows a sample form titled "Engineering Change Request (ECR)". It contains several input fields: "ECR Number", "Requestor", "Title", "Request Date", "Decision By", and "Decision Date". The "Status" field is a dropdown menu, which is highlighted with a red rounded rectangle. Below the form, the text "Reason for Change" is visible.

Sample Dropdown

### Properties

This Element utilizes beside the [General Element Properties](#) the following properties:

The image shows a screenshot of the "Properties" window for a dropdown menu. The "Property set" is "Modellierung". The properties are organized into three sections: "General Properties", "Input Properties", and "DropDown Properties".

General Properties	
Width	50 %
Disabled	<input type="checkbox"/> Edit
Id	5
Label	Status
Visibility	<input type="checkbox"/> Edit

Input Properties	
Hint	
Required	<input checked="" type="checkbox"/>
Variable	State

DropDown Properties	
Options	In Decision (1) Edit
Size	1

### Input Parameters

- Hint** A hint for the user, what to input. Hints are usually shown as tooltips if the user moves the mouse over the element or as error message if the user inputs invalid data
- Required** Specify, if the input is required or not. A form may only be submitted if all required inputs have been made.
- VariableName** The name of the variable, where the input will be stored in or where the preset value will be taken from.

- Options Specify the source, where the options are taken from.
- Size The Size of the Element. If 1 it will be shown as Combo Box, if >1 it will be shown as selectable list box.

## Drop Down Options

The user can choose between fixed values or a workflow attribute as option source. The workflow attribute is defined as a standalone global variable and not as part of the form variable.

Examples for workflow attribute as option source:

a)

```
list = ["Hello", "World"] →
```

b)

```
listOfDict = [{"key": "a", "val": "Hello"}, {"key": "b", "val": "World"}] →
```

## Checkbox

### Checkbox

A Checkbox is used to toggle one or more Yes/No states.

In order to add a Checkbox drag the shape from the Form Toolbox onto the Form and drop it at the desired location.

**Engineering Change Request (ECR)**

ECR Number  Title★  Status

Requestor★  Request Date★   Decision By  Decision Date

**Reason for Change**

Safety  Financial  Manufacturing  Other

Description★

Sample Checkbox

### Properties

This Element utilizes beside the [General Element Properties](#) the following properties:

Properties	
Property set	Modellierung
General Properties	
Width	50 %
Disabled	<input type="checkbox"/> Edit
Id	2
Label	Enter your Text here
Visibility	<input type="checkbox"/> Edit
Input Properties	
Hint	
Required	<input type="checkbox"/>
Variable	
CheckBox Properties	
Checked	<input type="checkbox"/>
Display inline	<input checked="" type="checkbox"/>
MinLength	1
Options	Option 1 Option 2 Option 3

## Input Parameters

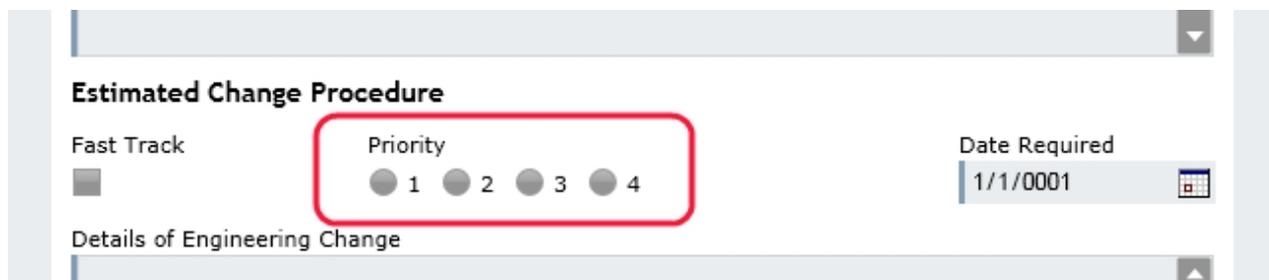
Hint	A hint for the user, what to input. Hints are usually shown as tooltips if the user moves the mouse over the element or as error message if the user inputs invalid data
Required	Specify, if the input is required or not. A form may only be submitted if all required inputs have been made.
Variable	The name of the variable, where the input will be stored in or where the preset value will be taken from.
Checked	Specify the default selected option.
Display inline	Decision between vertical or horizontal alignment of options.
MinLength	Specify how many checks must be made at least
Options	Specify for each option an entry in a separate line.

## Radiobutton

### Radiobutton

A Radiobutton is used to choose between multiple options.

In order to add a Radiobutton drag the shape from the Form Toolbox onto the Form and drop it at the desired location.



The image shows a sample form titled "Estimated Change Procedure". It contains several input fields: a "Fast Track" checkbox, a "Priority" field with four radio buttons labeled 1, 2, 3, and 4, and a "Date Required" field with a calendar icon. The "Priority" field is highlighted with a red rectangular box. Below the form, the text "Details of Engineering Change" is visible.

Sample Radiobutton

## Properties

This Element utilizes beside the [General Element Properties](#) the following properties:

Properties	
Property set	Modellierung
General Properties	
Width	50 %
Disabled	<input type="checkbox"/> Edit
Id	1
Label	Enter your Text here
Visibility	<input type="checkbox"/> Edit
Input Properties	
Hint	
Required	<input type="checkbox"/>
Variable	
CheckBox Properties	
Display inline	<input checked="" type="checkbox"/>
Radio Button Properties	
Options	Option 1 Option 2 Option 3
Selected	

## Input Parameters

Hint	A hint for the user, what to input. Hints are usually shown as tooltips if the user moves the mouse over the element or as error message if the user inputs invalid data
Required	Specify, if the input is required or not. A form may only be submitted if all required inputs have been made.
Variable	The name of the variable, where the input will be stored in or where the preset value will be taken from.
Display inline	Decision between vertical or horizontal alignment of options.
Options	Specify for each option an entry in a separate line.
Selected	Specify a default entry.

## Datepicker

### Datepicker

A Datepicker is used to select a date from within a calendar. In order to add a Datepicker drag the shape from the Form Toolbox onto the Form and drop it at the desired location.

### Engineering Change Request (ECR)

ECR Number  Title\*  Status

Requestor\*  Request Date\*  Decision By  Decision Date

Enter your Text here

**Reason for Change**

Sample Datepicker

## Properties

This Element utilizes beside the [General Element Properties](#) the following properties:

Properties	
Property set	Modellierung
General Properties	
Width	50 %
Disabled	<input type="checkbox"/> Edit
Id	2
Label	Enter your Text her
Visibility	<input type="checkbox"/> Edit
Input Properties	
Hint	<input type="text"/>
Required	<input type="checkbox"/>
Variable	<input type="text"/>
Date Picker Properties	
Skip Timezone	<input type="checkbox"/>

## Input Parameters

- Hint** A hint for the user, what to input. Hints are usually shown as tooltips if the user moves the mouse over the element or as error message if the user inputs invalid data
- Required** Specify, if the input is required or not. A form may only be submitted if all required inputs have been made.
- VariableName** The name of the variable, where the input will be stored in or where the preset value will be taken from.

## Date Picker Parameters

- Skip** Ignores the time zone if selected.
- Timezone** The exact date specified will be used.

## DateRangePicker

## DateRangePicker

A DateRangePicker is used to select a date range from within a calendar.

In order to add a DateRangePicker drag the shape from the Form Toolbox onto the Form and drop it at the desired location.

The screenshot shows a form titled "Vacation request" with a section for "Vacation duration". The text inside the section reads: "Please enter the desired vacation. You will be informed of the approval in the application process." Below this text are three input fields: "Personnel number", "Applicant", and "Available vacation days". The "Desired vacation" field is highlighted with a red box and contains a date range picker with two calendar icons and a minus sign between them.

Sample DateRangePicker

## Properties

This Element utilizes beside the [General Element Properties](#) the following properties:

Properties	
Property set	Modeling
^ General Properties	
Width	50 %
Max Width	0
Disabled	<a href="#">Edit</a>
Id	9
Label	Desired vacation
Visibility	<a href="#">Edit</a>
^ Input Properties	
Hint	
Required	<input type="checkbox"/>
Variable	

## Input Parameters

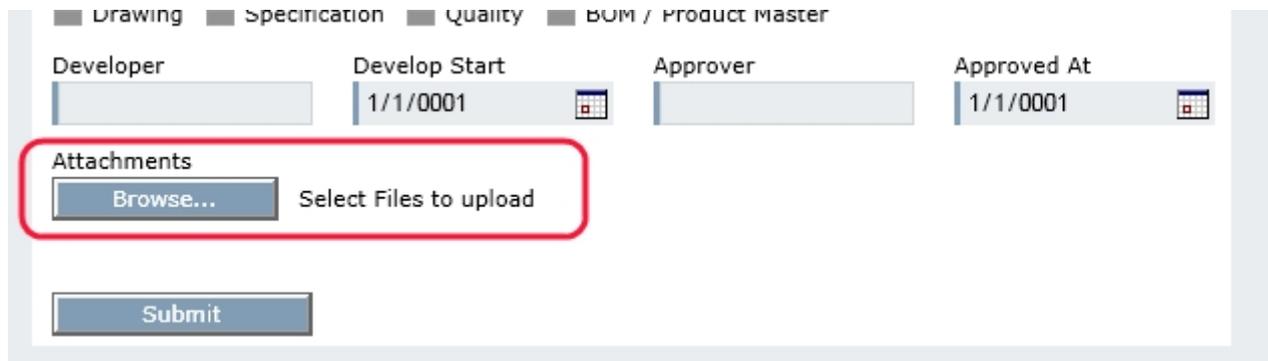
- Hint** A hint for the user, what to input. Hints are usually shown as tooltips if the user moves the mouse over the element or as error message if the user inputs invalid data
- Required** Specify, if the input is required or not. A form may only be submitted if all required inputs have been made.
- VariableName** The name of the variable, where the input will be stored in or where the preset value will be taken from.

## File Upload

### File Upload

A File Upload is used to upload a file to a process instance.

In order to add a File Upload drag the shape from the Form Toolbox onto the Form and drop it at the desired location.

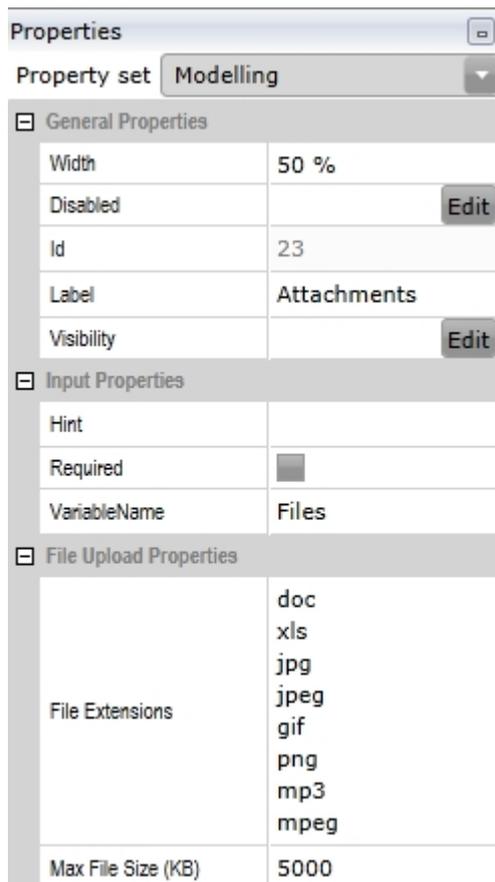


The image shows a sample form with several fields and a file upload section. At the top, there are four tabs: Drawing, Specification, Quality, and BOM / Product Master. Below the tabs are four input fields: Developer, Develop Start (with a date picker showing 1/1/0001), Approver, and Approved At (with a date picker showing 1/1/0001). A red box highlights the Attachments section, which contains a 'Browse...' button and the text 'Select Files to upload'. Below this is a 'Submit' button.

Sample File Upload

### Properties

This Element utilizes beside the [General Element Properties](#) the following properties:



The image shows a Properties window for a File Upload element. The 'Property set' is set to 'Modelling'. The window is divided into three sections: General Properties, Input Properties, and File Upload Properties.

General Properties	
Width	50 %
Disabled	<input type="checkbox"/> Edit
Id	23
Label	Attachments
Visibility	<input type="checkbox"/> Edit

Input Properties	
Hint	
Required	<input type="checkbox"/>
VariableName	Files

File Upload Properties	
File Extensions	doc xls jpg jpeg gif png mp3 mpeg
Max File Size (KB)	5000

### Input Parameters

Hint

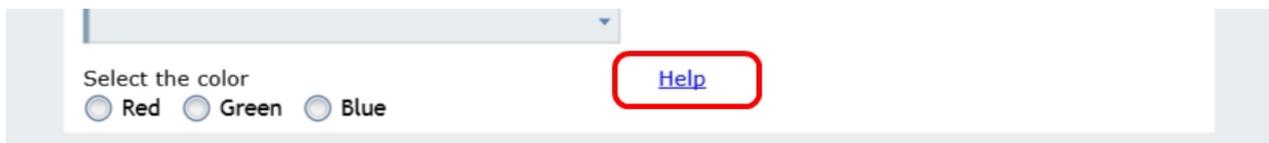
A hint for the user, what to input.  
Hints are usually shown as tooltips if the user moves the mouse over the element or as error message if the

	user inputs invalid data
Required	Specify, if the input is required or not. A form may only be submitted if all required inputs have been made.
VariableName	The name of the variable, where the input will be stored in or where the preset value will be taken from.
File Extensions	Specify a list of file extensions which are allowed to upload
Max File Size (KB)	Specify the maximum file size for upload.

## Hyperlink

### Hyperlink

A Hyperlink is used to reference information on external sites. In order to add a Hyperlink drag the shape from the Form Toolbox onto the Form and drop it at the desired location.



Sample Hyperlink

## Properties

This Element utilizes beside the [General Element Properties](#) the following properties:

Properties	
Property set: Modeling	
General Properties	
Width	100 %
Disabled	<input type="checkbox"/> Edit
Id	3
Label	Help
Visibility	<input type="checkbox"/> Edit
Link Properties	
Type	Static
Target	_blank
URL	

Properties	
Property set: Modeling	
General Properties	
Width	100 %
Disabled	<input type="checkbox"/> Edit
Id	3
Label	Help
Visibility	<input type="checkbox"/> Edit
Link Properties	
Type	Dynamic
Target	_blank
Variable	

## Input Parameters

Type	This indicates that the url will be static (specified in the URL property) or dynamic (read from the process instance attribute specified in the Variable property)
Target	The target where to open the reference. Leave it on _blank to open the reference in a new window.

Variable	The name of the variable, which will contain a dynamic url.
URL	The static url, i.e. https://www.iyopro.com

## Button

### Button

A Button is used to submit a form or execute an operation.

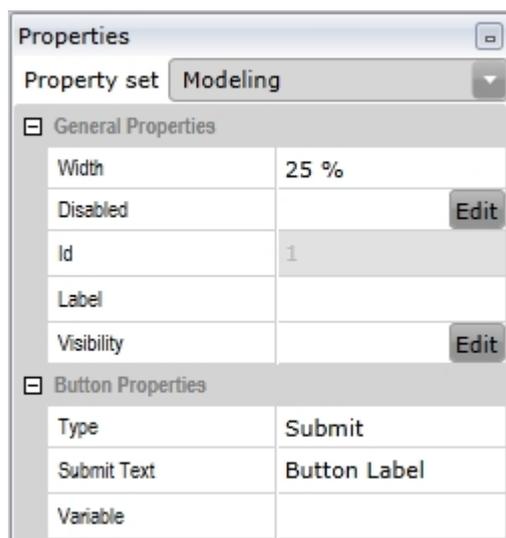
In order to add a Button drag the shape from the Form Toolbox onto the Form and drop it at the desired location.



Sample Button

### Properties

This Element utilizes beside the [General Element Properties](#) the following properties:



### Input Parameters

Type	Specify what kind of button it should be.
Submit Text	The label of the button
Variable	The name of the variable, which will be set to true when the button is pressed.

### Button Types

Submit	Submits the form
Reset	This indicates that the button will act

	as a Reset Form button, which will clear all input.
Action	You may specify an expression to be executed when the button is pressed. In this case, the button does not function as a submit button.
SubForm	Opens the linked subform when the button is pressed.

## Set link for subform

### Same Folder

If the subform is in the same folder as the calling form, the link looks like this:

- `../<name of the subform>`

### Different Folder

If the form is in a different folder, the link just mentioned is not sufficient.

If the subform is in an parent folder, the link could look like this:

- `../../<name of the subform>`

If the subform is in a subfolder, the link could look like this:

- `../< folder containing the calling form> / <subfolder name> / <subform name>`

### Find out the exact link

A possible way to find out the exact link is to drag the subform that you want to link into a collaboration diagram that is on the same level as the calling form.

The set link can now be seen in the properties.

Attention: The displayed link of a form, which is not a separate file but only a link in the project, points to the original. So here the method is not applicable.

## Group

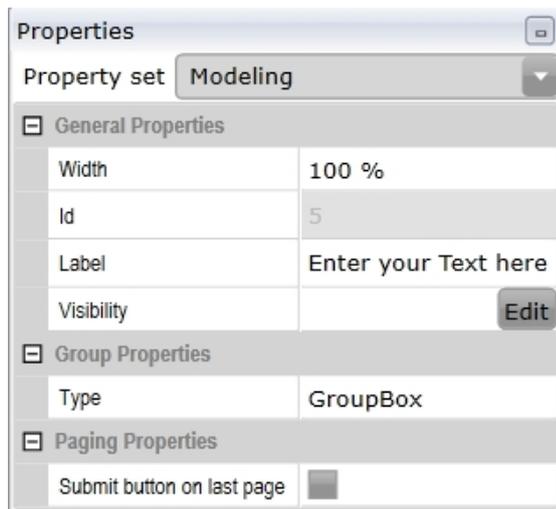
### Group

Groups can be used to either group elements visible on the form or to split columns into multiple sub columns. The IYOPRO form layout is based on a four column definition. If you drag a group onto a form or inside another group, the content of the group itself will also relay on this four column layout.

In order to add a Group drag the shape from the Form Toolbox onto the Form and drop it at the desired location.

## Properties

This Element utilizes beside the [General Element Properties](#) the following properties:



Based on the Type property the following visualizations can be archived:

### Group Box type

The Group Box type will group elements visible together. Please note, that the group element itself utilizes a margin and that the containing elements will be a little smaller than if they where used outside the group.

### None type

The None type will group elements together invisibly. The dashed border around the group is visible only in the form designer. Please note, that the group element itself does not utilize a margin and that the containing elements will align at the same bounds as if they where used outside the group.

With the None type you can split one or more columns into four new columns. Because groups may be nested, you can create any column style.

## Page Break Properties

If you use a [Page Break](#) within a Group, the last page of the Group may not be the last page of the entire form. Therefore, you may choose whether or not you want to display the submit button on the last page.

### Page Break

## Page Break

A Page Break is used to divide a form into multiple pages. The user may navigate between the pages by next and previous buttons.

On the last page, the Page Break element will have a submit button instead of a next button. You may decide if this submit button will be displayed or not. Go to the Paging Properties of the form or [group](#).

In order to add a Page Break drag the shape from the Form Toolbox onto the Form and drop it at the desired location.

Details of Engineering Change  
Sample Page Break

## Properties

This Element utilizes beside the [General Element Properties](#) the following properties:

Properties ☐

Property set **Modeling** ▼

[-] General Properties

Width	50 %	
Disabled		<b>Edit</b>
Id	7	
Label		
Visibility		<b>Edit</b>

[-] Workflow Properties

Back Action		<b>Edit</b>
Next Action		<b>Edit</b>

## Workflow Parameters

**Back Action** You may specify an expression to be executed when the "Back" button is pressed.

**Next Action** You may specify an expression to be executed when the "Next" button is pressed.

## Data Grid

### Data Grid

A Data Grid is used to display tabled data in a form.

In order to add a Data Grid drag the shape from the Form Toolbox onto the Form and drop it at the desired location.

sales@intellivate.com  
www.intellivate.com

**Invoice**

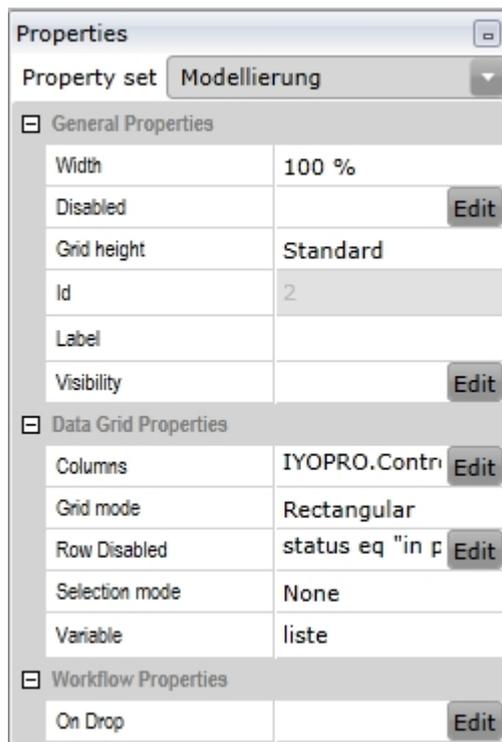
We delivered to you Invoice No. :

Amount	Article	Description	Price	Total	

Sample Data Grid

## Properties

This Element utilizes beside the [General Element Properties](#) the following properties:



## General Parameters

**Grid Height** Specify the height of the data grid.  
You may choose between

- Standard
- Custom
- Stretch (uses the rest of the available space)

**Grid Height Factor** Enter the factor by which the data grid should be higher (E.g.: 2 = double)

**Min Height** Specify the minimum height, that the data grid should have

**Stretch Proportion** Specify the size ratio of the control to other controls with stretch mode

## Data Grid Parameters

**Columns** Specify the Data Grid Columns with the Data Grid Column Editor

**Grid mode** Select the structure of the rows. You may choose between

- Rectangular
- Tree

**Tree variable** Choose a variable that identifies the child elements

**Row Disabled** You may disable specific rows

**Selection mode** Specify whether rows of the data grid can be selected

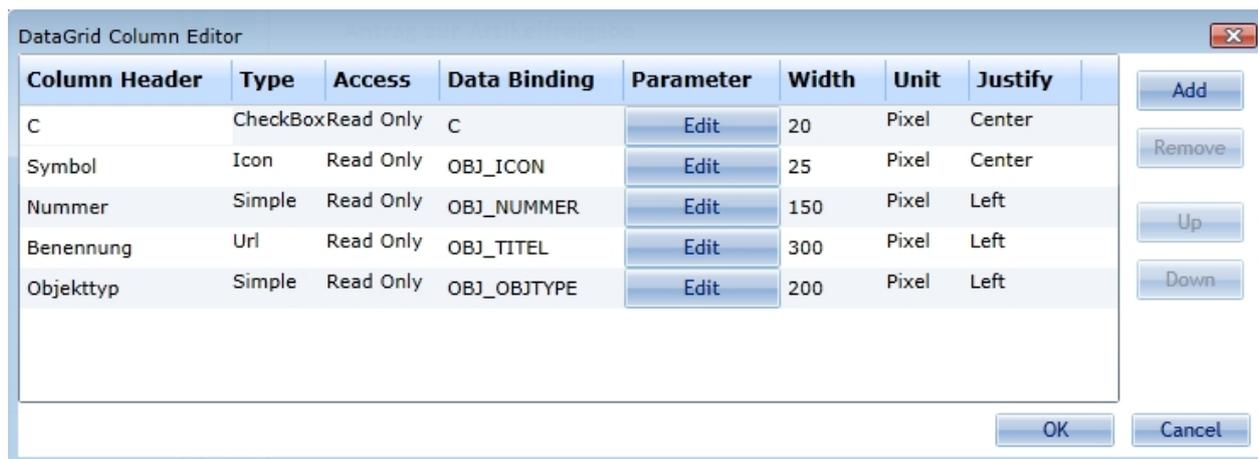
**Variable** The name of the variable, where the Data is taken from

## Workflow Parameters

- Grid Action** You may specify an expression to be executed when an Action button is / Action buttons are pressed
- On Drop** You may specify an expression to be executed when you insert content per drag and drop

## The Data Grid Column Editor

The Data Grid Column Editor is used, to specify the columns in the Data Grid.



## Column Types

IYOPRO provides the following column types:

- **Simple**  
A simple column simply shows a text value
- **Url**  
An URL column shows a text value as hyper-link. The underlying hyper-link must be specified in the parameter.  
Sample: `http://OpenEDM.local/OEWebService.asmx/OECallFunctionReturnTextFile?username=&password=&function=OBJ.ANYOBJ.WS.DOWNLOADLINKFILE&data={Binding OBJ_OBJJID}`  
The binding specified in the url may reference another column of the bound data table of the grid. If `{Binding Form:Var_Name}` is used another form variable can be referenced.
- **Icon**  
An Icon column shows a text value as icon. The underlying hyper-link must be specified in the parameter.  
Sample: `http://OpenEDM.local/OEWebService.asmx/OEGetIcon?iconname={Binding OBJ_ICON}`  
The binding specified in the url may reference another column of the bound data table of the grid. If `{Binding Form:Var_Name}` is used another form variable can be referenced.
- **CheckBox**  
A CheckBox column shows a bool value as checkbox.  
The underlying value will be 1=Checked, 0=Unchecked
- **ComboBox**  
A ComboBox column allows the user to choose a value from a given list
- **Function**  
A function shows a button with a predefined function. The function name must be specified in the parameter. The following functions are available:
  - DeleteRow  
This function deletes the row where the function has been executed
- **Action**

An Action shows a button which will execute a python script at the backend. This script is defined in Grid Action. The label of the button is the specified parameter or, if none is specified, the default "Action".

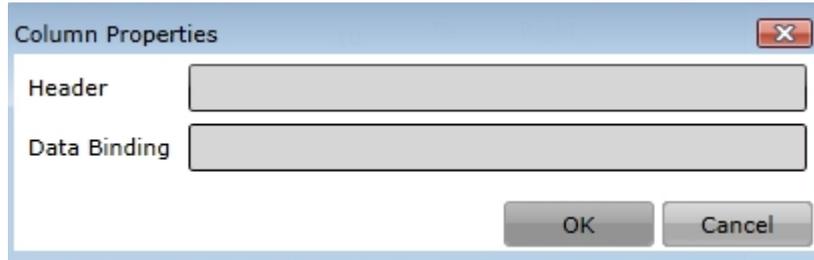
After pressing an action button, the dictionary of the line in which the button was pressed receives a new entry with "#ACTION" as key and the parameter as value. If no parameter was specified, the column name is set as value.

Sample: Get the row with "#ACTION" from a data grid with the variable "Data" and an Action button with the parameter "Details"

```
selected = filter(lambda row: '#ACTION' in row and row['#ACTION'] == 'Details', Form['Data'])
```

## Create new Column

To Create a Column press the Add Button. In the upcoming dialog specify:



Header: The header of the column shown in the Datagrid

Data Binding: The technical name of the column in the underlying value dictionary

### Delete Column

Select the column to delete and press the remove button

### Change Column Order

Select a column to move and press the Up / Down buttons to rearrange the columns

### Change Column Header

To change the header of a column double-click into the Column Header cell of the column and edit the value

### Change Column Type

To change the type of a column double-click into the Column Type cell of the column and choose the value

### Change Column Access

To change the access mode of a column double-click into the Column Access cell of the column and choose the value

### Change Column Data Binding

To change the Data Binding of a column double-click into the Data Binding cell of the column and edit the value

### Change Column Parameter

To change the Parameter of a column click the Edit button of the column and enter the value

### Change Column Width

To change the width of a column double-click into the Width cell of the column and edit the value

### Change Column Width Unit

To change the width unit of a column double-click into the Unit cell of the column and choose the value (A Data Grid only handles one unit. If you use pixel and percent in the same Data Grid every pixel value will be handled as a percent value)

### Change Column Justification

To change the justification of a column double-click into the Justify cell of the column and edit the value

## Use the tree mode - an example

This example, based on a family tree, shows how to display data as a child element in tree mode:

```
Tree variable = child
```

Without a child element:

```
form["grid"] = [{"Name": "Jack", "Date of birth": "16.07.1960"}, {"Name": "Paul", "Date of birth": "03.09.1963"}]
```

With a child element using the tree variable:

```
form["grid"] = [{"Name": "Jack", "Date of birth": "16.07.1960", "child": [{"Name": "Lily", "Date of birth": "02.04.1992"}]}, {"Name": "Paul", "Date of birth": "03.09.1963"}]
```

It would then look like this in the data grid:

Name	Date of birth
▼ Jack	16.07.1960
Lily	02.04.1992
Paul	03.09.1963

HTML Control

## HTML Control

An HTML Control allows you to write your own HTML source code or embed external content.

In order to add a HTML Control, drag the shape from the Form Toolbox onto the Form and drop it at the desired location.

**Create your bar graph live.**

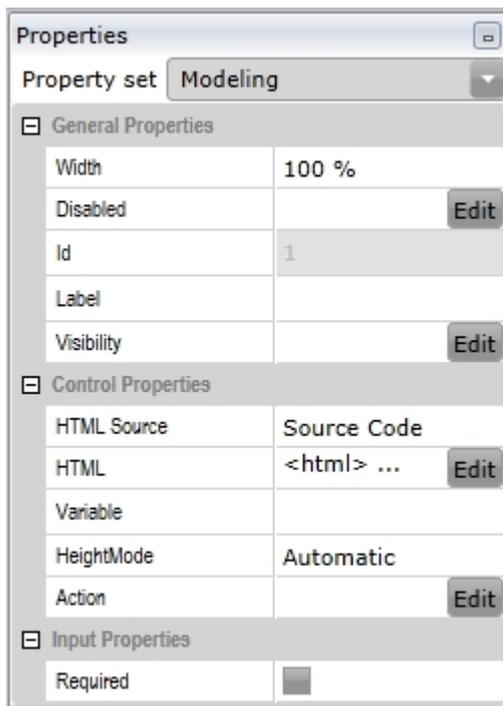
Value	Color	Tooltip	Delete
-------	-------	---------	--------

**Your bar graph:**

Sample HTML Control

## Properties

This Element utilizes beside the [General Element Properties](#) the following properties:



## Control Parameters

HTML Source	Specify which source you want to use for the HTML Control
HTML	You may specify an HTML source code
URL	The static URL, i.e. https://www.iyopro.com
Script File	Enter the link to the script file, e.g. simply type the script name if the script file is on the same level as the form
Configuration	Specify the key-value pairs which can then be read by the HTML Control
Variable	You may specify a variable name. The HTML Control can then handle the variable value. If you choose the Dynamic URL as the source, the value will contain a dynamic URL.
HeightMode	You may use the automatic mode, stretch mode or set a height in pixels. The stretch mode uses the rest of the available space
Height	Height of the control in pixels
Min Height	Specify the minimum height, that the HTML Control should have
Stretch Proportion	Specify the size ratio of the control to other controls with stretch mode
Action	You may specify an expression to be executed when the defined condition is met
Required	Specify, if the input is required or not.

A form may only be submitted if all required inputs have been made.

## HTML Source

- Source Code Design the HTML Control by defining your own HTML source code
- Static URL Embed content via a static URL, e.g. widgets, images, tutorials
- Dynamic URL You may specify a variable that allows you to dynamically set an URL to embed content, e.g. widgets, images, tutorials
- Script File Design the HTML Control by defining your own HTML source code in a script. This way, you could also reference it in other forms.

You can use the following in your HTML source code to interact with the IYOPRO form. Call them with "IYOPROForm."

## Members

 value: <code>any</code>	The value of the control variable
 <code>onValueChange(callback: (newValue?: <code>any</code>, oldValue?: <code>any</code>) =&gt; <code>void</code>)</code>	This method registers the value changed callback of the control's variable
 <code>getVariableName(): <code>string</code></code>	This method returns the variable name
 <code>executeAction(successCallback?: () =&gt; <code>void</code>, errorCallback?: (e: <code>any</code>) =&gt; <code>void</code>, alertError?: <code>boolean</code>): <code>void</code></code>	Executes the expression in the Action Property. You may specify what happens in case of a successful or failed execution and if there should be an error message
 <code>getFormValue(attribute: <code>string</code>): <code>any</code></code>	Retrieves the value of the given form variable
 <code>setFormValue(attribute: <code>string</code>, value: <code>any</code>): <code>void</code></code>	Stores a value in the given form variable
 <code>watchFormValue(attribute: <code>string</code>, callback: (newValue?: <code>any</code>, oldValue?: <code>any</code>) =&gt; <code>void</code>, deep?: <code>boolean</code>): <code>void</code></code>	This method registers the value changed callback of the given form variable deep: use this only if you want to monitor complex data structures or small changes. Attention: processing of the form may get slower
 <code>onVisibilityChanged(callback: (isVisible: <code>boolean</code>) =&gt; <code>void</code>): <code>void</code></code>	This method registers the visibility changed callback. The visibility change happens according to the conditions of the form property "Visibility"
 <code>onDisabledChanged(callback: (isDisabled: <code>boolean</code>) =&gt; <code>void</code>): <code>void</code></code>	This method registers the disabled changed callback. The deactivation change happens according to the conditions of the form property "Disabled"
 <code>onAllowSubmitChanged(callback: (allowSubmit: <code>boolean</code>) =&gt; <code>void</code>): <code>void</code></code>	This method registers changes to the condition of the control that prevents the submitting of the form. The property "Required" has to be checked
 <code>onDestroy(callback: () =&gt; <code>void</code>): <code>void</code></code>	Specify what is supposed to happen once the control no longer exists, for example closing connections to external systems
 <code>setRequiredCheck(allowSubmit: (value: <code>any</code>) =&gt; <code>boolean</code>): <code>void</code></code>	Specify the condition that has to be met in order to submit the form. The property "Required" has to be checked

⇒ <code>setRequiredErrorMessage(message: string): void</code>	Specify an error message for a required input. It will be displayed if the requirement is not met and the validation messages are visible
⇒ <code>getConfigValue(key: string, defaultValue?: string): string</code>	Retrieves the value of the given key from the configuration properties. You may specify a default value that will be returned if the given key doesn't exist
⇒ <code>loadRepositoryFile(id: number, successCallback?: (blob: Blob) =&gt; void, errorCallback?: (e: any) =&gt; void, alertError?: boolean): void</code>	Load a file from the repository. The file is determined based on its ID. The data is returned as a Blob.
⇒ <code>refresh(): void</code>	Use this method when changes occur within an object and IYOPRO does not recognize the changes on its own.

## Sample

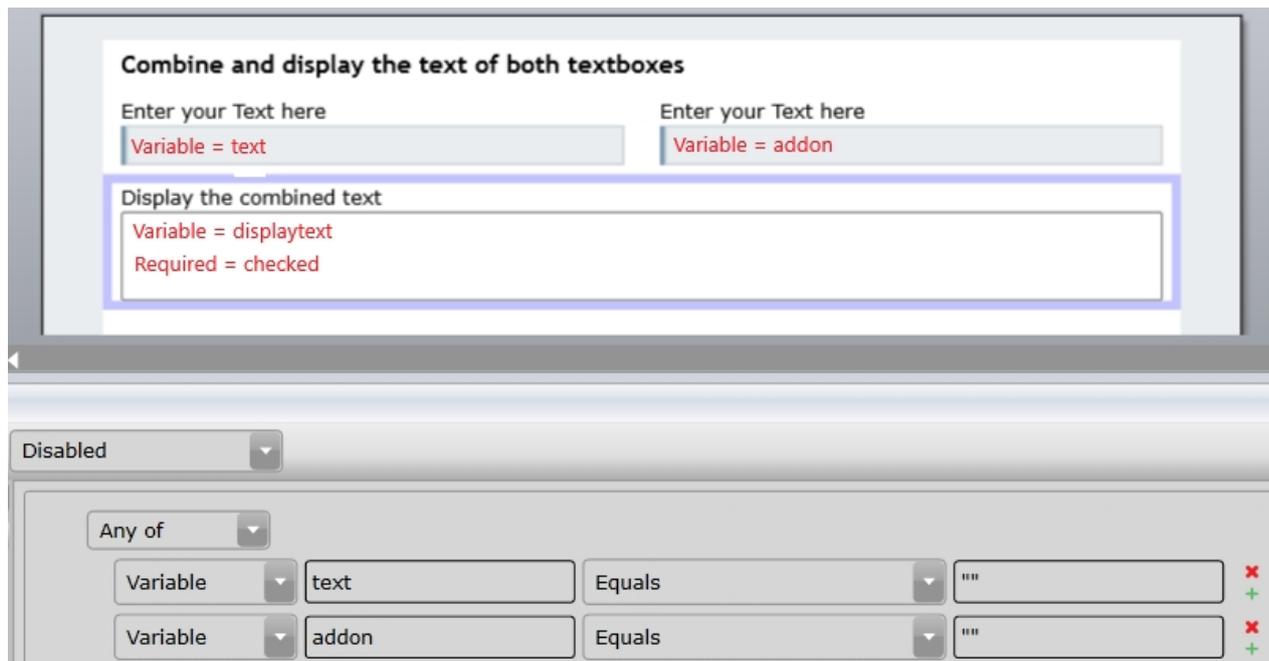
This sample sets the HTML Control variable via a time input control.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      input{
        width: 100%;
        margin: 4px 0;
      }
    </style>

    <script>
      function update(element) {
        IYOPROForm.value = element.value;
      }
    </script>
  </head>

  <body>
    <input type="time" id="time" onchange="update(this)">
  </body>
</html>
```

This sample gives you an idea of how some of the methods can be used.



```

<html>
  <head>
    <script>
      const TEXT_VARNAME = 'text';
      const ADDON_VARNAME = 'addon';

      var text = "";
      var addOn = "";

      function onLoad()
      {
        text = IYOPROForm.getFormValue(TEXT_VARNAME) || "";
        addOn = IYOPROForm.getFormValue(ADDON_VARNAME) || "";

        IYOPROForm.setFormValue(TEXT_VARNAME, text);
        IYOPROForm.setFormValue(ADDON_VARNAME, addOn);

        IYOPROForm.watchFormValue(TEXT_VARNAME, function (newVal) {
          newVal = newVal || "";

          if (newVal !== text)
          {
            text = newVal;
            update();
          }
        }, true);

        IYOPROForm.watchFormValue(ADDON_VARNAME, function (newVal) {
          newVal = newVal || "";

          if (newVal !== addOn)
          {
            addOn = newVal;
            update();
          }
        }, true);
      }
    </script>
  </head>
  <body>
    <div>
      <input type="text" value="Enter your Text here" />
      <input type="text" value="Enter your Text here" />
      <input type="text" value="Display the combined text" />
    </div>
  </body>
</html>

```

```

IYOPROForm.onDisabledChanged(function (isDisabled) {
    var paragraph = document.getElementById('display');

    if (isDisabled)
        paragraph.style.textDecoration = "line-through";
    else
        paragraph.style.textDecoration = "none";
});

IYOPROForm.setRequiredCheck(function () {
    if (text == "" || addOn == "")
        return false;

    return true;
});

IYOPROForm.setRequiredErrorMessage("Write something into both
textboxes.");

    update();
}

function update()
{
    IYOPROForm.value = text + " " + addOn;

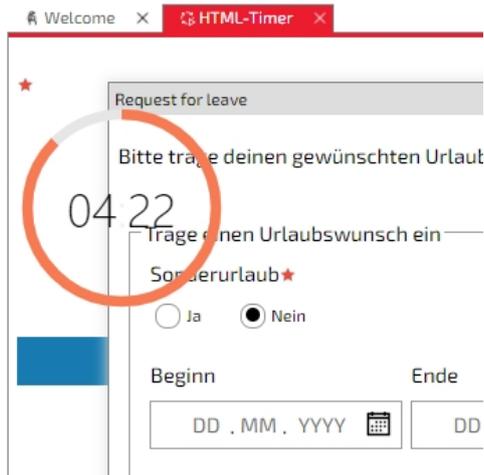
    var paragraph = document.getElementById('display');

    paragraph.innerText = IYOPROForm.value;
    paragraph.style.fontFamily = "'Balthazar', serif";
}
</script>
</head>
<body onload="onLoad()">
    <p id="display" style="margin-left:2em; margin-top:2em;
color:blue; font-size:300%">&nbsp;  </p>
</body>
</html>

```

### Note

We would like to point out a specific behavior regarding the HTML Control. For technical reasons, the iFrame is currently drawn over the actual content. This results in the HTML Control being displayed in front of a floating window (see fig.).



## QR-Code

### QR-Code

QR-Codes are currently only used on forms which are used as digital paper.

When these forms are printed out by the IYOPRO printer driver a QR-Code will be placed at the specified position.

With this unique QR-Code on each piece of paper it's possible to attach i.e. photos from a mobile phone to the digital form and pass all the data together to our workflow engine.

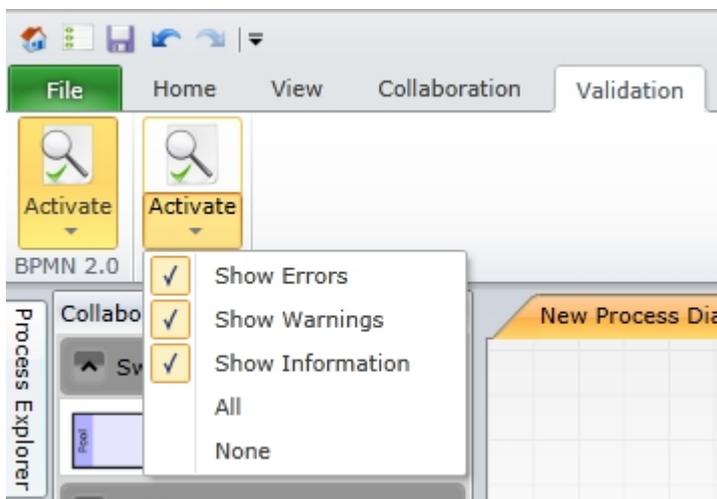
## Validation

IYOPRO provides a validation module which validates process models based on the following rule sets:

- BPMN 2.0 Validation Rules
- BPS Simulation Analysis Rules

To enable or disable the Validation navigate to the Validation Tab.

Validation distinguishes between Informations, Warnings and Errors. IYOPRO allows to individually adjust for both rule sets, which issues will be verified.



## Multi Language Diagrams

### Introduction

The international harmonization of business processes in companies is one of the main applications of business process management. Therefore it's absolutely necessary to be able to provide all results of BPM in multiple foreign languages.

IYOPRO provides an easy to use mechanism to archive this requirement.

### Conceptual Overview

The number and type of languages used in a diagram depend on various conditions which are often up to the organization behind. Therefore it's necessary to be able to define the required languages on document basis. This ensures, that you may share and exchange documents with other organizations without the need to do local customization. This results in the following definition

- All Languages  
With "All Languages" we name the collection of all predefined languages in IYOPRO.
- Document Languages  
"Document Languages" is the set of languages out of "All Languages", which have been assigned to a Document (i.e. a Diagram). This set may be changed at any time.
- Current Language  
The Current Language is the currently active language within a document. Each document has it's own Current Language. All changes to language dependent Attributes are done within the context of the Current Language.
- Default Language  
The Default Language is normally the initial language of the diagram. If you change the Current Language it might happen that several language dependent Attributes have not been translated yet. IYOPRO will then show the content of the Default Language, enclosed by ">>" resp. "<<". The Default Language may be set when [editing document languages](#)



Important notice:

There are several reasons to NOT show flags as a representation of a language because flags represent countries. However, most people love flags as a representation and many of you asked us to implement them.

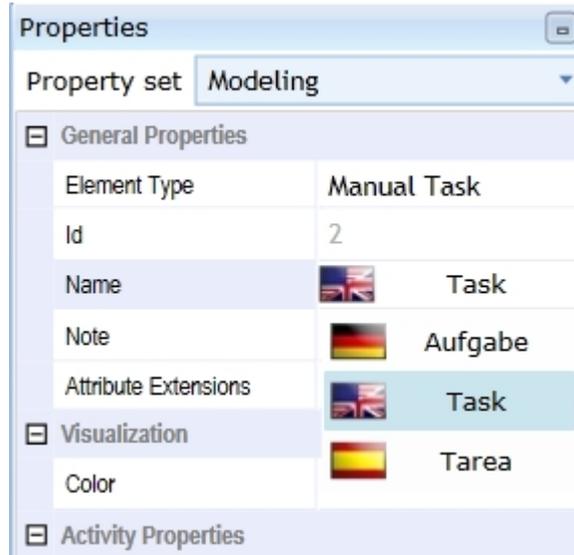
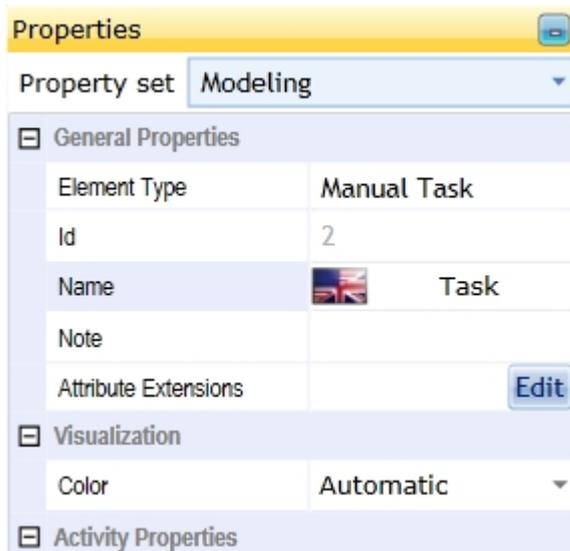
So it may happen, that you will not find your expected flag beside the language of your choice.

### Edit Multilanguage Term

#### Multilanguage Inline Edit

Most shapes have a name property that show a language flag left to the editable text.

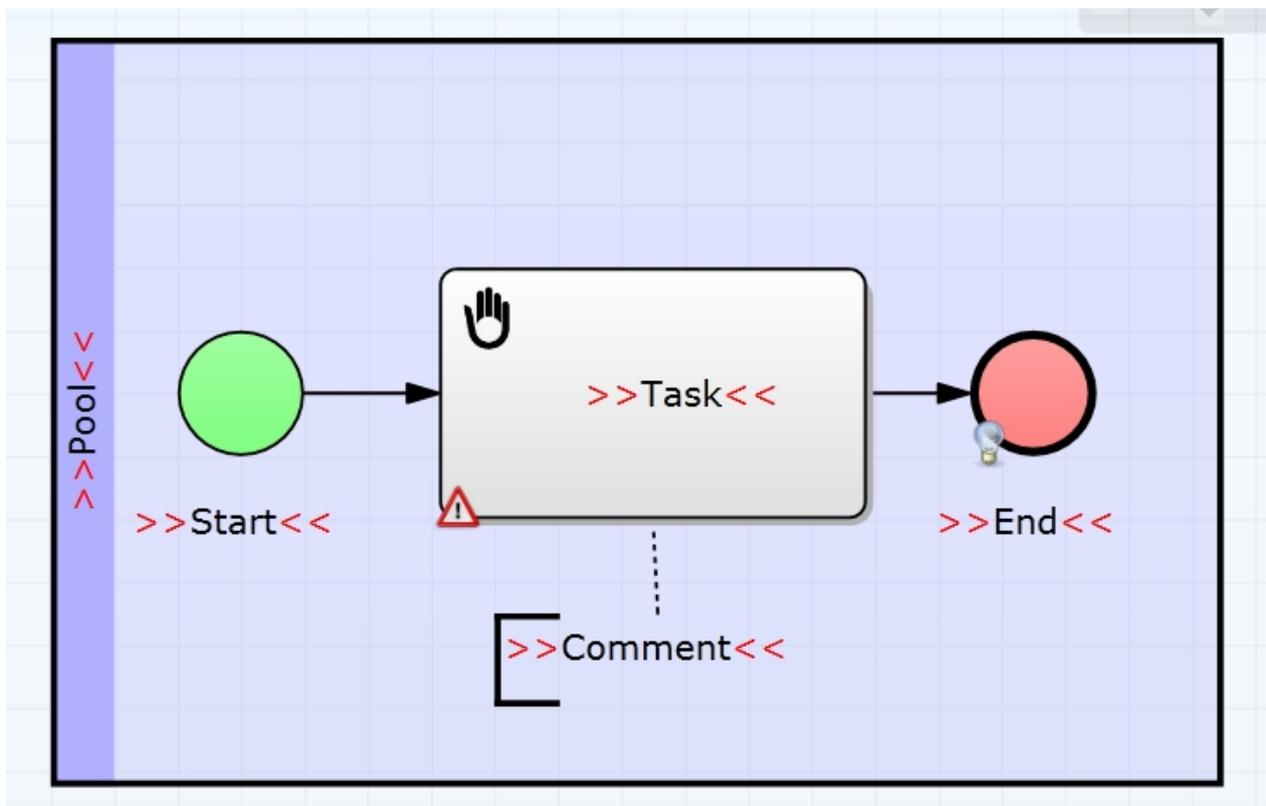
If the flag is selected a pop up appears which allows to inline edit the term in all defined document languages.



A language can also be selected from the pop up which [changes the current document language](#).

### Translate in diagram

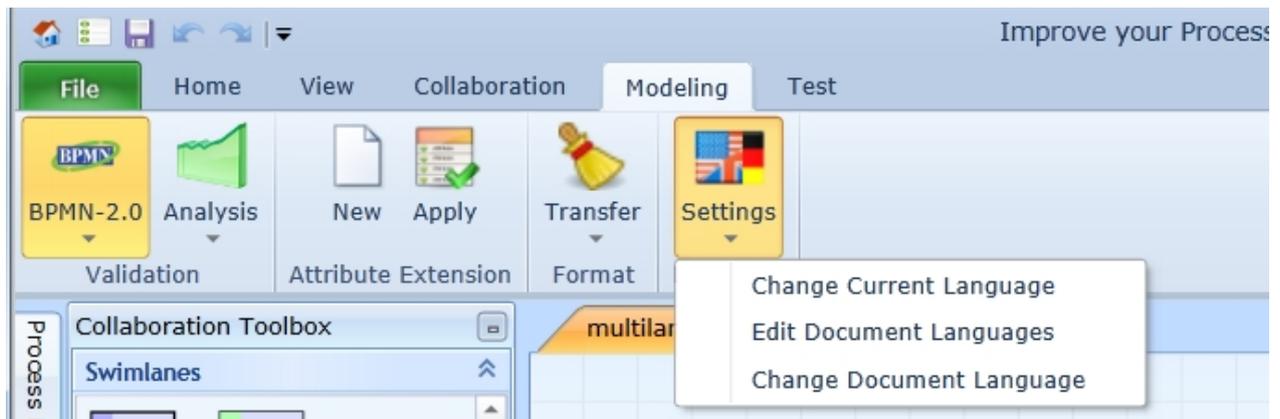
When the current language is changed all untranslated terms will be shown enclosed in red greater/smaller signs.



Tip: The [Find-Function](#) may be used to search for ">>" and find untranslated terms in large diagrams.

### Change Current Language

To Change the Current Language of the active document choose "Change Current Language" from the language settings button in the modeling ribbon tab.

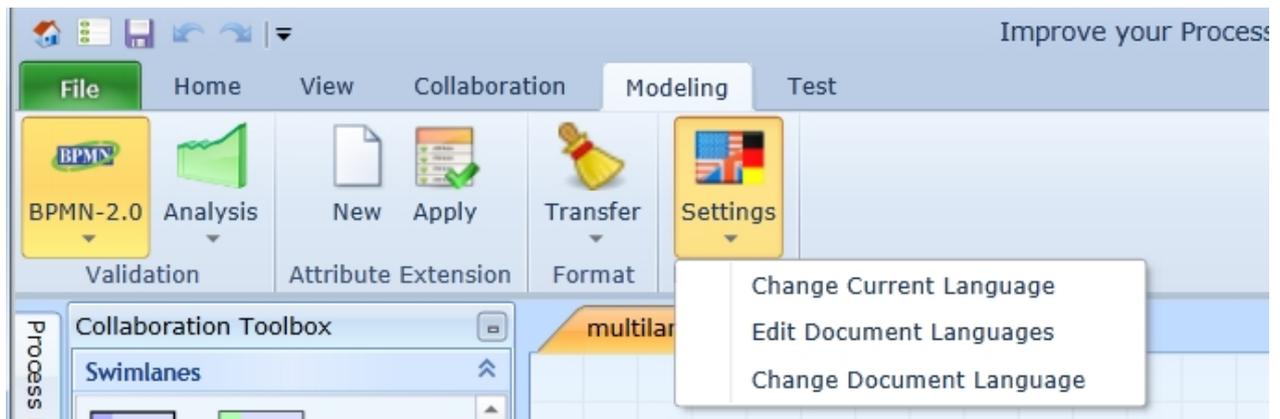


The Current Language will be highlighted. To change it select another language from the list of document languages and press OK.



### Edit Document Languages

To edit the assigned language list of the active document choose "Edit Document Languages" from the language settings button in the modeling ribbon tab.



The Edit Diagram Language Dialog provides the following options:

- Add languages
- Remove languages
- Make Default



### Add languages

To Add languages hit the Add Button. Then select one or more languages from the list of available languages and click OK.



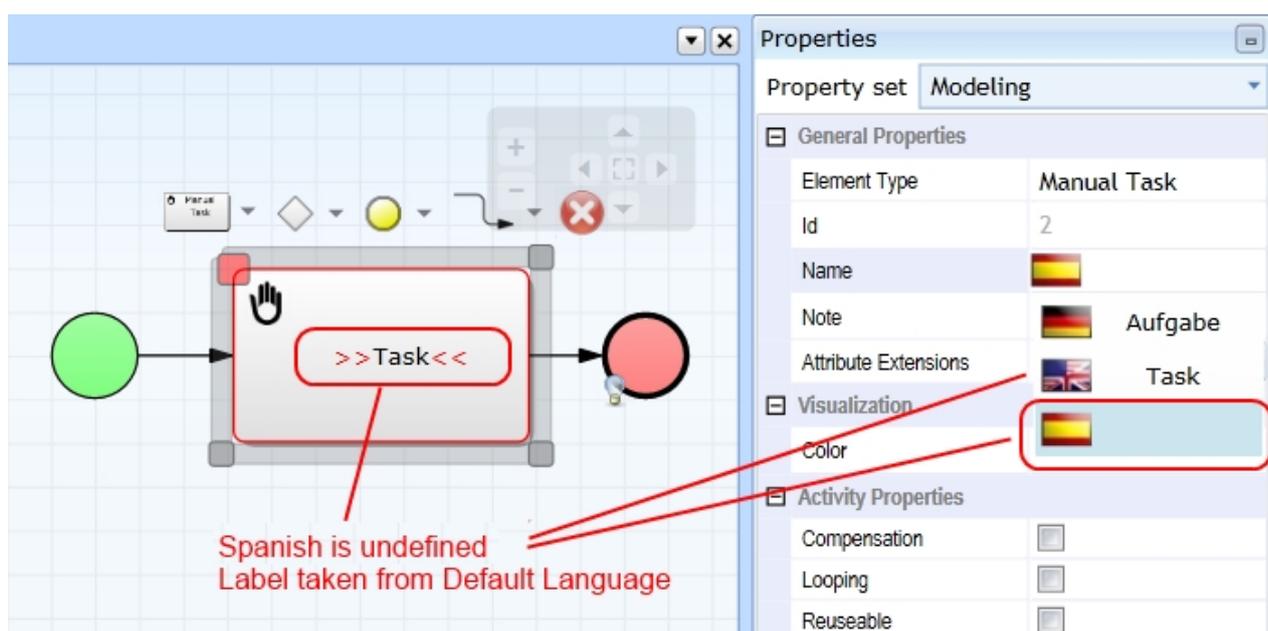
Adding new languages to the diagram language collection

### Remove languages

To remove languages select one or more languages from the list and hit the Remove Button. There must be at least one language remaining to be able to confirm the setting with OK.

### Make Default

The Default Language is the language, from which the visualization is taken if a shape is not translated in the Current Language.

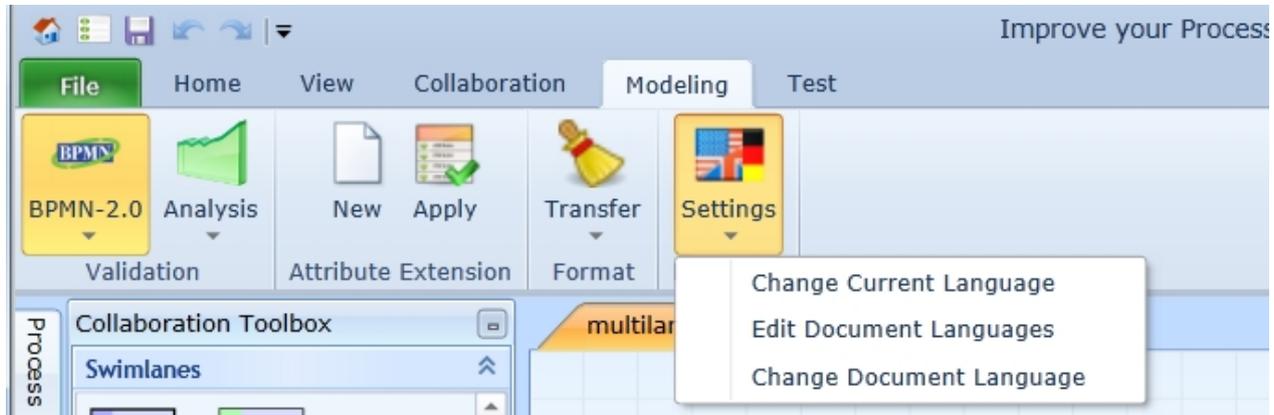


In this example English is the Default Language. Spanish is the Current Language. Because the term has

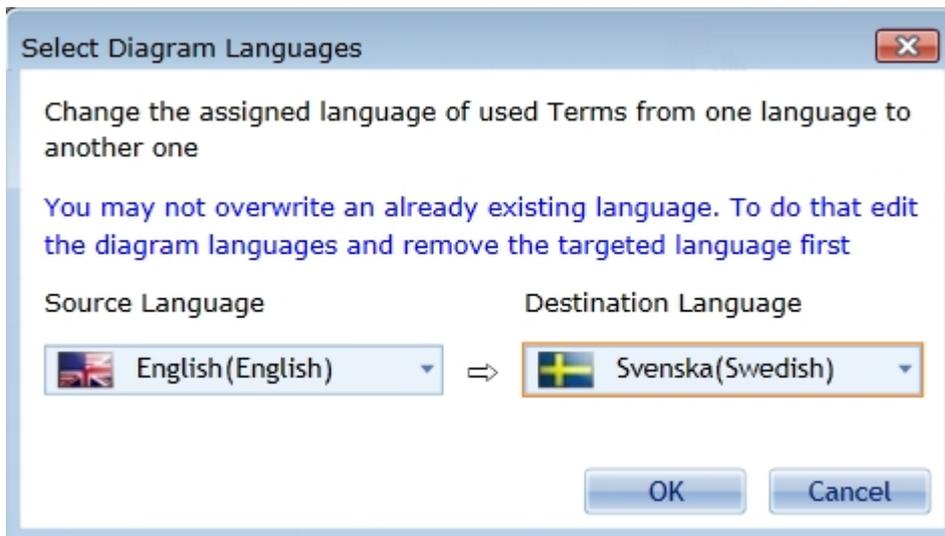
not been translated into Spanish, the English one is shown, but enclosed in ">>" resp. "<<".

## Change Document Language

With this operation it's possible, to change the language of associated terms from one to another language. To Change a language of the active document choose "Change Document Language" from the language settings button in the modeling ribbon tab.



In the upcoming Dialog choose which diagram language should be changed to which new language. Hit OK to proceed.



## The Property Concept

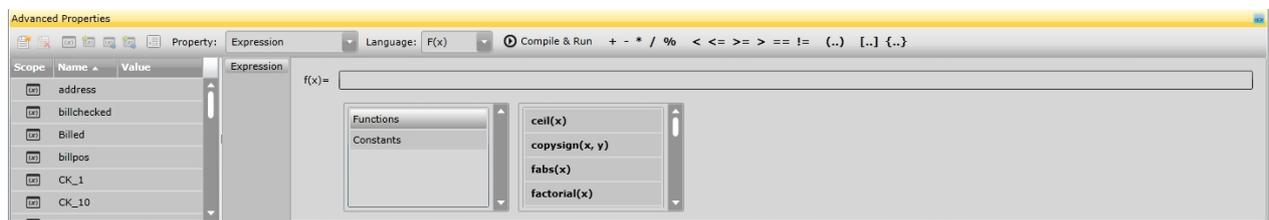
### The Property Concept

Every element in IYOPRO provides an amount of properties which control the elements characteristics. The properties are usually shown in the [Properties Area](#) within the Properties Grid.

Properties	
Property set: Modelling	
General Properties	
Element Type	Schema Table
Id	2
Name	Customer
Note	
Workflow Properties	
Variable Name	
Column Properties	
Size	0
Data Type	String
Name	Zip
Primary Key	<input type="checkbox"/>
Column Workflow Properties	
DataSet Key	<input type="checkbox"/>
Variable Name	

Sample Property Grid

Most properties are very simple and can be easily edited. But there are also some properties which require more input. These properties are called "Advanced Properties" and can be edited in the [Advanced Properties Window](#).



Sample Advanced Property

The following Plug-Ins for the Advanced Properties Window are available:

- [Expression Editor](#) - This is used whenever a Workflow Expression can be specified
- [Condition Editor](#) - This is used whenever a Workflow Condition can be specified
- [Feedback Editor](#) - This is used whenever a Workflow Feedback can be defined
- [Message Editor](#) - This is used whenever a Workflow Message can be defined
- [Message Map Editor](#) - This is used whenever a Workflow Message Map can be defined
- [Web Service Editor](#) - This is used whenever a Workflow Web Service can be defined

## Expression Editor

### The Expression Editor

Use the Expression Editor to specify an Expression within a Workflow. As an example you may define a Script Expression if you drag a Script Task onto the Workspace and select the Expression Property.

Expressions can be easy formulas as well as complex program code. To archive these different requirements IYOPRO provides a language selector to choose the best expression type.

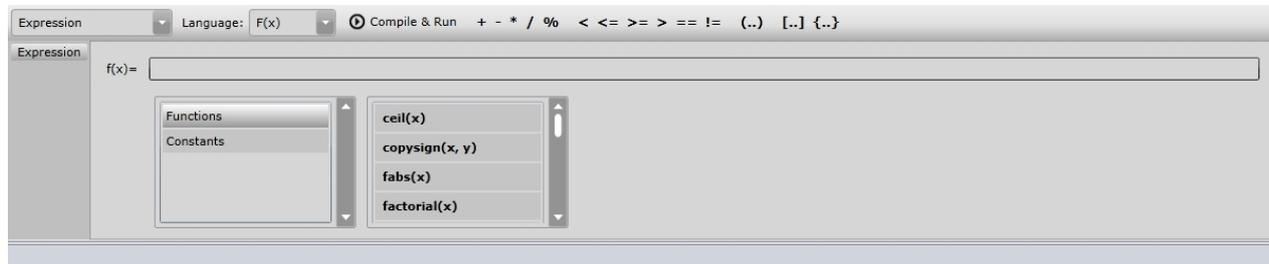
## Language

Currently IYOPRO provides the following languages:

F(x) - The expression consists out of a simple formula

Python - The Expression is based on Python Program code

### F(x) Formula Expressions



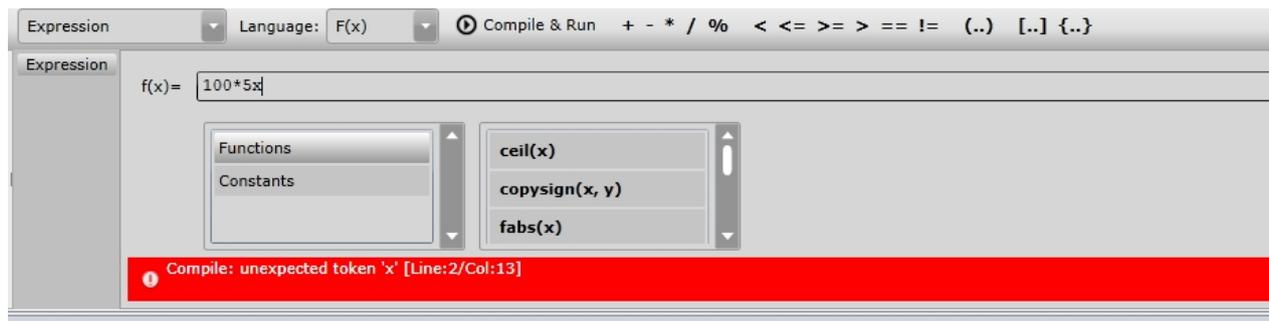
If you choose F(x) formula expressions you may enter a simple formula into the text box.

As an Example enter

100\*5

and press "Compile & Run". IYOPRO will open an output window with the result 500 shown.

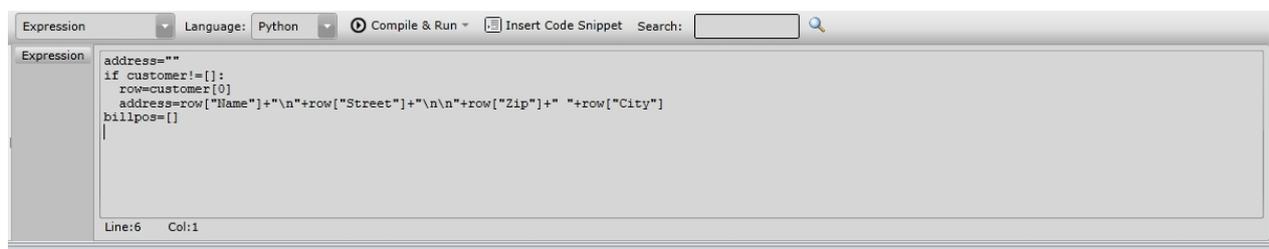
If your Formula contains errors, the translation process will fail and an error will be shown:



You may safely ignore the "Line 2" position of the error message. This is because IYOPRO will generate a python script out of your formula which inserts your formula at line 2.

There are many functions and constants available to let you specify more complex formulas. Please refer to the [Python Math Documentation](#) to get a detailed description of all available functions.

### Python Script Expressions



If you choose Python script expressions you may enter your program code into the text box.

As an Example enter

```
print 100*5
```

and press "Compile & Run". IYOPRO will open an output window with the result 500 shown.

If your Formula contains errors, the translation process will fail and an error will be shown as described in the f(x) section.

Please beware, that Workflow Expression will be executed on the Server where as others may be executed on your local PC.

If you do not specify anything else, the Compile & Run button will execute the python script on your local PC.

Both environments are sandboxed, what means that your scripts run in a controlled environment and cannot damage anything.

But keep in mind, that the server sandbox and the client sandbox have different security levels.

Where the server sandbox allows access to web services, the client sandbox doesn't. Therefore if you may want to test python scripts

for workflows you should test them also using the "Compile & Run (on server)" method.

You can access this method through the small arrow right to the Compile & Run Button.

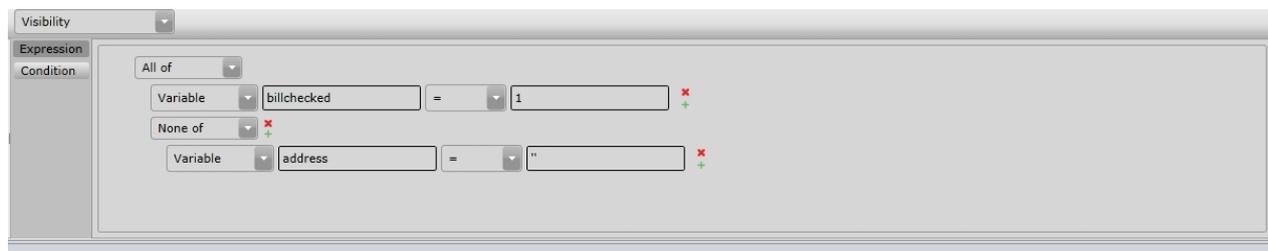
Please refer to the [Python documentation](#) for more information

## Condition Editor

### The Condition Editor

Use the Condition Editor to specify a Condition within a Workflow.

As an example you may define a Visibility Condition if you drag a Form Control onto a Form and select the Visibility Property.



Conditions are rules, which may be nested. They always return either True or False.

Each row begins with a Selector with the following options

- All Of  
This will create a sub rule. If all sub rules return True, the overall result will also be True, else False.
- Any of  
This will create a sub rule. If any sub rule returns True, the overall result will also be True, else False.
- None Of  
This will create a sub rule. If any sub rule returns True, the overall result will be False, else True.
- Variable  
This will add two operand fields divided by an operator selector. You may enter Workflow Variables into the operator fields as well as constants.  
If this row belongs to a sub rule, there are +/- buttons at the end of the row which let you remove the row or add another row.

## Feedback Editor

### The Feedback Editor

A Feedback is used, if an user action proceeds the execution of a workflow whereas the users activity has

been done.

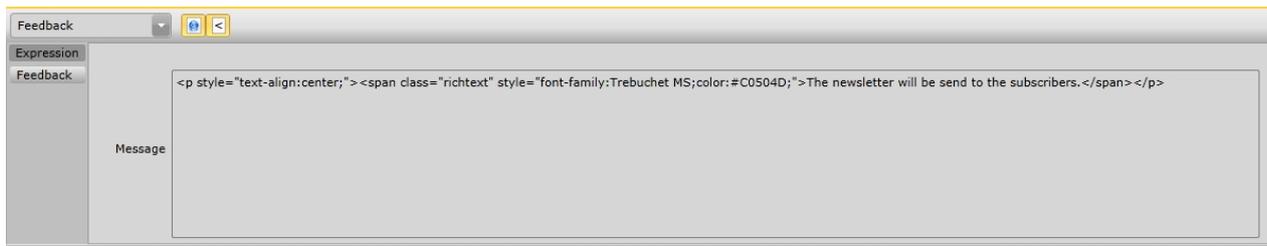
Examples are:

- Submit a web form - Receive a "Processing your Input..." feedback
- Click onto an email link - Receive a "Thank you" feedback



The feedback consists out of a simple text box or a rich text box. You may toggle this with the  Button in the Toolbar.

In case of rich text you may also specify to enter plain HTML code. To do this toggle the  Button.



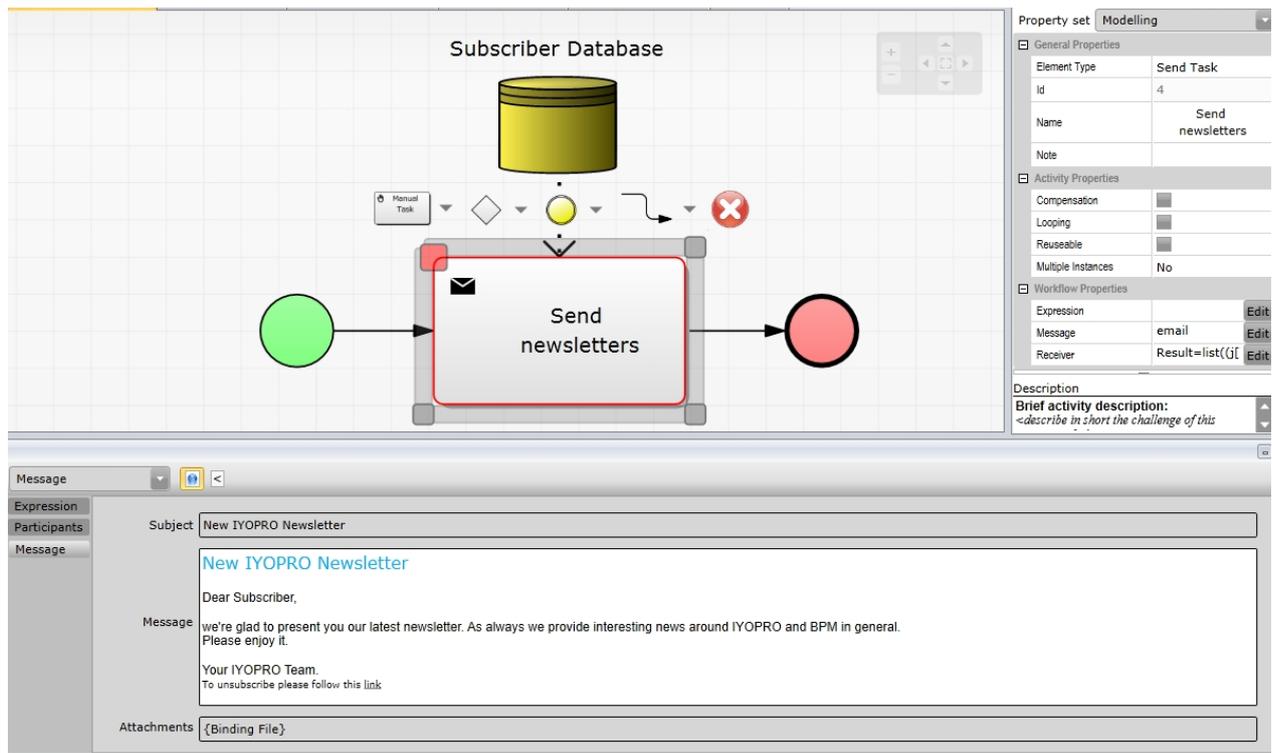
## Message Editor

### The Message Editor

A message is used to inform participants about process states.

Examples are:

- A User Task must be completed
- A Send Task must send a message



The message consists out of a subject, a message body and optional attachments.

The Subject specifies the headline of the message.

The message body can be entered into a simple text box or a rich text box. You may toggle this with the  Button in the toolbar.

In case of rich text you may also specify to enter plain HTML code. To do this toggle the  Button.

Optional you may add attachments. They must be assigned to the workflow instance and can be uploaded for instance by a form.

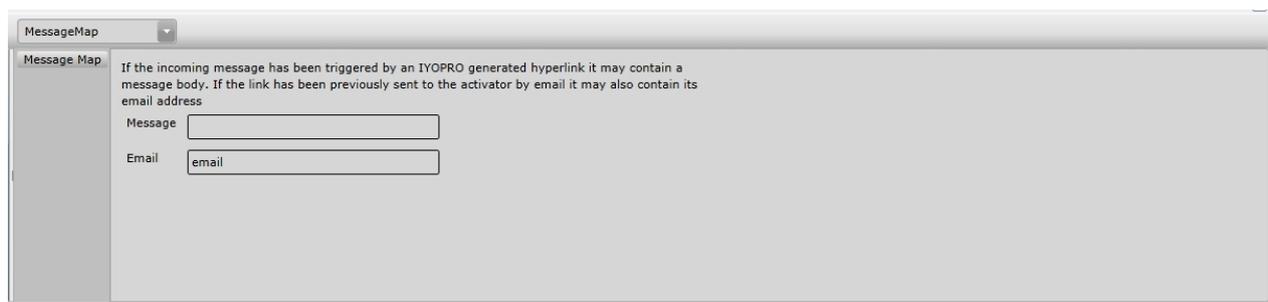
## Message Map Editor

### The Message Map Editor

A message map is used to map message data into process instance attributes.

Examples are:

- A Receive Message Start Event is triggered with incoming data



### Input Parameters

Message      Enter the process attribute, to which

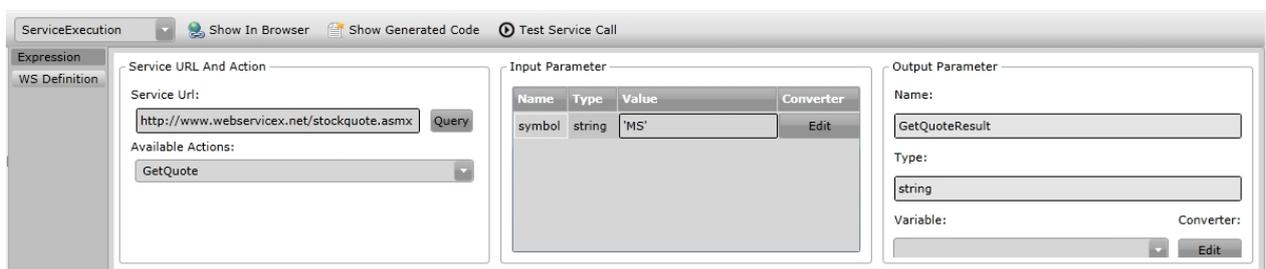
the message will be saved.

Email If you previously send a workflow message hyperlink within an email message and the receiver activates the hyperlink, IYOPRO encodes the senders email-address which can be stored into the attribute name here specified.

## Web Service Editor

### The Web Service Editor

A Web Service is used to communicate to another device over the Web (Internet)



To define a web service the following steps are needed:

#### Service Url and Action Section

1. Enter the service URL into the appropriate text field and hit Query
2. Choose one of the available actions, the web service provides

#### Input Parameter Section

1. Assign to each input parameter either a constant value or a workflow variable.  
A constant value may be 1, 1.5, 'stringvalue'. A variable may be a single word i.e. MyVar.
2. If necessary edit a value converter. This is useful, if the data in your variable must be transformed

You may now hit the "Test Service Call" Button in the toolbar to view the returned value

#### Output Parameter

1. Assign a workflow variable to the web service output parameter (ie. the result).
2. If necessary edit a value converter. This is useful, if the data of the result must be transformed

#### Show in Browser

If you hit the "Show in Browser" button IYOPRO will open a new browser window with the URL specified in the text box "Service Url".

#### Show Generated Code

IYOPRO translates the web service definition into a Python script. You may view this script if you hit the "Show Generated Code" button

## Participant Editor

### The Participant Editor

Use the Participant Editor to assign participants to activities.



The participant list consists out of the following columns:

Participant Name	The name of the participant. This will be automatically filled if you add a participant
Responsibility	<p>The way how the participant is responsible at the activity:</p> <ul style="list-style-type: none"> <li>● Responsible Those who will perform the activity</li> <li>● Accountable Those who will approve the task</li> <li>● Consulted Those who will give hints to archive the task</li> <li>● Informed Those who are kept up to date</li> <li>● Supportive Those who may help to complete the task.</li> <li>● Verify Those who check whether the activity result meets the specification</li> <li>● Sign Off Those who approve the verify decision</li> <li>● Omitted Those who are excluded from participating at the task</li> </ul>
Amount	This parameter is for simulation purposes only. You can specify, how many Resources of this type (i.e. a Role) is required to perform the task.

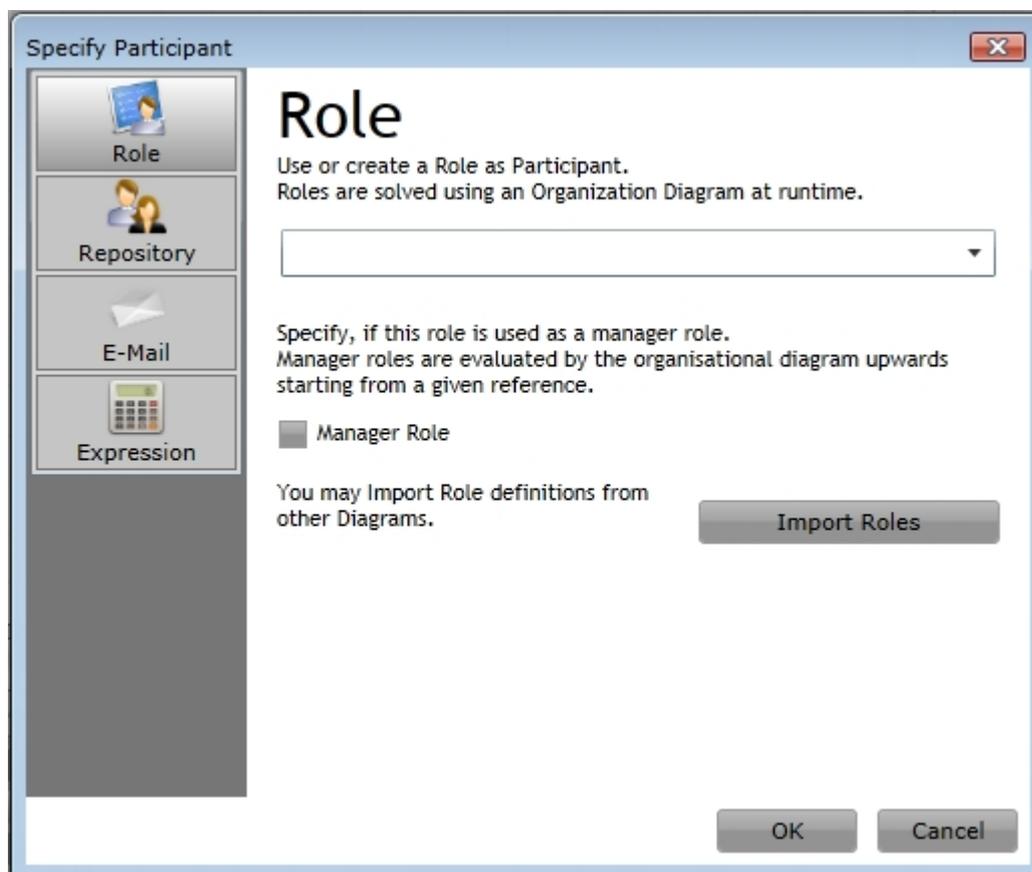
### Add Participant

To add a participant click on the add button. The "Specify Participant" Dialog appears.

The following options appear:

## Role

Use this to add a role as participant. Roles will be solved on workflow runtime using an organizational diagram.



There are two different methods to solve roles within an organizational diagram:

### Manager Roles:

If a role is interpreted as manager role IYOPRO will traverse up in the organizational diagram starting at a reference until an element is found that matches the role. This will result in exactly one result or, if nobody matches, in an empty result. You need to specify how the reference is taken:

- **Process Initiator**  
This is the user which initiated the process instance. Note that this may be unset if the process instance has been started by an anonymous form / message trigger ....
- **Latest Task Performer**  
This is the user which executed the task before the current task. Note that this may be unspecified if there has no user task been executed before.
- **Process Attribute**  
The reference is taken from a process attribute. You need to specify the process attribute name in the text field that appears if you choose this option. The attribute value may either contain a repository user id or better an email address. IYOPRO will search the organizational diagram for an element with the given property and use this as reference.

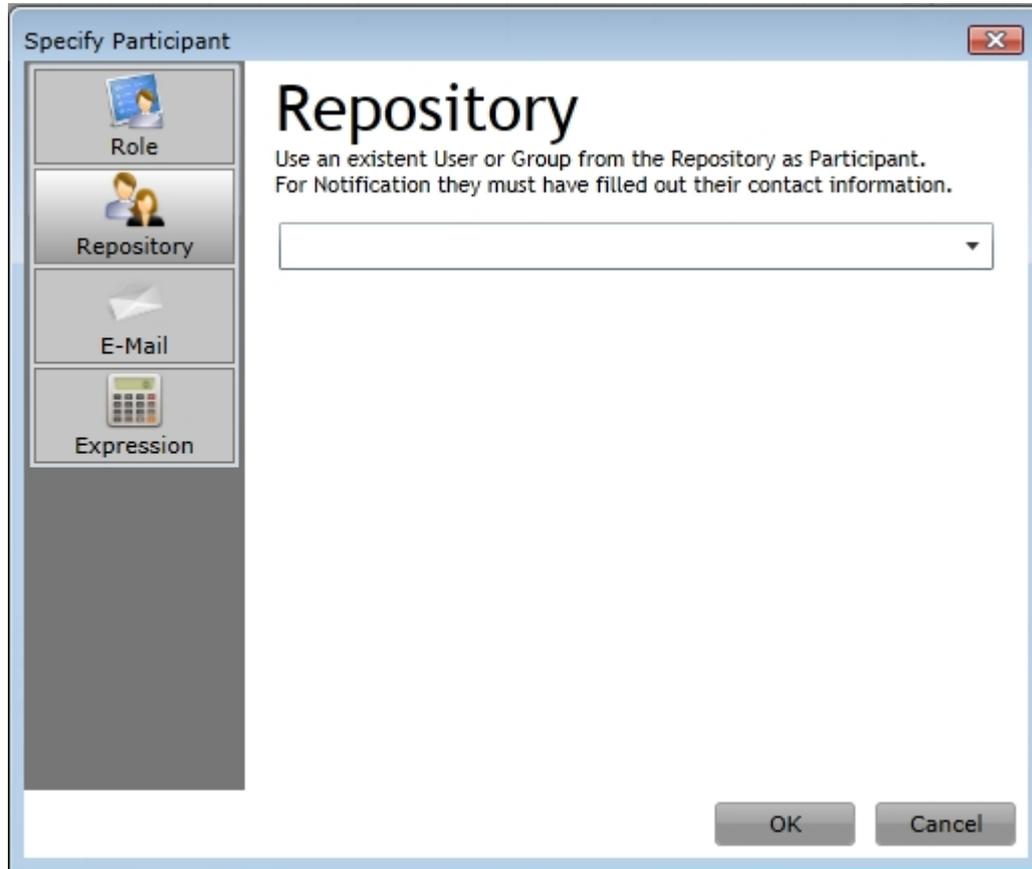
### Default:

The default method is that all elements, which own the role will be selected in the result list. If nobody owns the role the result will be empty.

If you have already defined roles in another diagram which is not part of your active BPI project you may import these roles by a click on the import roles button.

## Repository

Use this to add a repository user or group as participant.

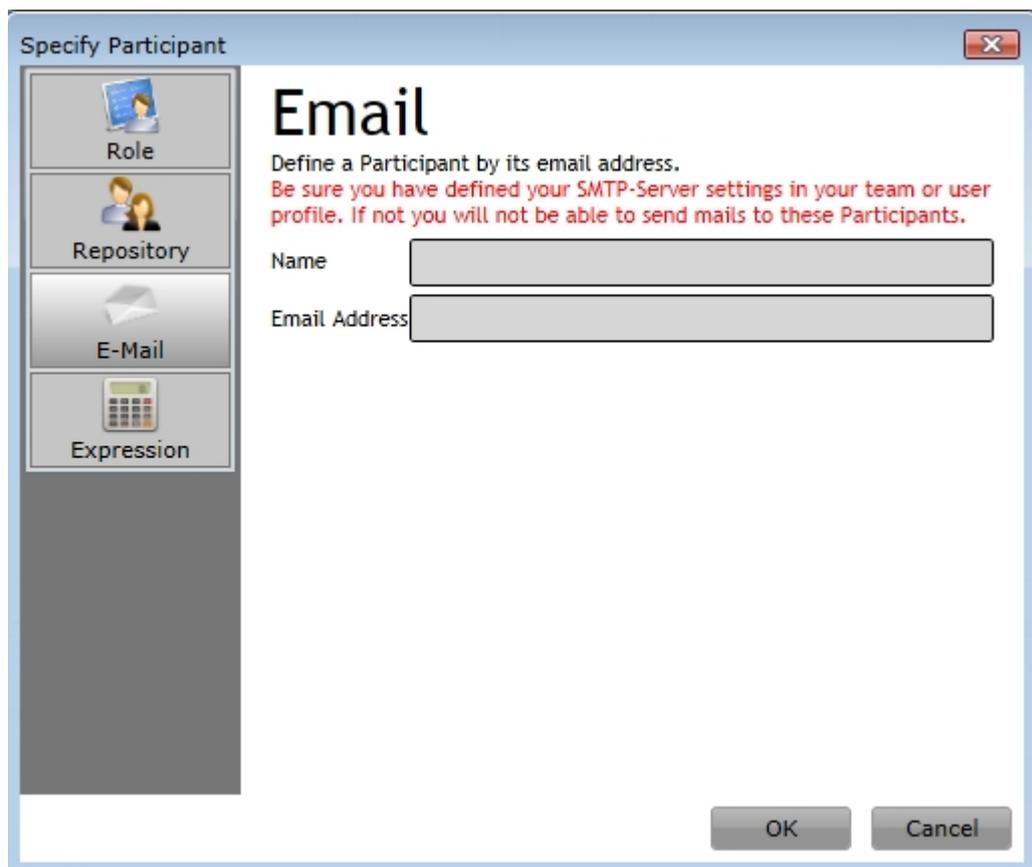


The image shows a dialog box titled "Specify Participant" with a close button (X) in the top right corner. On the left side, there is a vertical list of four options, each with an icon and a label: "Role" (person icon), "Repository" (two people icon), "E-Mail" (envelope icon), and "Expression" (calculator icon). The "Repository" option is currently selected. The main area of the dialog box has the heading "Repository" and the text "Use an existent User or Group from the Repository as Participant. For Notification they must have filled out their contact information." Below this text is a dropdown menu. At the bottom right of the dialog box are two buttons: "OK" and "Cancel".

If you choose a group all members of the group will be assigned the result set.

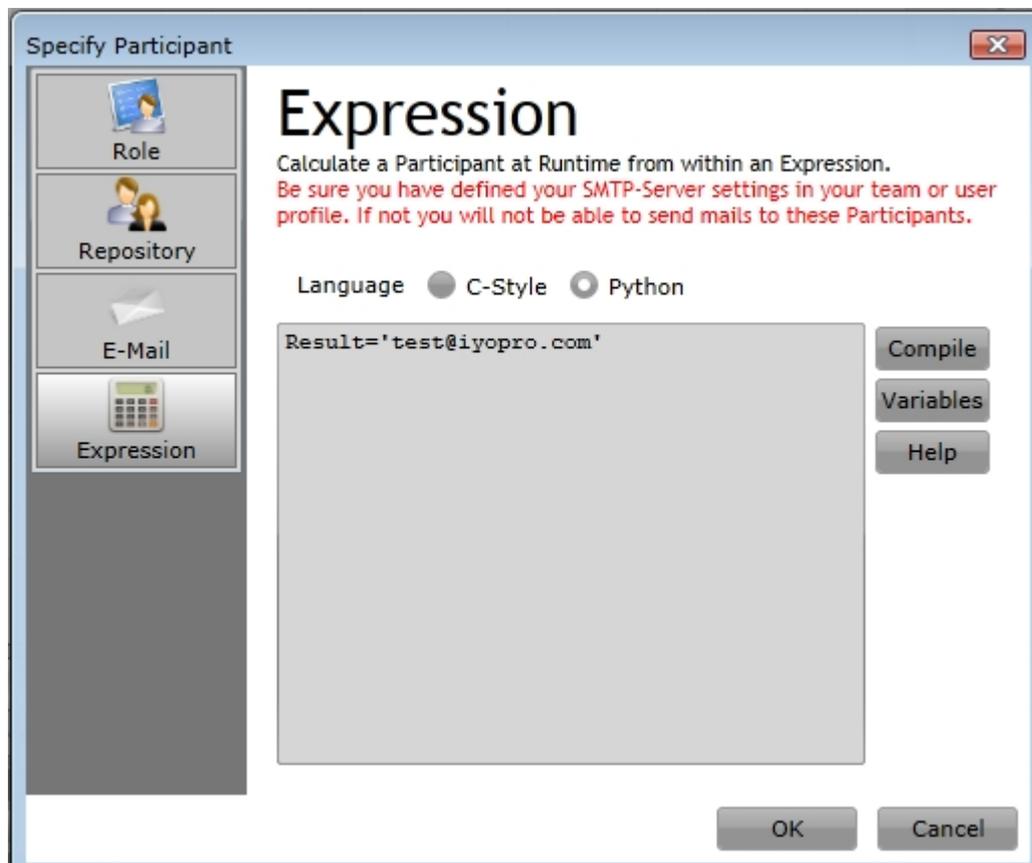
## Email

Sometimes it's required to assign external users as workflow participants. Use this to specify an external participant by its name / email address.



### Expression

If you need to calculate the participant by an expression this option is the right one.



The Result variable of an expression may contain the following content:

single email address	Result='test@iyopro.com'	The Result will contain a single string value which represents the email address of the participant.
list of email addresses	Result=[ <code>'test1@iyopro.com'</code> , <code>'test2@iyopro.com'</code> ]	The Result will contain a list of email addresses
single user record	Result=Workflow.Users["Smith"]	The Result will contain a single <a href="#">user record</a>
list of user records	Result=[Workflow.Users["Smith"],Workflow.Users["Jones"]]	The Result will contain a list of user records

## Remove Participant

To remove a participant either click on the remove button at the end of the participants row or select one or more rows and click on the Remove button in the toolbar.

## Stakeholder related Diagram Views

### Introduction

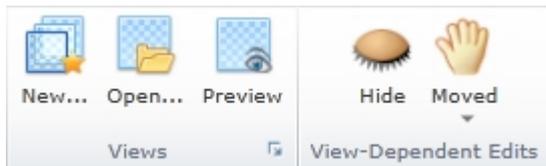
Business Process Diagrams often contain lots of information which are not always intended for every user. Sometimes it's also a must to hide information from certain people. IYOPRO supports this requirement with it's Stakeholder related View feature, the possibility of view dependent edits and the opportunity to specify privileges on top of these views.

### Conceptual Overview

IYOPRO supports an unlimited number of Views within a diagram. Views provide the following features:

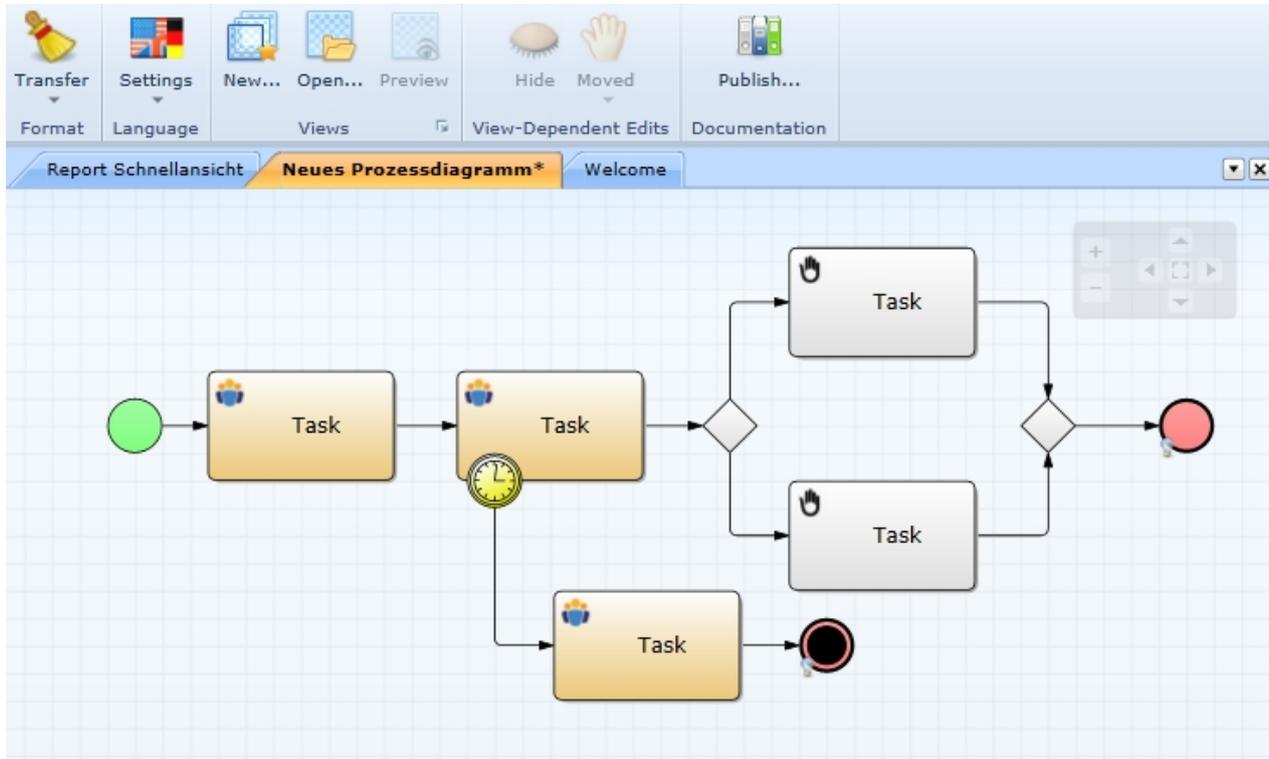
- View creation based on the original diagram, as copy of an existing view or from a list of preselected Elements
- Hide Elements  
Use this to hide Elements which should not be visible on the view
- Add Elements  
Use this to add Elements to the view. A typical use case is to remove multiple activities and replace them with a single, simplified activity.
- Move Elements  
Use this to move Elements view dependent to a new location
- View dependent properties  
Some Elements support view dependent properties. You may for example toggle the black-box property of a pool view dependent to hide the content in that view.
- Preview  
View dependent edits are highlighted within the view editor. use the preview function to see the view in it's finished state.
- Permissions  
Specify for views as well as for the original diagram who will be authorized to read/subscribe/comment the view.

The View operations can be found in the Modeling tab of the [Ribbon](#).

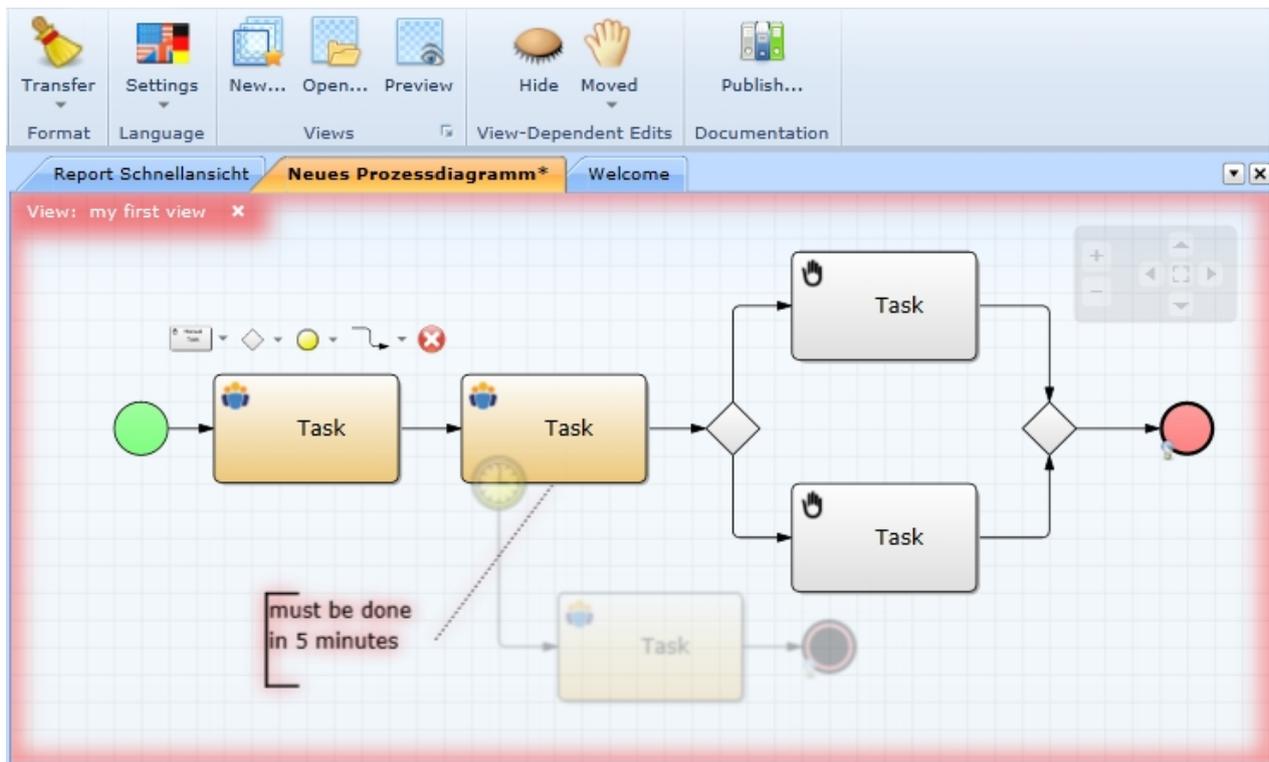


## Example

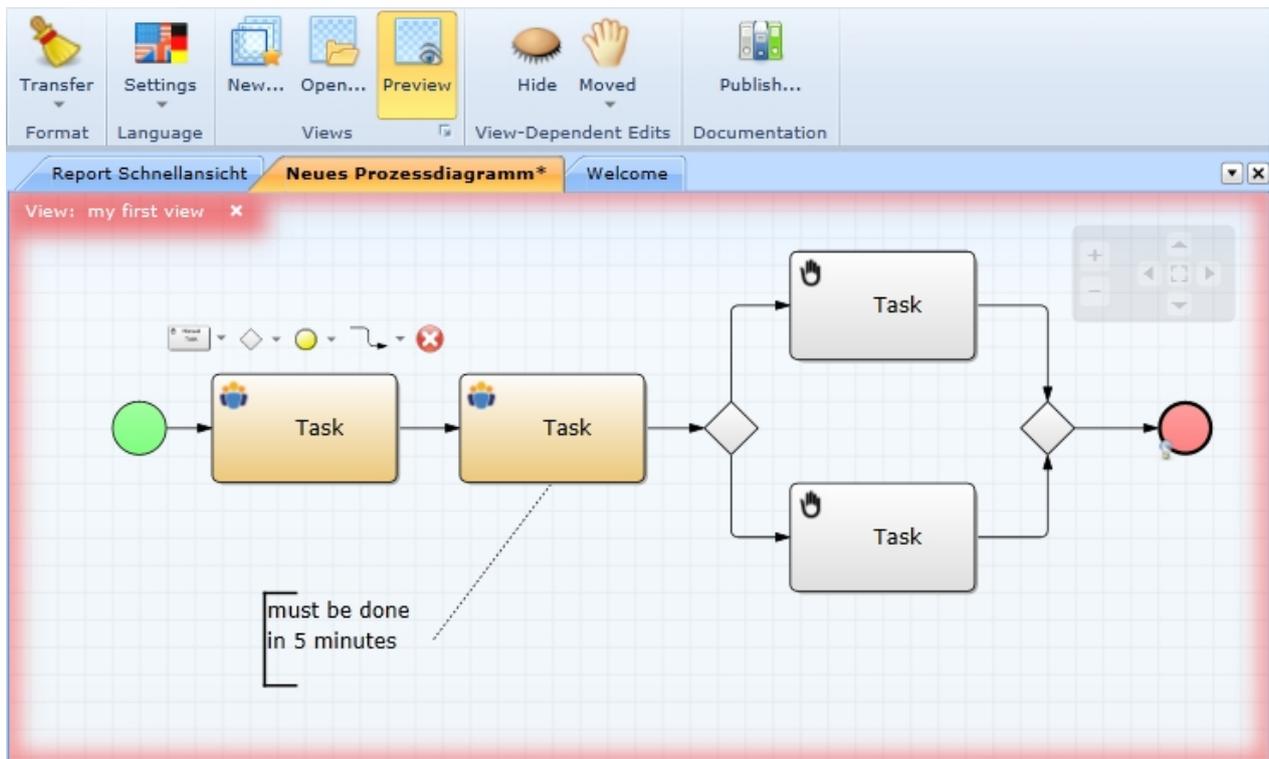
This is a sample process diagram which will be enhanced by a view. The aim is to hide the timer exception and show the happy path only



Without preview enabled View dependent edits are visualized within the editor by transparency and red blurred boundaries.

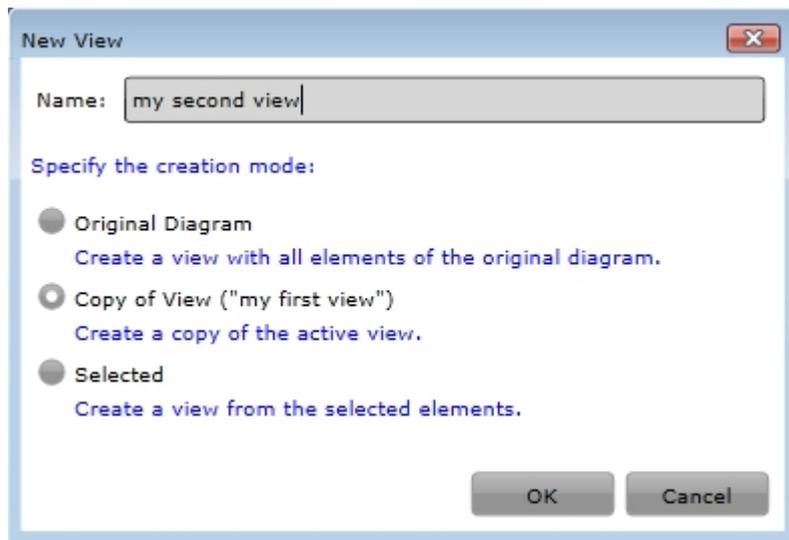


If you toggle the Preview Button from the View ribbon group of the modeling ribbon tab IYOPRO will show the view in final state.



## Create View

To create a View go to the Modeling tab at the Ribbon. Click on "New" in the Views Ribbon group. Dependent on your current environment you'll receive the following dialog:



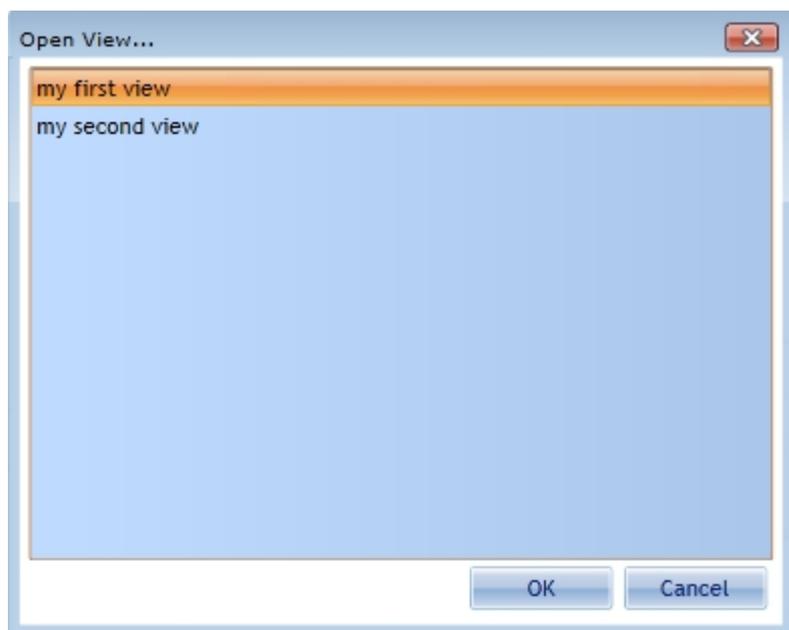
First enter a unique name for the view. Then specify the creation mode:

- Original Diagram  
Create a view which contains all Elements of the underlying original diagram. This option is always available
- Copy of View  
This option is only available if you try to create a view while you're currently editing a view.
- Selected  
If you choose this method all currently selected elements will be shown on the view. All other will be hidden. If you choose this option and do not have selected any object, the resulting view will only contain hidden elements which may later become unhidden.

## Open / Close View

### Open a view

To open a view click on the Open Button of the Views Ribbon group in the Modeling tab.

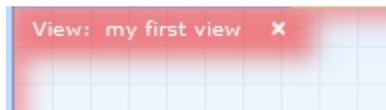


Select a view and click OK to activate the view.

This operation is also available in the [Views Toolbox](#) .

## Close a view

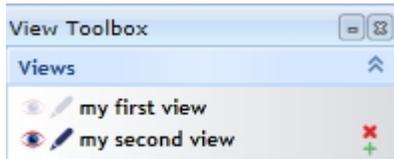
To close a view click the close button (x) in the upper left corner of the view



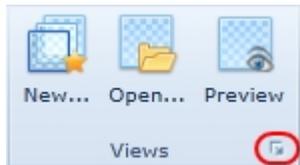
This operation is also available in the [Views Toolbox](#) .

## The Views Toolbox

The Views Toolbox is a small Toolbox which is shown in the [Toolbox Area](#) of IYOPRO.



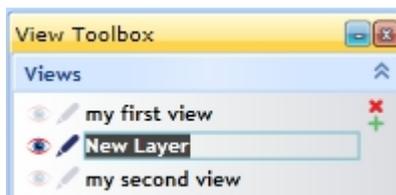
The Views Toolbox will always be opened if a new View is created or an existing view is opened. If you need to open the Toolbox manually click on the dialog launcher at the Views RibbonGroup.



The following operations are available:

## Create View

To create a new view move the mouse over the Toolbox entries. A small green + Sign appears. Click on it to create a new view.



Type in the Name, the view is activated automatically.

## Delete View

To delete a view move the mouse over the associated Toolbox entry. A small x appears. Click on it to delete the view.

## Rename View

to rename a view click on the views name and edit the value.

## Activate View

To activate a view click on one of the icons left of the views name

-  This method is used to only visualize the view. Edits will affect the original underlying model.
-  This method will fully activate the view. Edits will affect the view, not the underlying model

## View Dependent Edits

To perform view dependent edits you need to create a view first or open an existing one.

## Hide Element

Select the elements to hide and perform a delete operation as you would do it in a normal diagram. Hidden elements will be dimmed unless you have activated the preview button.

You may also toggle the Hide button of the View Dependent Edits ribbon group. It also indicates if an element is hidden or not.



## Unhide Element

Hidden elements can be restored if you repeat the hide operation used to hide the element.

## Move Element

Elements can be moved in the same manner as on the original diagram. Moved elements will be marked with a blurred border to indicate that the element has a view dependent edit.

## Unmove Element

To move an element to its original position select that element. The Moved button of the View Dependent Edits ribbon group will indicate that this element has been moved. Click on the button to restore the original position.



## Identify moved Elements

To identify all moved elements open the menu of the moved button and choose "Select Moved". This will select all moved elements.



## Edit Properties

Most properties of inherited elements of a view are disabled while view editing. However there are some properties, which can be overwritten.

On example is the Black Box property of a pool which allows you to view dependent expand or collapse a pool.

Another is the Permission property on diagram level which allows you to control the access restrictions for the diagram and for each view.

## Publishing and Deployment

### Introduction

With the IYOPRO Portal not only experienced but also occasional users find it comfortable to utilize their organizational process documentation and to participate at workflows.

The Publishing and Deployment features in IYOPRO ensure, that the process documentation shown in the portal consists only out of released and approved process models.

With the integrated lifetime management you can control the validity period of your process landscape and navigate on the timeline in a very easy way.

## Conceptual Overview

The Publishing and Deployment process is based on the following steps:

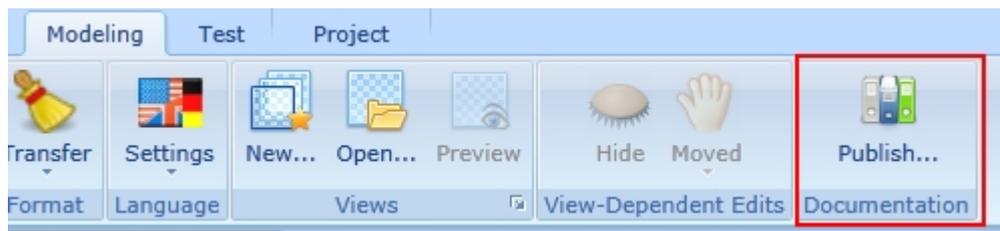
1. [Edit](#)  
Create one or more BPI-Projects and describe your process models
2. [Publish](#)  
Click on the publish button in the modeling tab of the ribbon. The upcoming dialog is used to create a deployment package.
3. [Deploy](#)  
The deployment package can be released or declined by an auditor. This can be done on demand or guided through a [workflow](#), depending on your [team settings](#).  
Note, that the team settings also allow to auto deploy a deployment package. That means, that the deployment is automatically done within the publishing step.

After a successful deployment, the process documentation is available in the teams process documentation folder.

## Publish

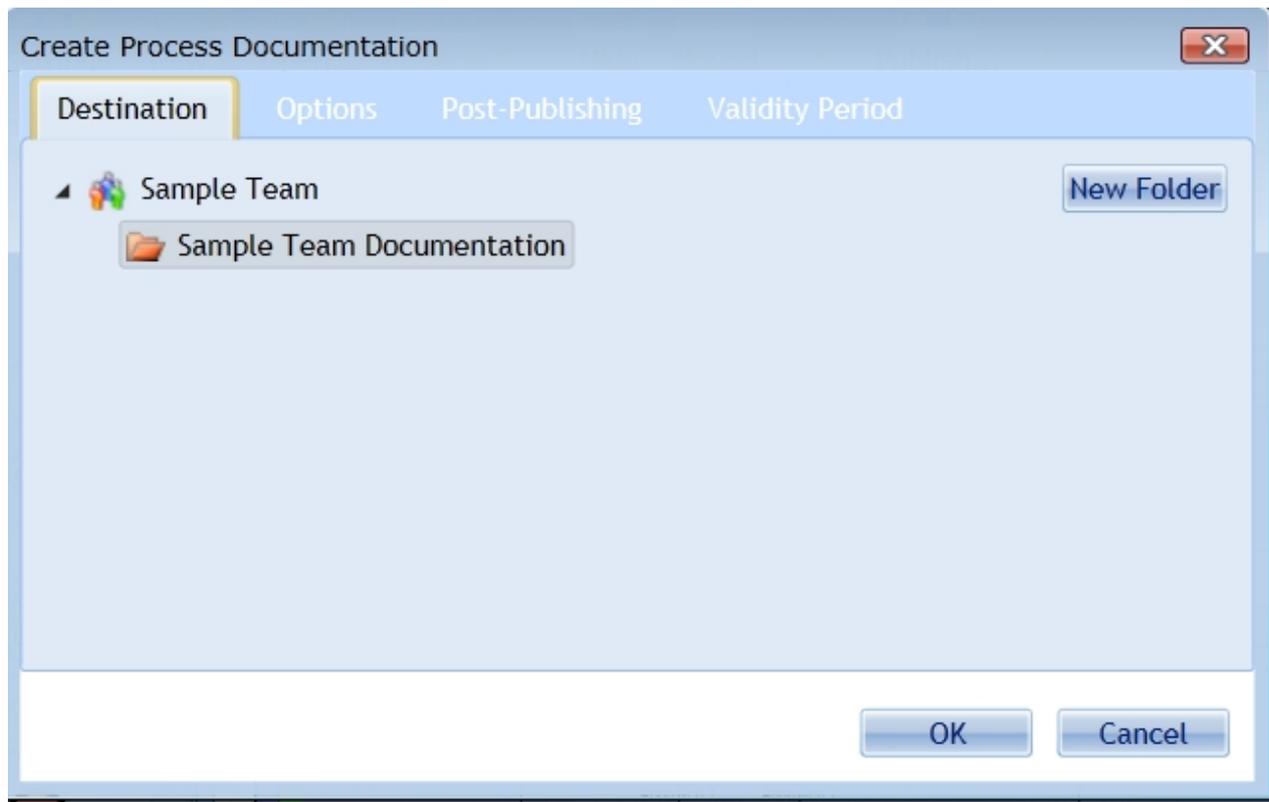
To publish a process documentation to the process portal first open your BPI-Project which contains your process documentation.

Then click on the Publish-Button on the modeling tab in the ribbon.



 Depending on the team adjustment, how process documentation has to be published, the following dialogs may differ

The publishing dialog consist out of four sections:



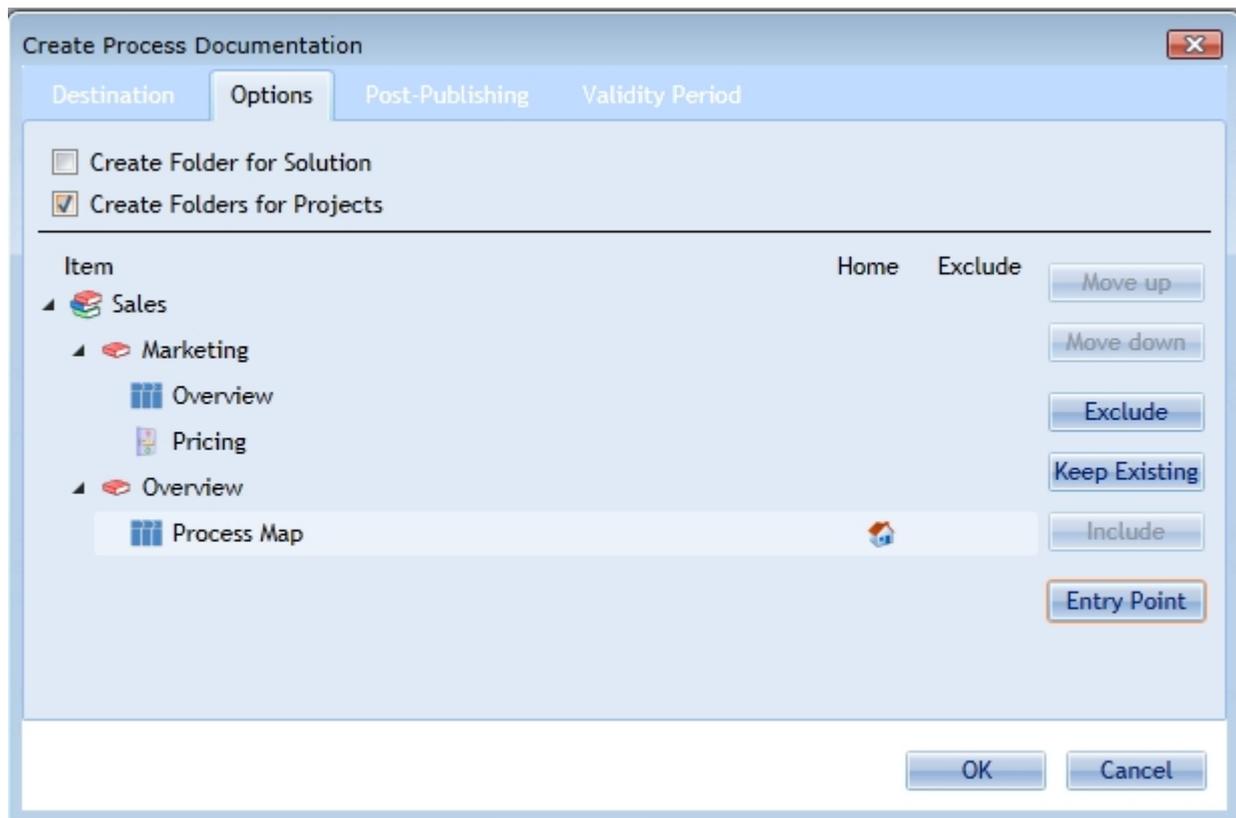
### Destination

Specify the destination, where to insert the process documentation into your existing documentation folder tree.

To create a new folder within the tree, click on the "New Folder" button and specify a folder name. The folder will become part of the deployment package and therefore it will be visible not until the deployment package is released.

### Options

Specify, which elements of your BPI-Project should be part of the deployment package. You may necessarily exclude diagrams which have not been completed or which shall not be published.



To exclude an element highlight it and click on exclude. Excluded elements will be marked with an exclude icon.

To include an element highlight it and click include. All elements are included by default.

It might be necessary to keep an existing element of the process documentation when deploying a new version. If you wish to keep an item, mark it and click the "keep Existing" button.

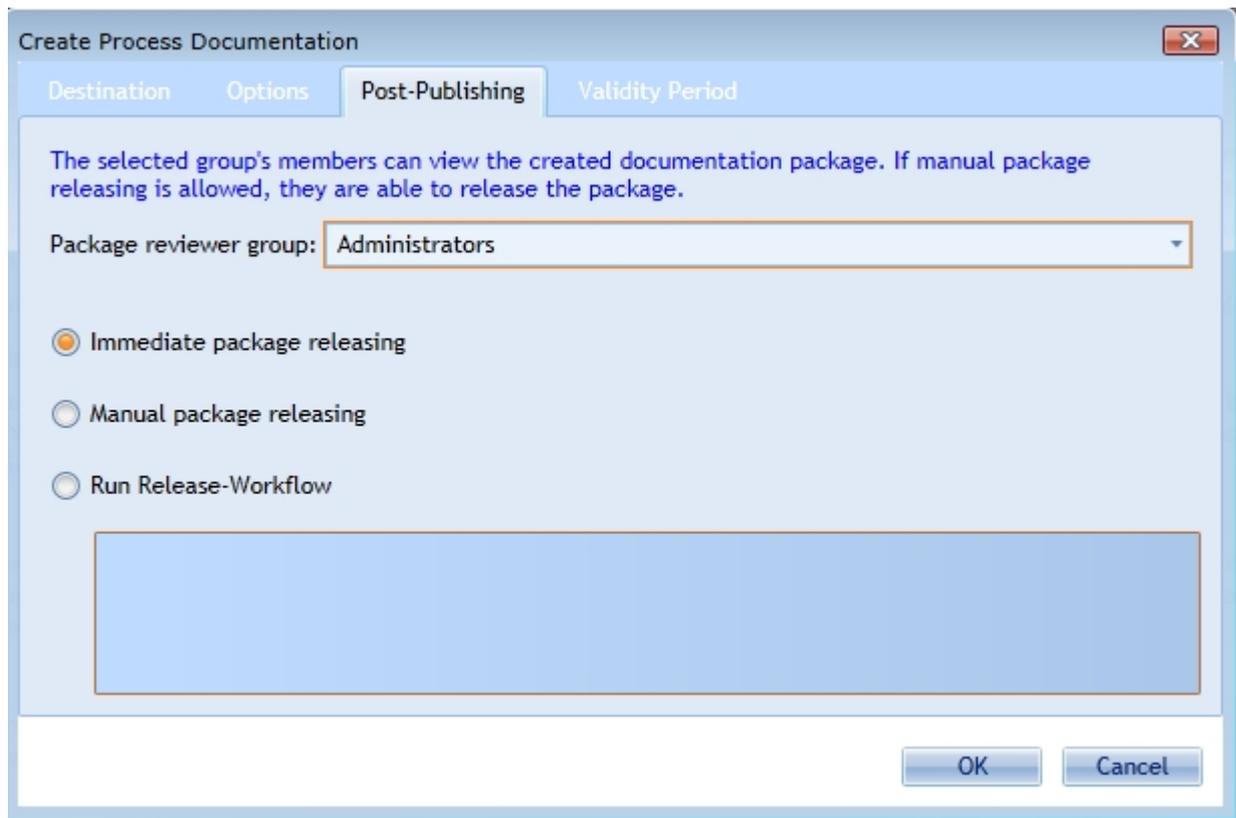
Entry points are root diagrams (usually process maps). They are visualized in the root of the process portals documentation node. To mark a diagram as entry point highlight it and click on "Entry Point"

If you wish to create a folder for the solution beneath the destination folder check the "Create Folder for Solution" toggle.

If you wish to create a folder for each project beneath the desired parent folder check the "Create Folder for Projects" toggle.

## Deployment

Specify how the deployment process will be archived. This setting is affected by your [team settings](#)



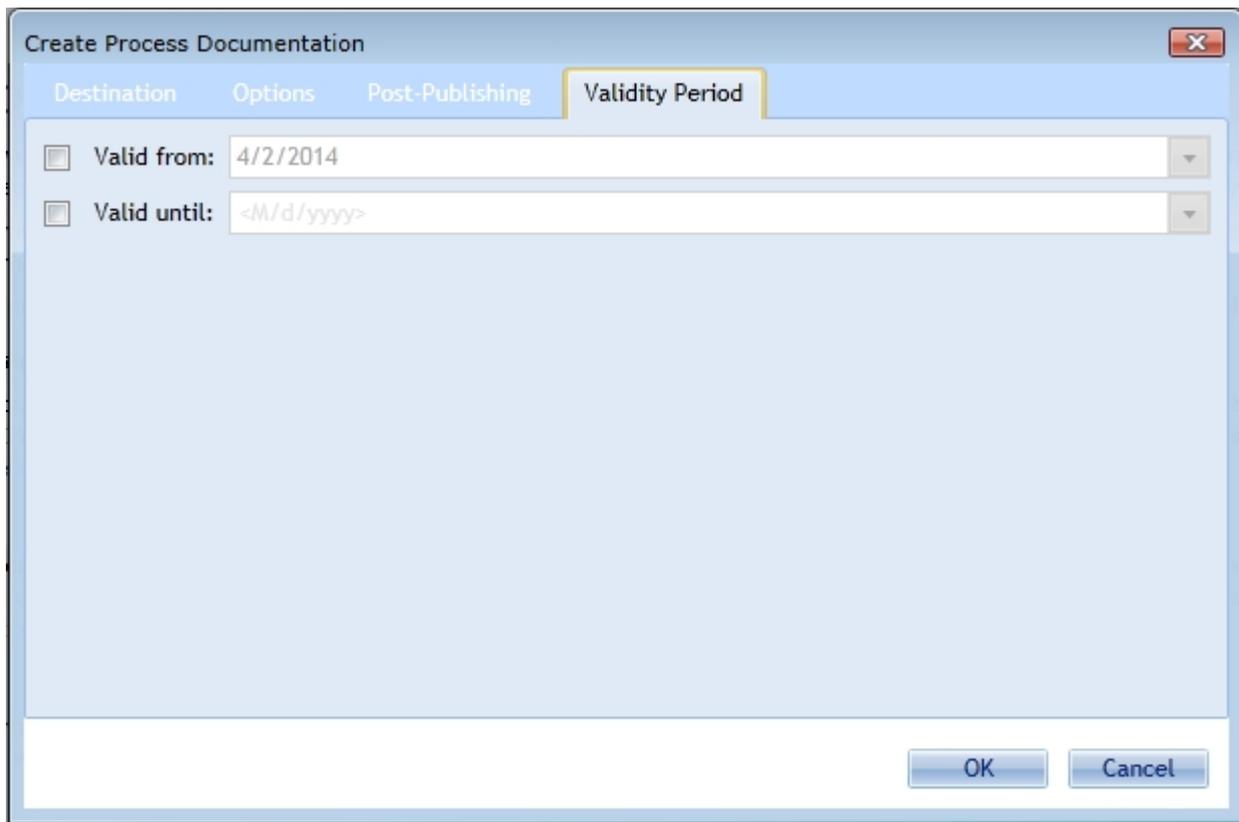
Package reviewer group: Specify a group of users which are responsible to release the process documentation.

Specify the release mode:

- Immediate package releasing  
If you wish to release the package immediately to the portal without auditing choose this option
- Manual package releasing  
Choose this option if a member of the package reviewer group needs to review the documentation package and will deploy the package manually
- Run Release-Workflow  
Choose this option if you wish to automate the release process with the IYOPRO workflow system.

## Validity Period

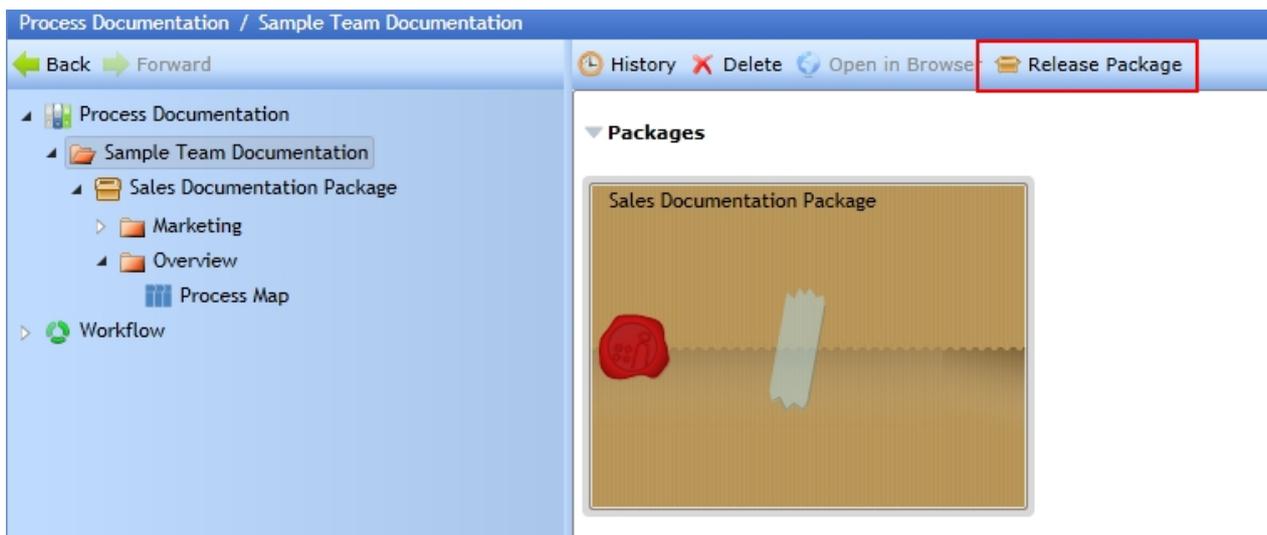
Specify the lifetime, when the process documentation will be in effect.



If nothing other defined, the documentation will be valid from the moment of package release without end time. You can specify other times as necessary.

## Manual Deployment

The manual deployment is done by an auditor which must be member of the package reviewer group. The auditor can navigate through the process documentation and will find the deployment package at the specified location.



The auditor can navigate through the deployment package as it would be deployed to verify, that the package is complete and correct.

The auditor then has two options:

- Release Package

Click on the "Release Package" button to deploy the package. This makes the documentation available for the readers

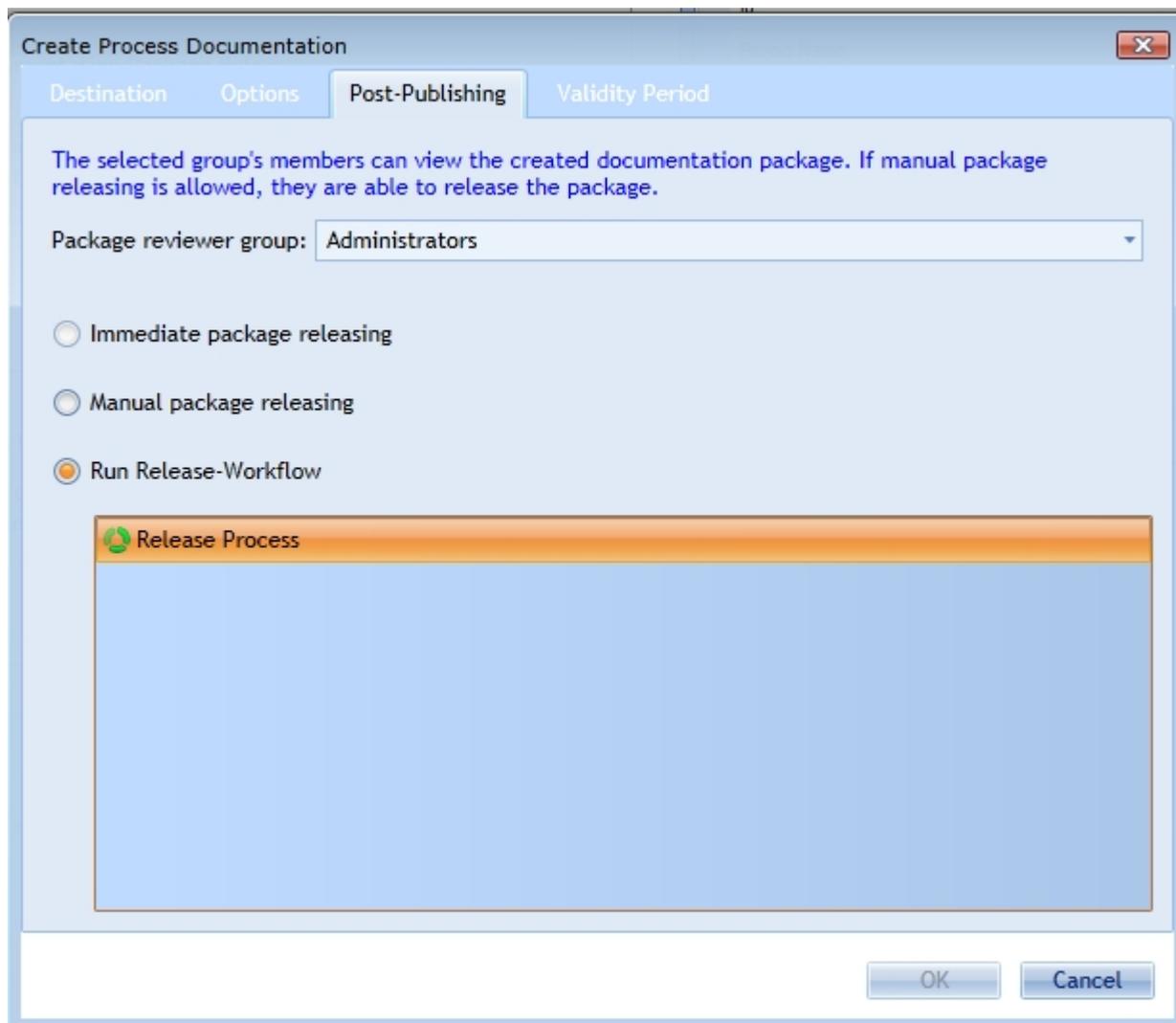
- Delete
- Click on the "Delete" button to decline the package. This will delete the deployment package. The author now can revise his work and repeat the publishing process.

## Workflow Deployment

The release of deployment packages in YOPRO may be performed by a release workflow. This has the advantage, that the release is documented and that the publisher can monitor the state of the release process.

[Here](#) you find a step by step guide to set up a sample documentation release workflow.

If you have created at least one release workflow it can be chosen in the publishing dialog:



# Business Process Simulation

---

IYOPRO distinguishes between

- Process Flow Animation
- BPS Process Simulation & Analysis

## Process Flow Animation

Process Flow Animation allows you to visualize the flow of your Process in a graphical interactive way. It is used to verify, if the process flow is correct, the shapes are right chosen and a possible proper behavior of a workflow definition.

Therefore Process Flow Animation is very easy to use and became a standard tool for most process designers.

## Process Simulation & Analysis (BPS)

BPS Process Simulation & Analysis is a highly sophisticated method to analyze business processes, based on a given business scenario and to provide key performance indicators (KPI's). Examples for such KPI's are:

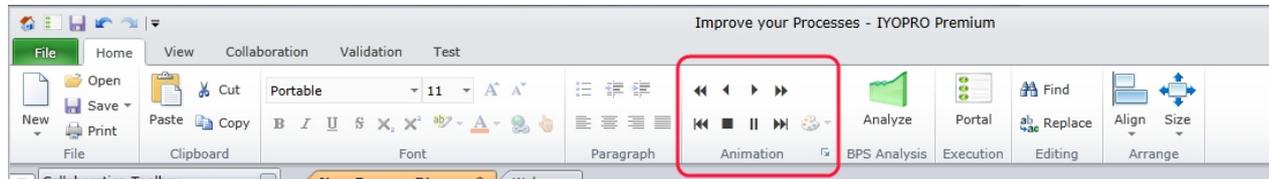
- Utilized Capacity  
The utilized capacity describes to what extent a resource (e.g. a machine, work force) is utilized or alternatively how much capacity is still available. It is determined by the ratio of the degree of utilization and the maximum capacity. A utilization of 75% is aspirational. Exceeding the KPI or falling below permanently or considerably often can be seen as an indication for possible over capacity or resource bottlenecks.
- Process Cost  
Process cost is the accumulated cost of a business process beginning with the first process step until the process end, incorporating the process result. Generally it's the partial cost of involved resources (personnel, material cost) as well as transport cost.
- Process Cycle Time  
The process cycle times are the accumulated times of a business process beginning with the first process step until the process end. Generally we are taking execution times, transport times, holding times and changeover times into account.
- Capital Commitment  
The capital commitment KPI shows how much capital is tied up by the respective process steps. That way you can for example identify whether your process involves extensive activities at the beginning which tie up company capital.
- Efficiency  
The efficiency view shows the ratio of execution times and holding times aiming for a minimization of the holding times.

Using these KPI's you can achieve the following goals in addition:

- Optimization of sequential and parallel process steps
- Simulation & documentation of various process scenarios (worst case - best case)
- Process step analysis
- Consideration of time, resources and costs in the process context
- Map exercises for identification of the optimal process flow
- Target-performance comparison: Do I achieve my objectives?

## Process Flow animation

The Process Flow Animation is found in the Animation Group of the Home Tab.



It supports the following operations:

[Start flow animation](#)

[Stop and pause flow animation](#)

[Step / Fast forward](#)

[Step / Fast backward](#)

The result may be visualized in the following manner:

[Normal](#)

[Color by split / join elements](#)

[Path to selected element](#)

[Timeline](#)

### Start flow animation

To start a process flow animation load or create a BPMN 2.0 Collaboration diagram.

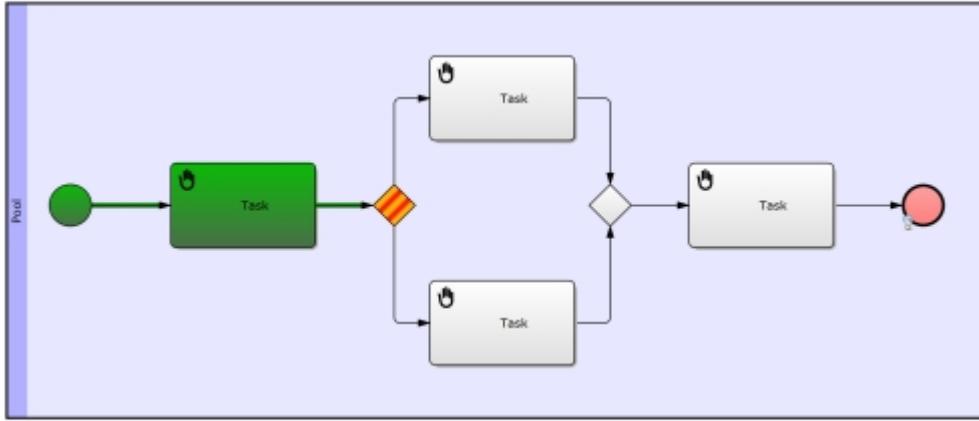
Ensure, that no elements are selected and click on the ▶ Button. The start events will change their color to red which indicates, that a Token is owned by this Element.

Every click in the ▶ button will move each token along the process flow to the next position.

If a token leaves an element, its color changes from red to green, indicating that the element has been processed whereas the receiving element changes its color to red.

If a token cannot leave its color changes to yellow. If for Example a receive message task got a token and waits for a message, its color will be yellow until the message arrives.

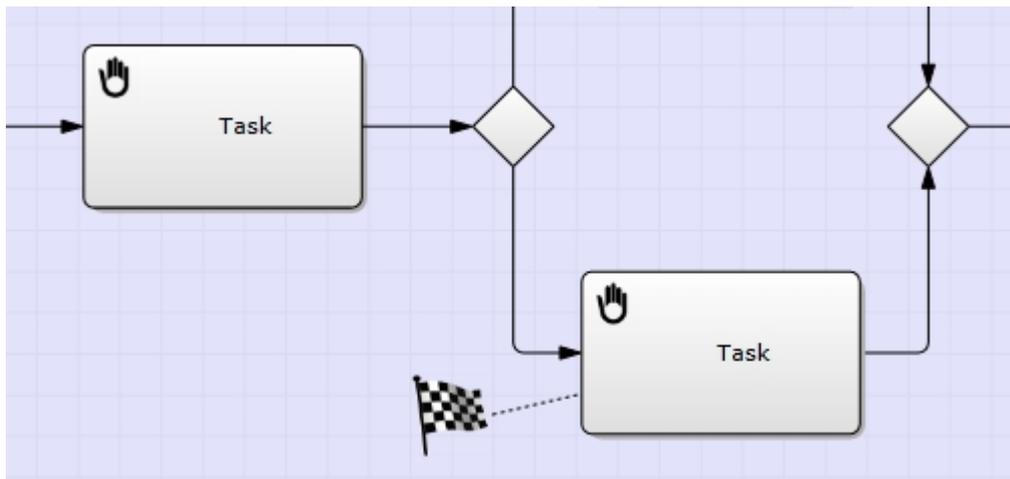
If a token enters a decision gateway or an activity with attached events its color changes to red/yellow striped. The decision about the outgoing flow can be made through a selection of the outgoing path / paths (if applicable).



You may also start the flow animation by using the ►► button. In this case the animation will automatically step forward until all tokens cannot move any more.

If you start the process flow animation on a diagram, which contains selected elements, the animation will begin on these selected elements.

Tip: If you wish to start the process from a predefined position, you may also drag a start flag from the [animation toolbox](#) to your diagram and attach it to the start element.

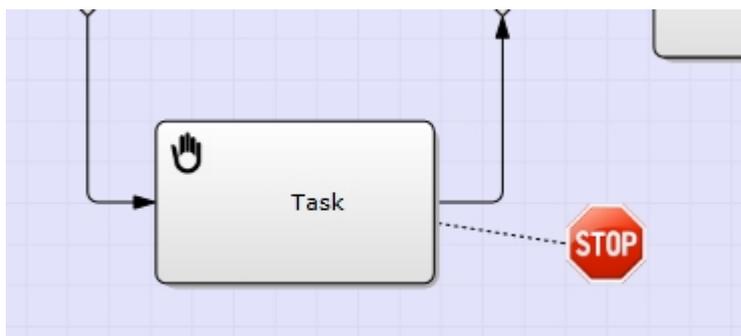


### Stop and pause flow animation

To stop a process flow animation click on the ■ button.

To pause a process flow animation click on the || button.

If you wish to stop at a predefined position you may wish to drag a stop sign from the [animation toolbox](#) to your diagram and attach it to the stop element.



## Step / Fast forward

To step a process flow animation click on the ▶ button.

To fast forward a process flow animation click on the ⏩ button.

If you wish to run to a point, where all tokens are unable to move you may click on the ⏸ button.

## Step / Fast backward

To step back a process flow animation click on the ◀ button.

To fast backward a process flow animation click on the ⏪ button.

If you wish to step back to the beginning you may click on the ⏮ button.

## Result visualisation

The result may be visualized in the following manner:

[Normal](#)

[Color by split / join elements](#)

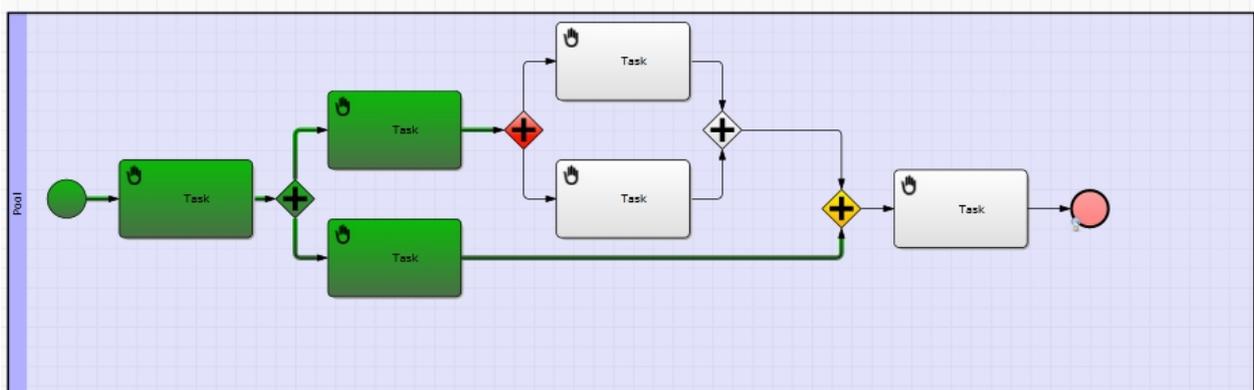
[Path to selected element](#)

[Timeline](#)

### Normal

The normal process flow animation uses the following colors:

Original	The Element has not yet been involved in the animation flow
Red	A token is currently owned by the control
Green	A token was previously owned by the control.
Yellow	A token is currently owned by the control. It cannot be processed because the control awaits more input
Red / Yellow striped	A token is currently owned by the control. The flow direction offers multiple possible outputs which must be chosen by the user.

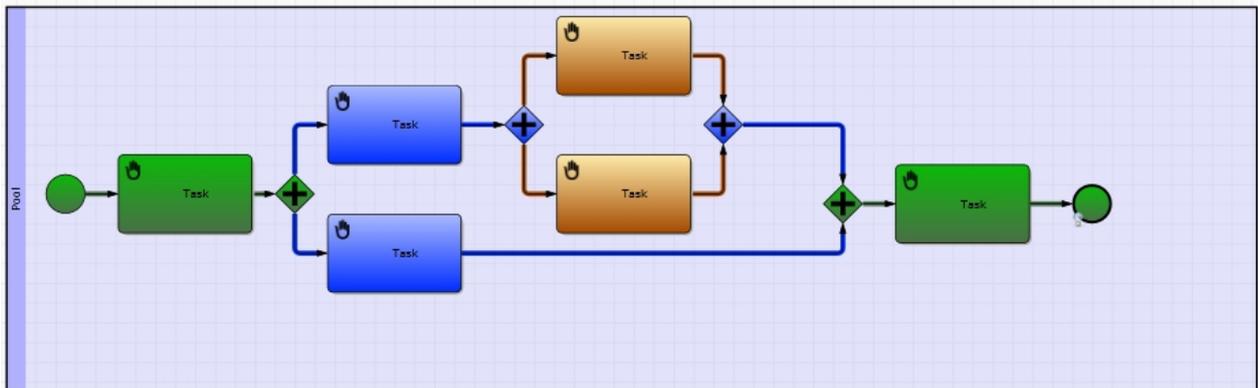


Normal animation colors

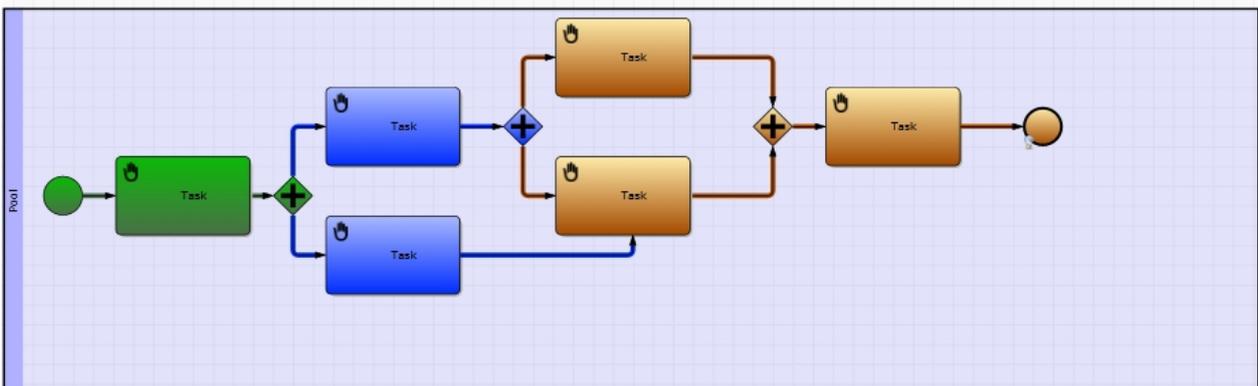
### Color by split / join elements

The Color by split / join elements method provides a way to visualize, if splitting and joining gateways match or not.

IYOPRO will change the color at each splitting gateway and revert it at the corresponding joining gateway.



Matching process definition

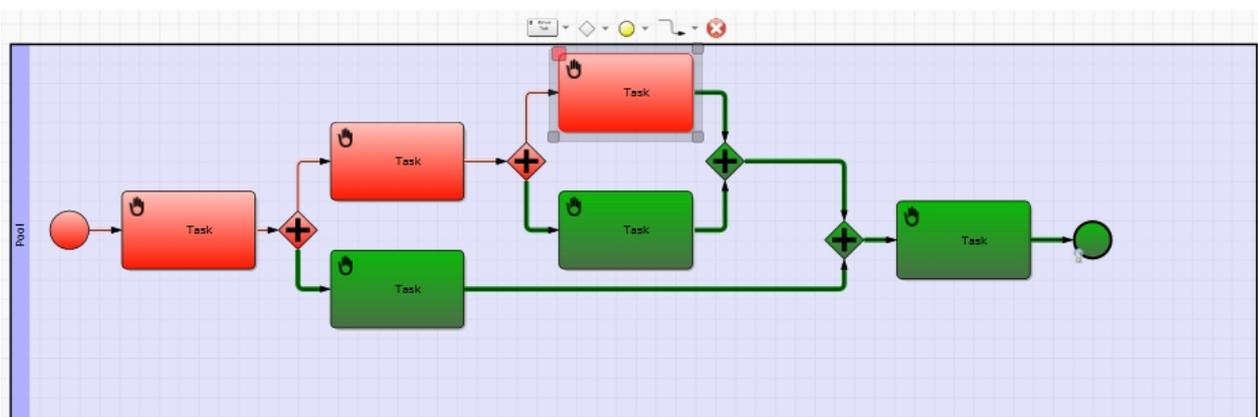


Not matching process definition

### Path to the selected element

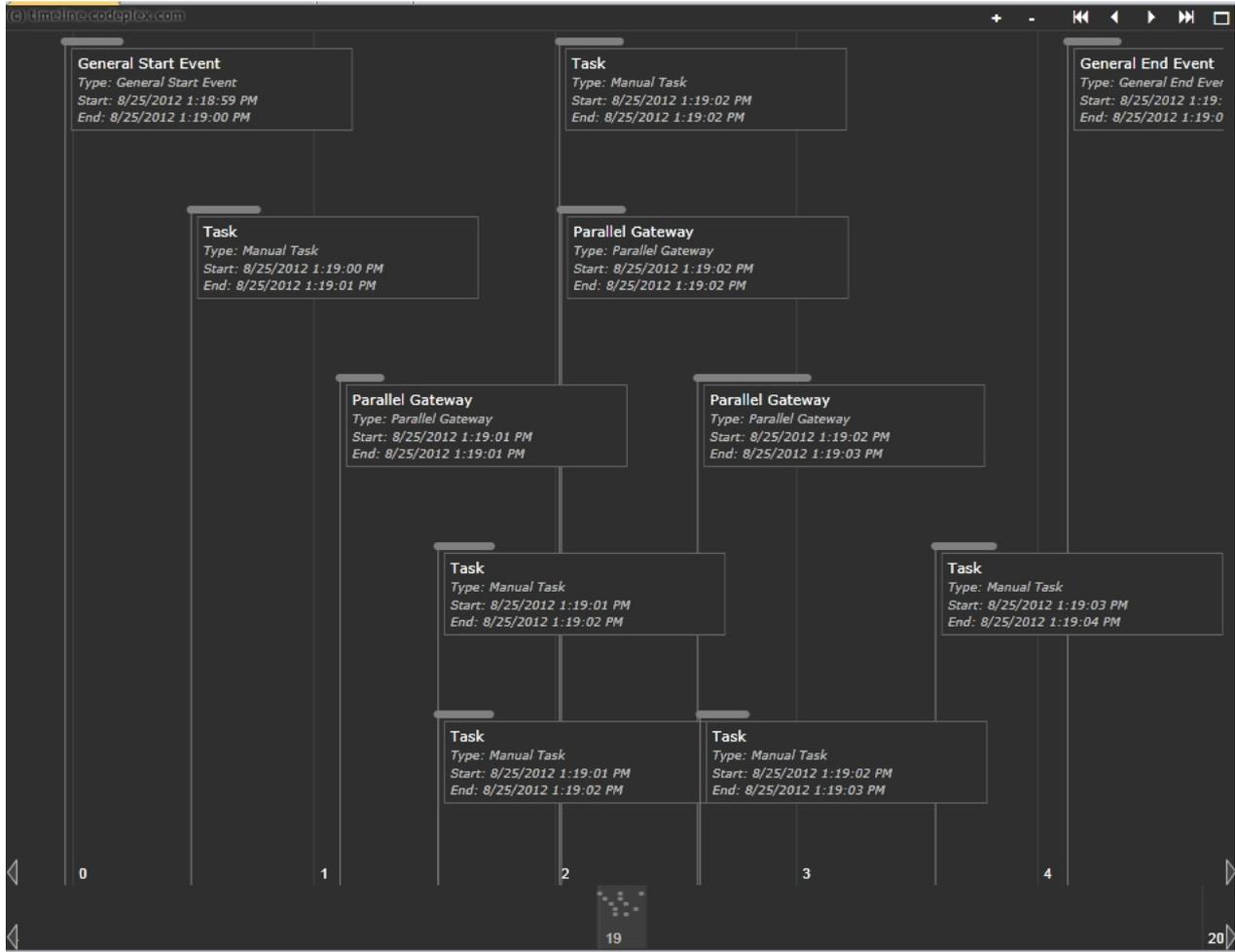
The Path to the selected element method provides a way to visualize, how the process flow was from the beginning until the selected element was reached.

IYOPRO will change the color for each visited element to red, all others to green.



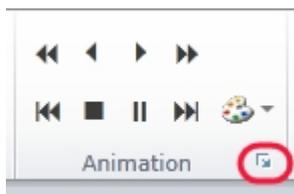
### Timeline

The Timeline method provides a way to visualize, in which order the elements have been reached.

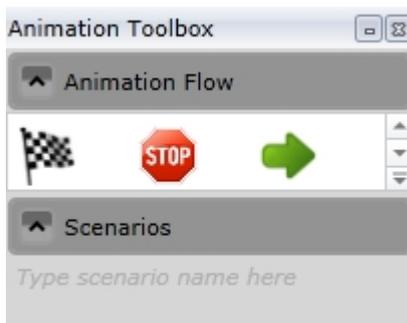


## Animation Toolbox

The Animation Toolbox can be opened by the dialog launcher of the animation ribbon group.



The Animation Toolbox will appear.



The animation toolbox contains the following shapes:

Symbol	Element	Description
	Start Flag	Attach a start flag to the Element where to start your animation from
	Stop Sign	Attach a stop sign to the Element where to stop your animation at
	Arrow	Attach an Arrow to the Sequence flow where to follow if multiple conditions are possible

## Business Process Simulation & Analysis

### Why simulating BPMN 2.0 processes?

The aim of business process simulation is the improvement of process characteristics like quality, performance and/or resource utilization. Therefore ideally known processes are measured and key performance indicators are calculated. The results of the simulation experiments are presented in diagrams. By comparing the results of different process configurations, it's easy to determine the best configuration to archive the requested target.

### How to simulate BPMN 2.0 processes?

You need to enhance your models with some additional parameter information. First of all, the average [duration of activities](#) or the [effective dates of events](#) have to be assigned to the process diagram elements. For mapping the realistic variability of such parameters, pseudo-random-numbers are used, which are generated by [stochastic distributions](#). Secondly, you ave to ensure that all [exclusive and inclusive gateways](#) have determinable conditions and/or probabilities. After entering all relevant parameters you can [start the simulation experiment](#) and afterwards analyze the [results](#). Since simulation is based on stochastic, you need to execute various simulation experiments.

IYOPRO also offers some advanced simulation concepts:

- You can determine [resource utilization and bottlenecks](#), if you link your processes with entities from [organization diagram models](#).
- You can [calculate activity based costing](#) of your processes and products.
- You can use Python scripts to access [simulation context data](#) for complex computations

### Content

IYOPRO Premium offers a simulation tool kit for substantial business process analysis. It has the capability of stochastic, discrete event driven simulation of processes together with the ability to analyze the dynamic behavior of business processes. It offers forecast indicators to estimate the potential run-time of processes in order to consider timescale, resources and costs to determine the optimal process by experimenting with different options.

- [General Simulation Properties](#) - Some general settings you need to execute simulation experiments
- [Stochastic / Random distributions](#) - Why and how to use stochastics. Explanation of available distributions in IYOPRO.
- [Assigning Inter-arrival Times / Durations](#) - Explains how to use events and activities in simulation experiments
- [Element settings](#) - Offers a complete list of available options for your model elements
- [Gateway Decision Making](#) - Explains how simulation chooses the next element after an exclusive or inclusive gateway
- [Process Variables](#) - Explains how to model implicit knowledge in processes, e.g. for decision making at gateways
- [Resources](#) - How to analyze resource utilization e.g. for bottleneck analysis.
- [Report](#) - How to analyze experiment results. This chapter explains the large number of provided statistics.
- [Simulation Tutorial](#) - Learn how to simulate BPMN 2.0 models step by step with a lot of example models

## Starting Simulation

Once your model includes all simulation specific parameters / settings, you can execute simulation experiments. Afterwards you can analyze the experiment results.

### Start Buttons

Select Simulation from the property set in the Properties window. After that, you can see the Ribbon Tab Simulation in the main menu, which contains the following buttons:

 <p>Diagram</p>	<p>Clicking this button will start the simulation on the currently selected diagram (BPMN model).</p>
 <p>Project</p>	<p>Clicking this button will start the simulation on the currently selected BPI project. All the models of the project will be simulated, and there will be a report generated for each one.</p>

### Requirements

You need to enrich your models with additional information, like [duration parameters for activities](#) or [point in times for events](#). You have to ensure that all [exclusive and inclusive gateways](#) have determinable conditions and/or probabilities. If you have not set all of these parameters, simulation will not start. Once you clicked a start button, IYOPRO will help you to enter the missing parameters. Open the Output view for a list of missing information.

### Experiment Result

If simulation was successful, a report is generated. See section [Report](#) to learn more about experiment results.

### General Simulation Settings

In order to simulate your models, you need some additional settings. This page introduces the general properties like the duration of a simulation run and the stochastic seed.

Once you have selected Simulation from the property set in the Properties window, you can see these General Simulation Properties by clicking any area in the document (in the IYOPRO-modeling window) but not an element itself.

### General Simulation Properties - Overview

<p>Must-Have Simulation Properties</p>	
<p>Experiment Description</p>	<p>Your experiment's description. This text will be shown in the simulation report. (Can be any meaningful text on the model and the experiment.)</p>
<p>Stochastic Seed</p>	<p>The global stochastic seed for all random distributions in the model. See <a href="#">stochastic distributions</a> and <a href="#">stochastic seeds</a> for more information about that.</p>
<p>Simulation start date</p>	<p>Pick the simulation start time.</p>
<p>Simulation stop time (in hours)</p>	<p>This value determines the stop time of the simulation, the default value is represented in hours.</p>
<p>Simulation stop time unit</p>	<p>Pick the time reference unit (Hours, Days, Years) for the simulation stop time. The default value is "Hours".</p>
<p>Report Properties</p>	
<p>Trace record</p>	<p>This option allows you to turn on the trace report, which contains all state changes at discrete points in time throughout the simulation. So if you want to reconstruct some specific area in the result, you can use this detailed report to comprehend why something happened the way it impacted in the result.</p>
<p>Debug record</p>	<p>This option allows you to turn on the debug report, which mostly is used by software</p>

	developers. The typical simulator should stick to the trace record.
Debug / Trace start time (hours)	Specify the start time of the debug or trace report. The debugging or tracing function will hook in to the simulation at this particular point of simulation time.
Debug / Trace stop time (hours)	Specify the start time of the debug or trace report. The debugging or tracing function will be terminated at this particular point of simulation time.
Optional Simulation Properties	
Expression	This property allows you to define globally (throughout the simulation) known variables. See <a href="#">Sequence Flows</a> or <a href="#">Activities</a> for more information on the expression property.

You also have to set simulation properties at each single BPMN 2.0 element. See [inter-arrival times / durations](#) and [element settings](#) for more information.

## Random Distributions

Simulation of process models relies on random distributions also called stochastic distributions.

Why? Not all potential influences on a system can be detected and modeled deterministically. E.g. the execution time of an operation exported by a human being is always being subject to fluctuations. Typical other examples are order sizes, processing times or success probabilities. Nevertheless, the realistic variability of such characteristic factors can be indicated with help of pseudo-random-numbers, which are generated by stochastic distributions.

However, the use of stochastic requires a sufficiently large number of simulation experiments. Thus, only the aggregation of results of several experiments and/or very long running experiments will provide reliable information about a system.

Note: The generated 'random-numbers' are so-called pseudo-random-numbers, because micro-processors work deterministically. No micro-processor and no software will ever be able to produce 'real' random-number like a dice does. That's why random-number-generators need [seeds](#) for initializing, and as a consequence we are luckily able to reproduce any set of random-numbers. A dice never could do that.

- [Description of stochastic Distributions](#): Offers a complete list with descriptions to the available stochastic distributions.
- [Stochastic Distribution Editor](#): How to use the distribution editor of IYOPRO.
- [Stochastic Seeds](#): Explains the term Stochastic Seed and how it can be used.

## List of Stochastic Distributions

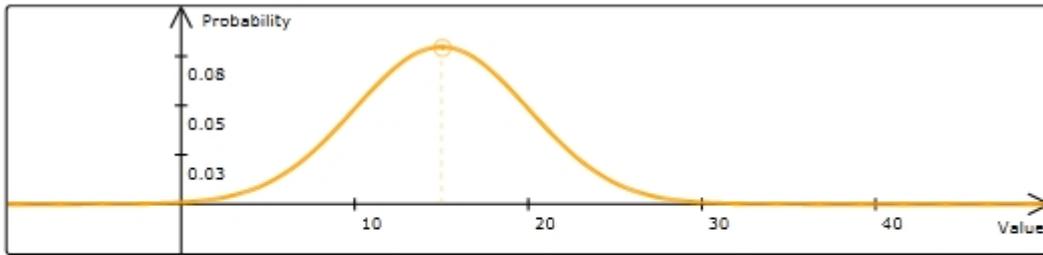
This section explains the stochastic distributions you can use in IYOPRO.

### Constant Distribution

The constant value distribution has only one single possible value. There is therefore no "randomness". In some cases this distribution can be very useful; e.g. for debugging and for comparisons of different distributions' effects on simulation results.

### Normal Distribution

The normal distribution is the continuous counterpart of the Binomial or Poisson distribution. The sum of "sufficiently" many Bernoulli or Poisson distributed random processes (e.g. the number of customer arrivals during a given time span) approximates a normal distribution. The sum of measurement errors or the number of order completions of a machine with stochastic processing times provides good examples for this type of distribution.



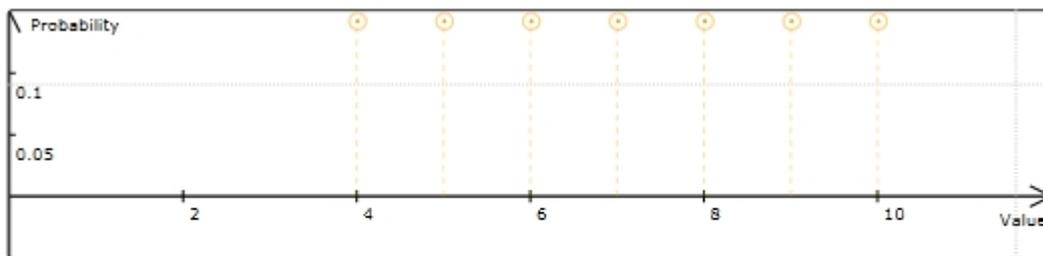
0.00 | 0.00

Mean:

StandardDeviation:

### Discrete Uniform Distribution

The discrete uniform (or Laplace) distribution assigns the same probability to each of  $N$  events. It is often used where no better estimates can be given.



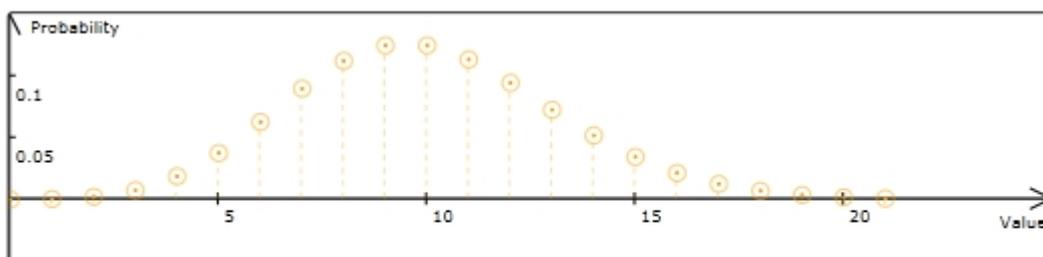
11.62 | 0.09

MinValue:

MaxValue:

### Poisson Distribution

A Poisson distribution approximates expected results of a binomial distribution well if probability is small and sample size is large. As long as the chosen interval is quite short, Poisson distributions can be successfully used to describe e.g. the number of arrivals of customer's telephone calls in a model.

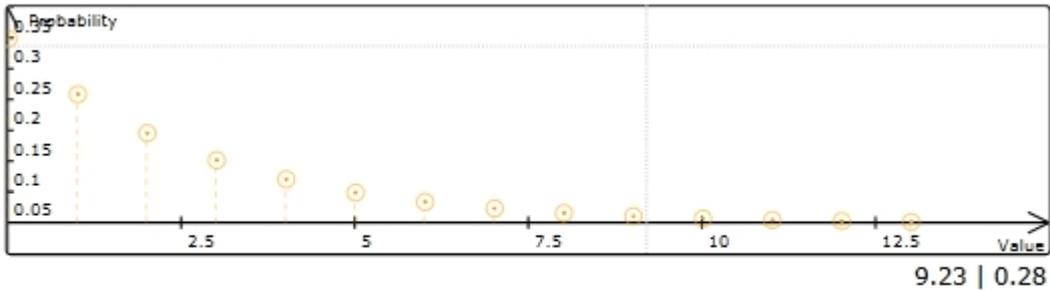


0.00 | 0.00

MeanValue:

### Geometric Distribution

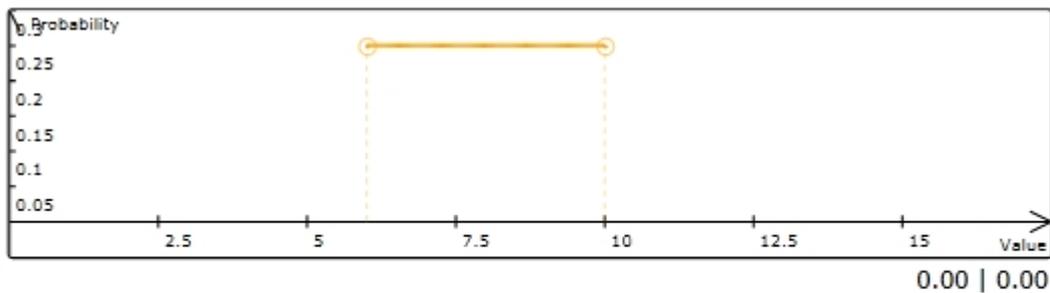
The geometric distribution describes the probability that exactly  $k$  Bernoulli experiments will fail before the first success. For example, consider a production system in which a certain step can be described as Bernoulli experiment: The production step may fail, thus the success probability is  $p < 1$ . Assume that for each item produced this step is repeated until success. The number of repetitions necessary will be geometrically distributed.



Probability:

### Continuous Uniform Distribution

Random numbers which follow a rectangular distribution (or continuous uniform distribution) are distributed over an interval, whose subintervals all have the same probability of occurrence. While this type of distribution is rare in reality, it is useful where no better knowledge is available and cannot be obtained with acceptable effort.

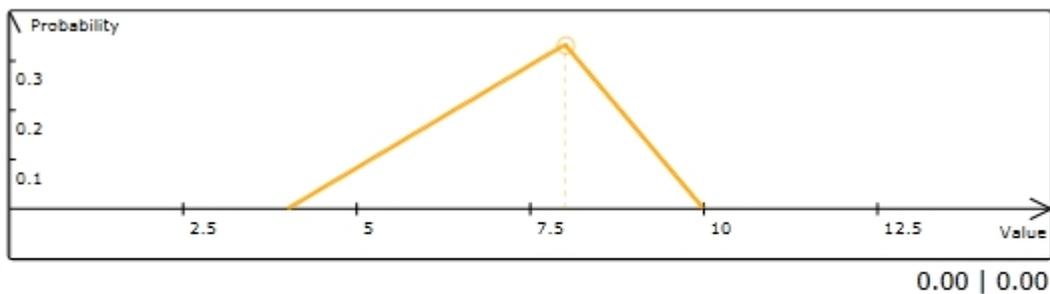


LowerBorder:

UpperBorder:

### Triangular Distribution

In contrast to a rectangular distribution, in a triangular distribution values closer to the mode occur more probably than values near a or b. Triangularly distributed random numbers are also used where a part from other indications only knowledge about most frequent values (in addition to lower and upper bounds) is available. Capacity estimates for natural resources like coal mines or oil wells are an example.



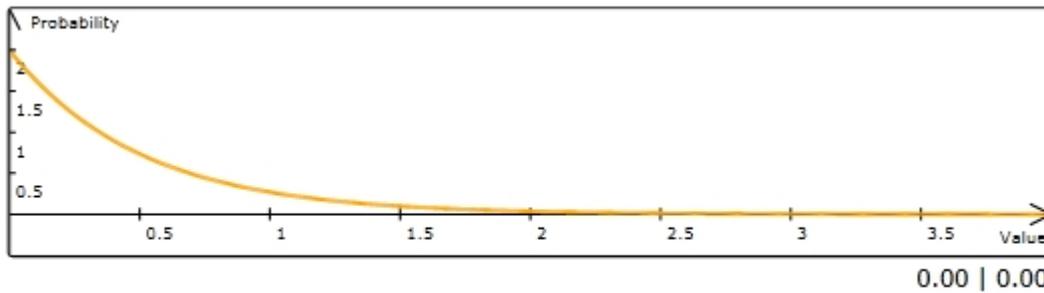
Lower:

Upper:

Peak:

### Exponential Distribution

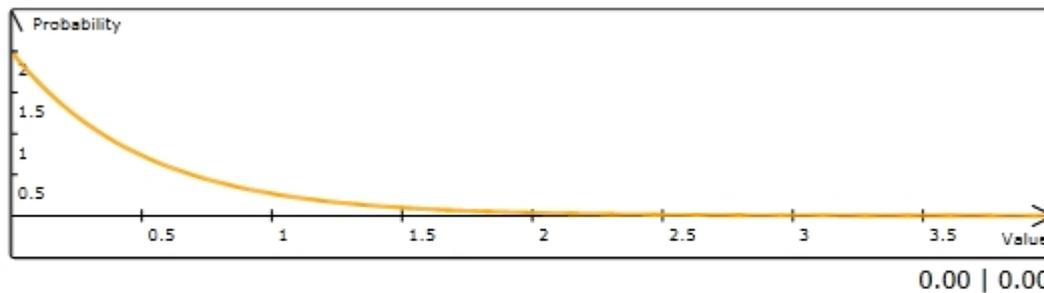
The exponential distribution is the continuous counterpart of the geometric distribution. Inter-arrival times of customers or orders in systems without fixed time intervals typically follow exponential distributions.



Mean:

## Erlang Distribution

The Erlang distribution is created by a aggregation of independent exponential distributions. In practice this is done by adding  $n$  exponentially distributed random numbers for each sample. Erlang distributions describe waiting times in a queuing system. Characterizing the life time of products or the length of time between system failures are other common examples for their application.



Order:

Mean:

## Empirical Distribution

The Empirical distribution assigns probabilities of appearance to empirically observed values. This distribution should be only used, if no other probability distribution fits with the given empirical data.

## Stochastic Distribution Editor

The stochastic distribution editor (see the picture below) is used to parametrize stochastic distributions, e.g. [duration parameters for activities](#), [inter-arrival-times for events](#) and stochastic based process variables. You can open the editor via the property editor of the corresponding element. See also a complete [list and description of available distributions](#).

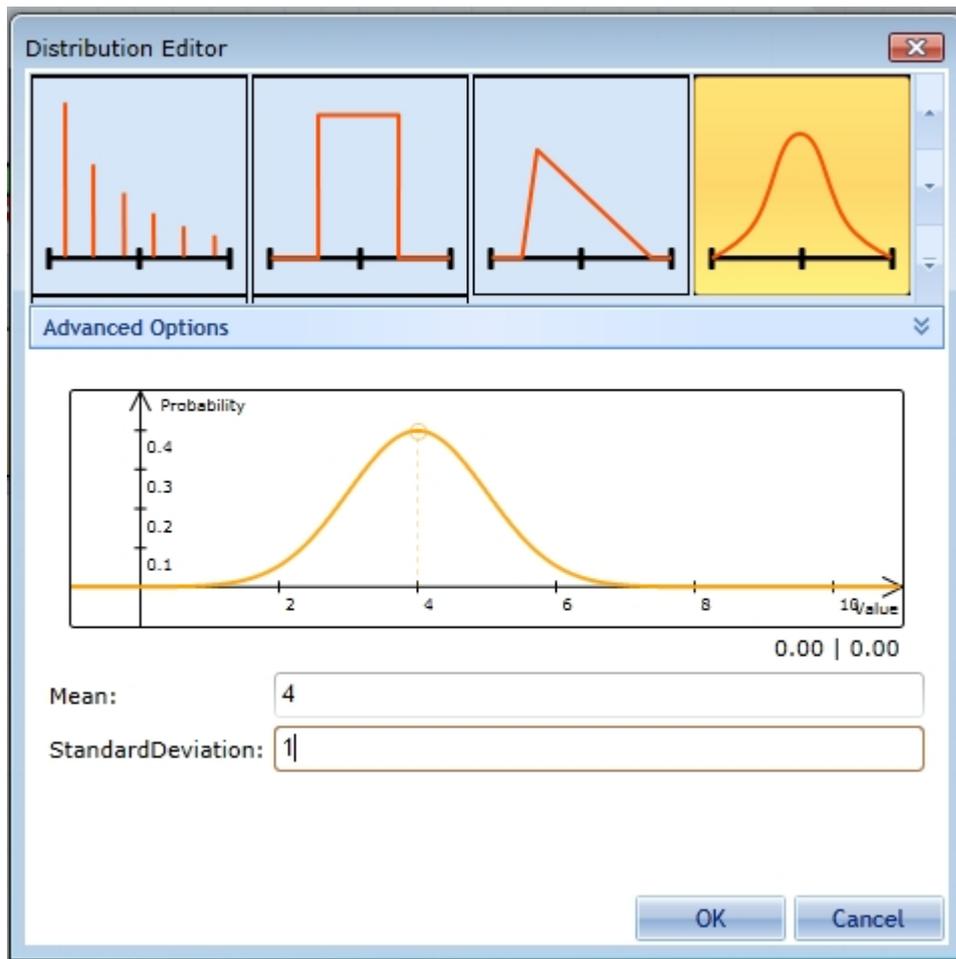


Figure 1: Stochastic Distribution Editor

Every time you select and parameterize a distribution a curve will be drawn, which predicts the value distribution in the assigned space.

Note that:

- every distribution possesses its own parameters, which determine the probabilities of certain values.
- when specifying durations
  - you can enter the values as simple numbers like in the figure above)
  - or you can use the time format to assign parameters.
  - For the latter, e. g. you could enter "02:30" or "2:30" for every two and a half hours, which is also represented by the number 2.5.
  - If you want to go further and assign a duration of ten seconds, you will need to add the leading zeros for hours and minutes. Example: "00:00:10"
  - You can also add days to this format by simply putting the "number of days" followed by a dot. Example "13.02:30:10" which means that the time span is 13 days, two hours, 30 minutes, and 10 seconds.

## Advanced Options

Advanced Options	
No negative number generation	Check this if you do not want the simulation to generate any negative numbers from the selected distribution. This value is checked by default (true). It cannot be changed, if the distribution is used to generate values for a time consumption. But you do have the possibility to generate negative numbers where it makes sense (see this help's topic on <a href="#">Process Variables</a> for more

	information).
Antithetic number generation	<p>This option allows you to generate the exact opposite values of the actually drawn values. The definition of an exact opposite (antithetic) value depends on the distribution type. Every distribution implements an own algorithm to determine this value. E. g. you select the Discrete Uniform Distribution with the parameter setting: MinValue= 2, MaxValue=6 for the duration of an activity. If you check this option and if we assume the first (actually) drawn value is 2, then the first result will be 6, because it is the antithetic value to 2 in this particular distribution.</p> <p>This option can be useful, if you want to analyze a specific activity. Just like setting a manual seed for a certain distribution, this is a way to obtain significantly different generated numbers for exactly one BPMN element that uses any kind of distribution.</p> <p>It is unchecked (false) by default.</p>
Manual seed instead of autoseed	<p>By checking this option, you can set the seed for exactly this distribution to a value, that you determine. This way the distribution will not be dependent on the global seed.</p> <p>This approach is useful when you want to analyze one particular activity (or anything else using a distribution) in several simulation runs. The point is to leave all the other options constant and changing only the seed for this particular distribution.</p>

## Stochastics Seeds

Stochastic seeds permit reproducing a set of random-numbers. For getting a new set of random-numbers you just have to change one value: the seed. If you reuse a seed, you'll get the same random-numbers again.

- The stochastic seed is a basic value for all stochastic distributed values in the simulation from where the random numbers are generated.
- Running the same experiment twice without changing any parameters will result in two identical results.
- By changing the stochastic seed between two simulation runs (and by keeping all other parameters unchanged), the results will be different.
- You can set the Stochastic Seed as an integer in the Must-Have Properties of the General Simulation Properties.
- You should just try running the same model with a different stochastic seed and then compare the reports.
  - It is indispensable to run multiple simulations on a model, each with different general seeds.

## Local Stochastic Seed

- You can also assign an individual stochastic seed for every stochastic distribution.
- The individual seed is independent of the general seed.
- For example, giving the distribution of any activity an individual seed will result in the same values taken from this exact distribution in every simulation
  - even though the general seed could have changed.

## Activity's Duration and Time-based Events

You need to enhance your models with information about the duration of activities or the effective dates of events. When you are going to assign an inter-arrival time (to an event) or a duration (to an activity), the Time Editor appears (see picture below).

## Three Perspectives

The Time Editor offers three perspectives, which provide different approaches for assigning timing information:

- The [Time Span](#) mode: This allows you to set up any distribution for an inter-arrival time or a duration
- Time [Time Instant](#) mode: This allows you to pick the start date from a calendar and the assign a rate at which the event will be repeated
  - This mode allows further formalization of the time instants ([Scripted Time Instants](#)) by using the Python programming language.
  - Note that the Time Instant mode is not available for durations of activities

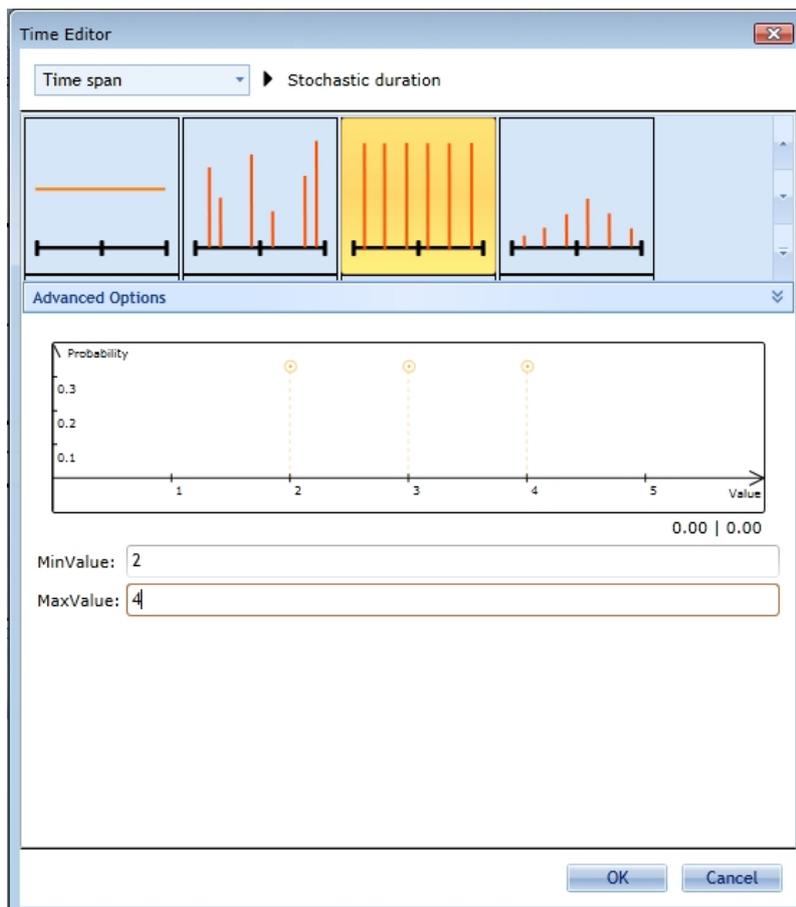


Figure 1: The Time Editor (Stochastic Distribution view)

## Stochastic Event Trigger

### Time Span

If you choose Time span from the drop-down box (top), you can assign one of the distributions (that we also provide for the [stochastic attributes](#)) to get a stochastic duration. Set one of those [stochastic distributions](#), and the simulation will draw sample values from it and use them for the duration of activities or for the inter-arrival time of start events.

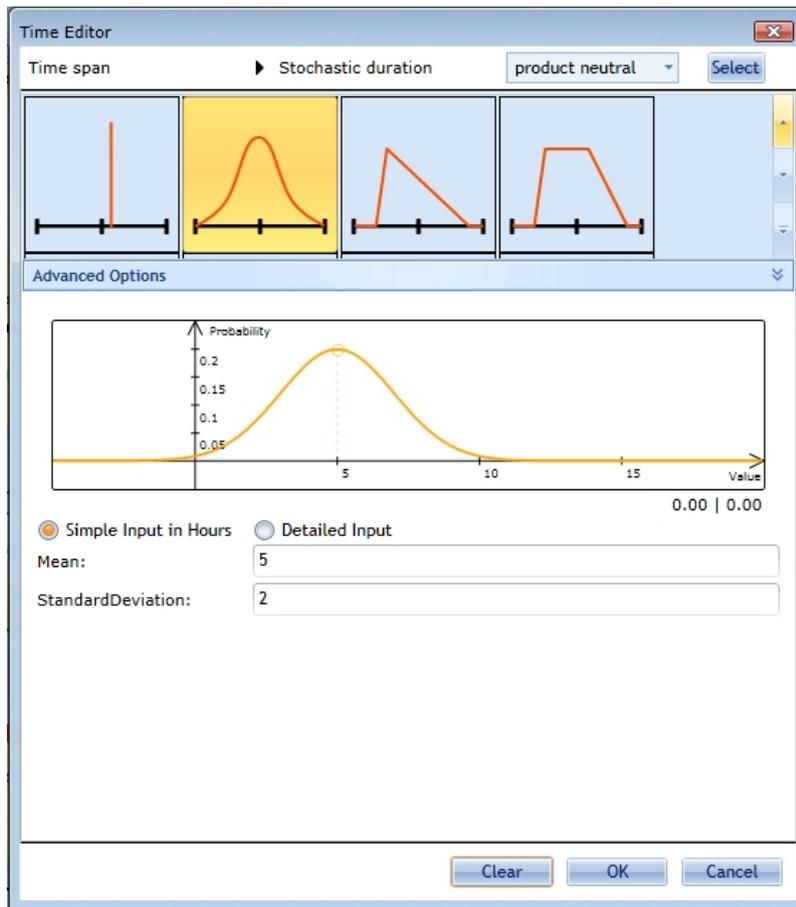


Figure 1: The Time Editor (Stochastic Distribution view)

## Approach

- Simply pick one of the distributions
- Depending on the selected distributions, you will have to assign its parameters
- In the picture above we selected the Normal Distribution, which contains 2 parameters
  - The mean value
  - The standard deviation value

## Note

- We offer you two ways of setting the distribution's parameters:
  1. Simple Input, Figure 1: In this case you need to supply the number of hours for each parameter. Note that the number can be entered as a float.
  2. Detailed Input, Figure 2: The detailed input view allows you to enter an exact time span by specifying the number of days, hours, minutes, and seconds. Due to this restriction you can only enter integers.
    - Attention: The Detailed Input is only available when setting up a time span (e. g. inter-arrival time of events or duration of activities).

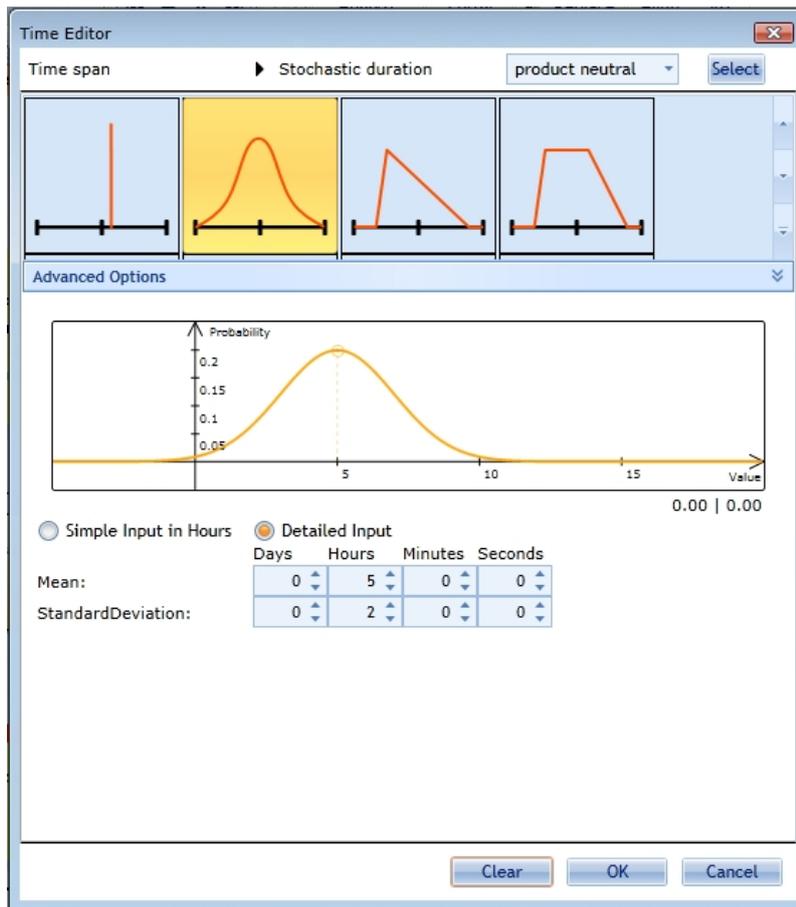


Figure 2: The Time Editor (Stochastic Distribution view, Multiple Input Boxes)

## Calendar based Event Trigger

### Time Instant

Contrary to Time Span a Time Instant determines a specific point in time during the simulation. Since time instants can be specified in a very strict way, we do not have the influence of stochastics, at least not in this element. The "Time Instant" option can be selected in the drop-down menu at the top of the window.

You can define these time instants either as:

- a Recurring Time Instant (See [Model 1c - Time Instants for Start Events](#) from the [Simulation Tutorial](#) for an example)
- or you can script them yourself by choosing the Scripted Time Instants-option from the second drop-down menu at the top of the window.

Note that time instants can only be provided for events, especially general / timer start events. Activities do not support this feature.

### Recurring Time Instant

The name says it all. These time instants recur at certain points of time, which have been specified before. See below for different methods on how to handle the finiteness (or in infiniteness) of the recurring.

The main approach is to select a start date (and time), then select the time unit for the recurrence (e. g. hours). Subsequently you will have to choose the span, (f. e. every 2 hours) . Afterwards you need to determine the end type, which can be open end, after a specific amount of recurrences or by a time at which the recurrences shall stop. See below for more information.

### End - Open End

This one is quite simple. As mentioned before you need to choose the date, time, recurrence unit and span.

In the example below we chose 1/1/2014, 8 AM and an hourly recurrence, which occurs every hour. We chose end type in the "End"-drop-down box, which means that there will be endless recurrences of this time instant. Well, the term endless is barely right, because the simulation stops at a given time.

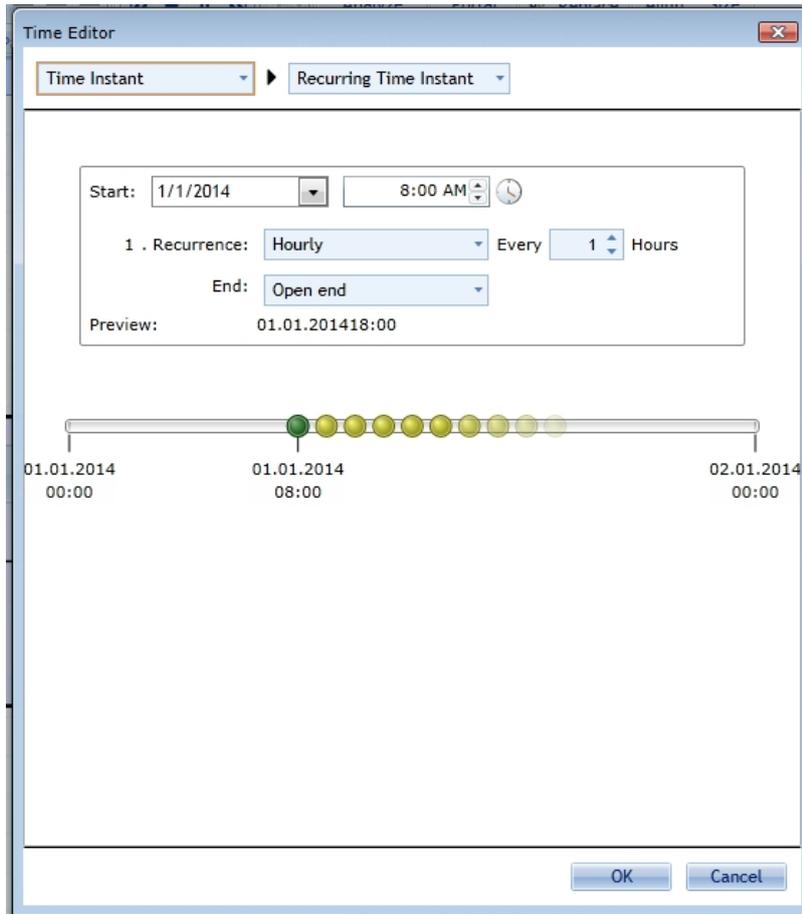


Figure 1: Time Instant - End Condition: Open End

### End - After Amount

This time instant ends after the given amount of 8. By specifying this kind of end type, you will get the possibility to declare a second recurrence, which will be valid for the next day in the simulation time.

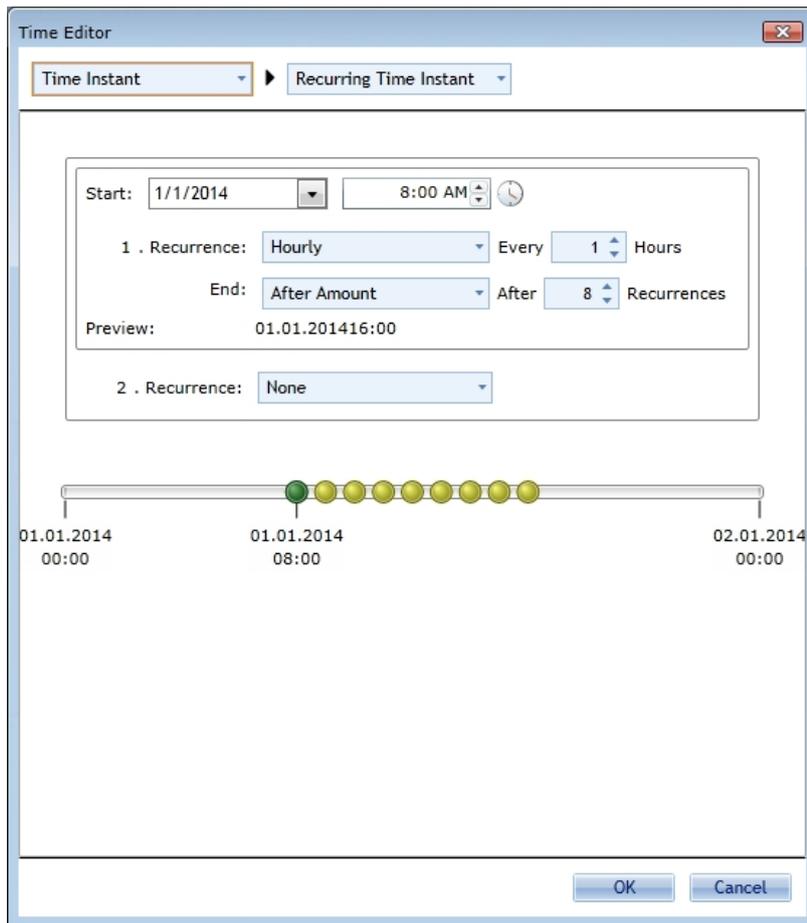


Figure 2: Time Instant - End Condition: After Amount

### End - By Time

By Time means that this recurrence will stop at the specified time. Here you can also declare a second recurrence.

When you select By Time and After Amount as the end type, it will always be possible to declare a subsequent recurrence.

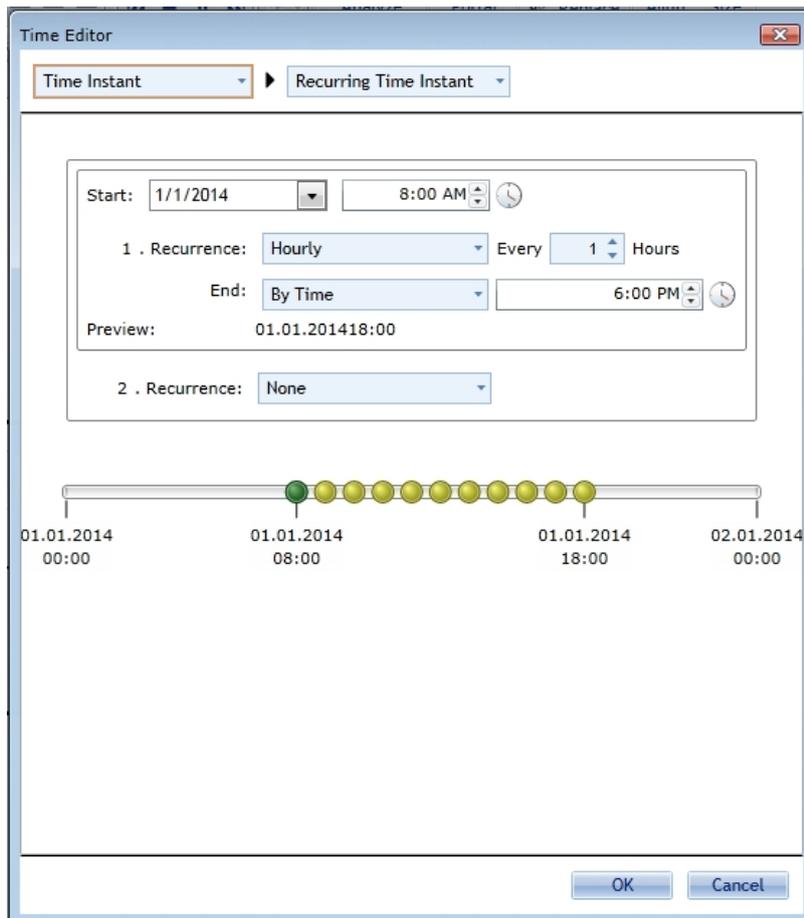


Figure 3: Time Instant - End Condition: By Time

### Scripted Event Trigger

## Scripted Time Instants

We also provide the possibility to script Time Instants in Python (a common scripting language). This is only used, if timing is very complicated, e.g. it depends on complex computations. The following simple statement "Result=SimulationContext.Time.GotoHour(9);" from the picture below tells the simulation that this event fires every day at 9 o'clock. Of course, this example is very trivial, and could be achieved more easily with [Time Instants](#). But maybe, you want to substitute the '9' with a variable, which is the result on an individual algorithm, you programmed in Python language.

See [Model 1d - Scripted Time Instant for Start Events](#) from the [Simulation Tutorial](#) for an example.

Also have a look at [Simulation Context Data](#) for an overview of the available data from the simulation context.

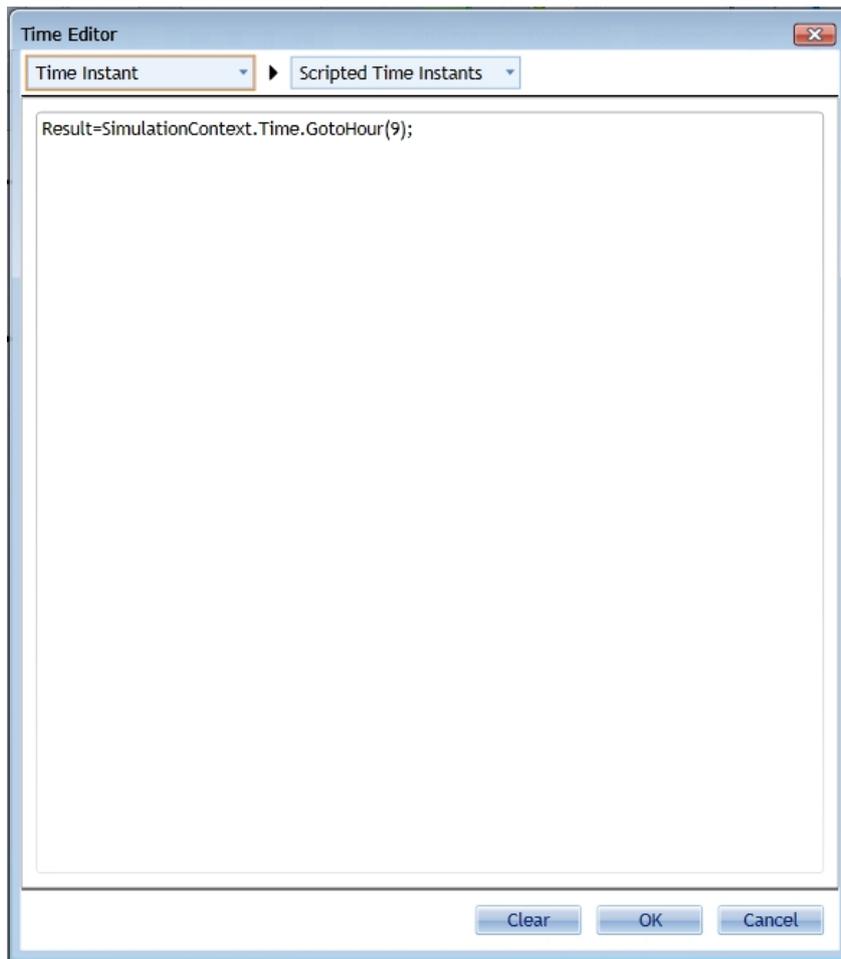


Figure 1: Time Editor in Python-mode

## Gateway Decision Making

You have to ensure that all exclusive and inclusive gateways have determinable conditions and/or probabilities.

In this chapter you will first be given a short introduction to the [gateways](#) that are supported by the simulation. To be able to make decisions we need a decision basis. In section [Process Variables](#) we will show you how to set up variables, which then will be used in conditions to determine the path after an exclusive or inclusive gateway. Finally you can find [example models](#) to test and deepen your knowledge about gateways.

## Content

- [Gateways](#)
- [Process Variables](#)
- [Using Probabilities](#)
- [Example Models](#)

## Gateways

Gateways do not contain any customizable simulation properties.

There are currently three gateway types available in the simulation, the exclusive, parallel and inclusive gateway.

- Exclusive and inclusive gateways need conditions on their outgoing sequence flows.

- These conditions are not set at the gateway itself but on the outgoing sequence flows.
- You have to set the property Expression of every outgoing sequence flow to obtain a simulation-ready model.
  - These expressions have to describe conditions, which depend on process variables, e.g. "x>10" or "var1==100".
  - See the following section [Process Variables](#) for more information on how to define variables and conditions.

## Exclusive Gateway

Merge: The exclusive gateway is simple. It fires whenever a token reaches it on any incoming sequence flow.

Fork: Every expression on the outgoing sequence flows will be verified, prior to choosing exactly one sequence flow as the successor. If there are two expressions that evaluate to true, the simulation will add an error entry to the error log and that explicit process instance will never finish. Such an error will lead to useless simulation reports.

## Parallel Gateway

Merge: Once a parallel gateway with more than one incoming sequence flows is reached, it will wait until each sequence flow delivers a token.

Fork: The parallel gateway's outgoing sequence flow do not possess any expression (if so, they will be ignored), so it passes a token to each of these.

## Inclusive Gateway

Merge: Once a token reaches an inclusive gateway with more than one incoming sequence flow, the gateway checks whether there are still tokens "alive", which can reach them. Have a look at the model below. Assume that the task "Stay Awake" lasts longer than "Work". In which case the token from "Work" reaches the inclusive gateway first. The inclusive gateway knows, that there is still one token at "Stay awake" which can reach it. So the gateway will have to wait. After "Stay awake" is finished, there is a decision ahead. If there is no need for a coffee, the token reaches the inclusive gateway, which now knows that there is no token alive anymore, so it fires. On the other hand, if there is a need for coffee, the inclusive gateway will be noticed and fires too.

Fork: Just like at the exclusive gateway, the expressions on all outgoing sequence flows are verified. But, contrary to the exclusive version, the inclusive gateway passes a token to every sequence flow, whose expression evaluated to true.

## Event-based Gateway

Fork: Every outgoing sequence flow has to be connected to an event. The process follows the path of the event which occurs first.

Merge: Does not exist in BPMN 2.0, because this does not make any sense. Use exclusive Gateway instead.

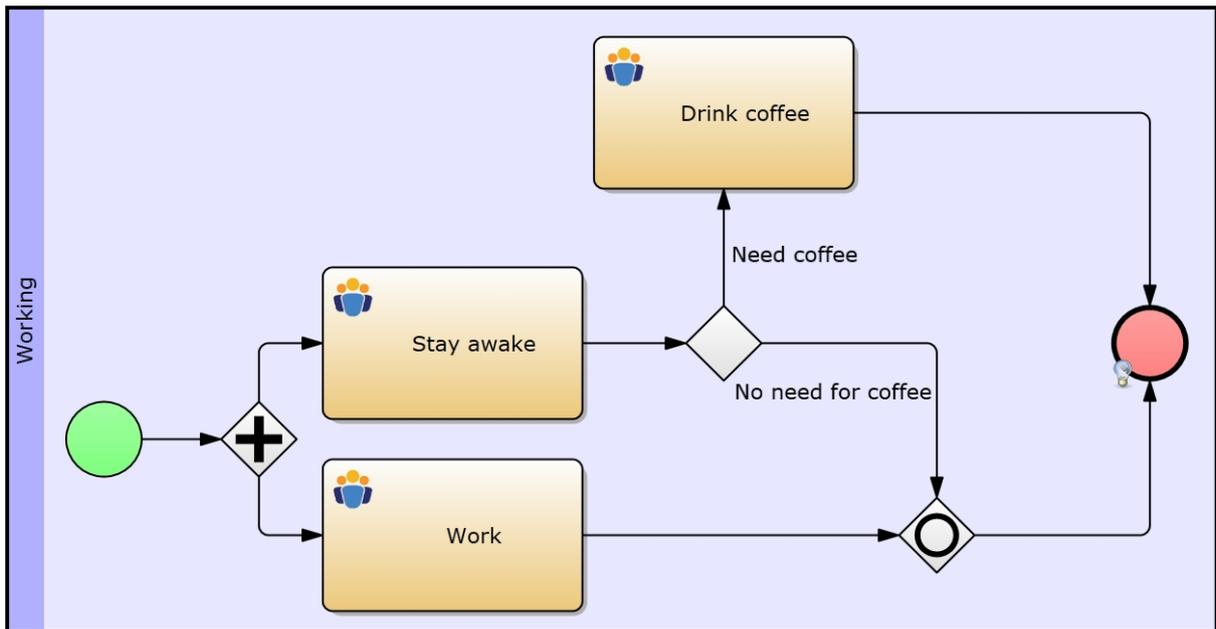


Figure 1: Example Model containing Exclusive, Inclusive and Parallel Gateways

## Process Variables

This section covers process variables, which are mostly used for the decision making at exclusive or inclusive gateways.

## Content

- [Assigning Process Variables](#): Learn how to define and afterwards manipulate variables. Also, get to know how to set up stochastic variables, which will give you the possibility to bring in some more dynamics into your models.
- [Assigning conditions](#): Learn how to define conditions at gateways to enable the decision making at gateways.
- [Scope of Variables](#): This section describes the visibility of variables throughout the simulation
- [Simulation Context Data](#): This section explains the usage of data, which is generated by the simulation, and which can be used in BPMN models' configurations.
- [Additional Information](#): Here you will find extra information regarding variables and their definitional languages.

## Assigning Process Variables

### Goals of this lesson

- Defining local variables
- Changing the values of already defined variables
- Defining Stochastic Attributes / Variables to increase your models dynamics and by that making it more realistic

### How-to

Model 3.a explains how to use exclusive gateways for a simulation. Contrary to parallel gateways a branching exclusive gateway needs a condition on every outgoing sequence flow depending on (a) process variable(s). During the simulation the conditions on the outgoing edges will be evaluated to determine exactly one edge as the next path element. You have to ensure that the conditions will determine exactly one sequence flow, otherwise (two or more edges meet the conditions) the simulation results will be useless due to this modeling fault.

But first we will learn how to setup variables and conditions.

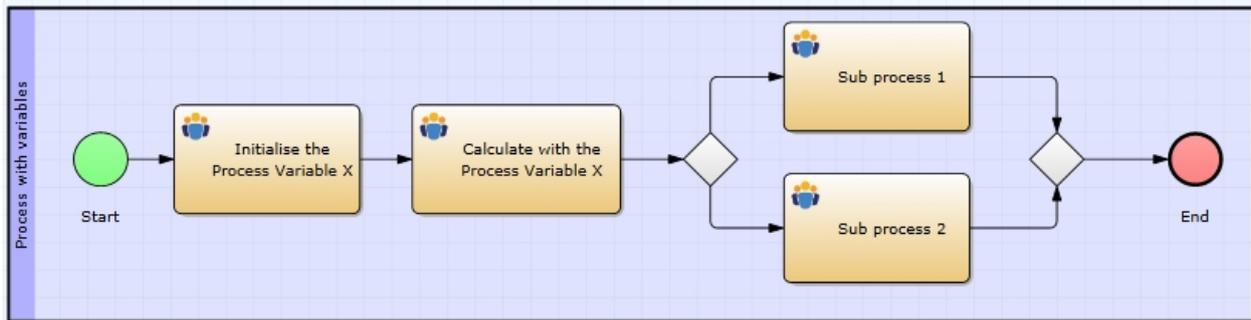


Figure 1: [Model 3a](#) from the [Simulation Tutorial](#)  
Approach

### Setting up variables

To set up conditions you need at least one process variable. We will create two variables called x and y to make the decision at e. g. an exclusive gateway:

- 1) Click any activity
- 2) Click the Edit button next to the Expression property under the Optional Simulation Properties of the activity
- 3) IYOPRO's Expression Editor will show up. Here you can define your variables.
- 4) To give an example we will define the variables x and y by putting the following lines into the expression editor (See Figure 2)
  - a)  $x=1$ ; which defines the variable x with a value of 1
  - b)  $y=1$ ; which defines the variable y with a value of 1
- 5) These variables are now known to the simulation

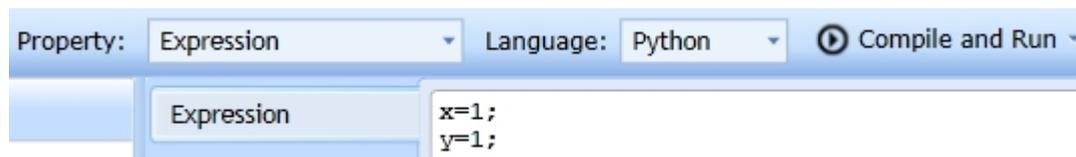


Figure 2: Expression Editor - Defining variables

### Changing the variables' values

Since our both variables contain the same value, we will alter them

- 1) Repeat the steps 1 to 3 for the second activity of the model depicted above
- 2) We will now increment x by one and define the new variable z that takes its value from y (See Figure 3)
  - a)  $x=x+1$ ; which increments x's value by 1
  - b)  $z=y$ ; which defines a new variable z that will be given y's value
- 3) Our variables have now these values:  $x = 2$ ,  $y = 1$ ,  $z = 1$

We can now use them in [logical conditions](#) to make decisions at exclusive or inclusive gateways

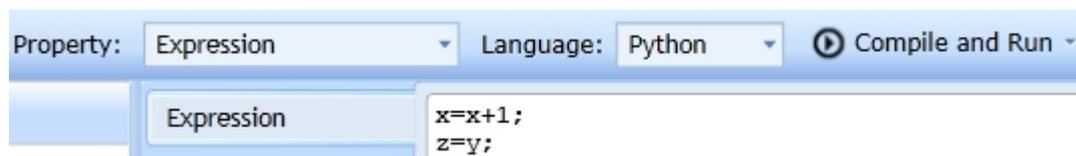


Figure 3: Expression Editor - Changing variables' values

### Creating Stochastic Attributes

Stochastic attributes will increase your model's dynamics and by that make it more realistic. Since they base on a [Stochastic Distribution](#), their values will always change (except for the Constant Distribution).

To use a stochastic variable, we need to define it at an activity:

1. Make sure, you selected the Simulation option from the Property Set drop-down-box
2. Click the activity

3. Click the button right to the Stochastic Attribute Assignment property under Optional Simulation Properties
4. The Process Attribute Editor shows up (see the picture below)
5. Click the Add button to add a new stochastic attribute
6. The Distribution Editor shows up. You will notice, that it is the same Distribution Editor that you already got to know by assigning durations of activities or inter-arrival times of start events.
7. Now e. g. select the normal distribution with a mean value of 10 and a standard deviation of 3
8. The variable will be given a name (something like "v1"). You can change this by double-clicking into the Name field. Rename it to X
9. You can use it in decision making at gateways just like the simple variables from above

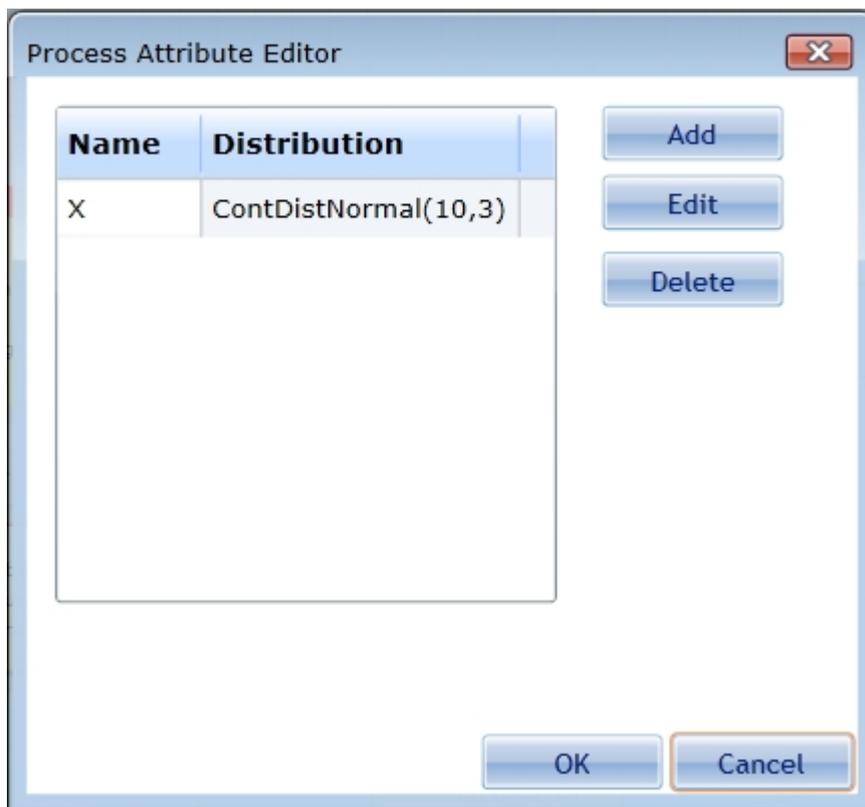


Figure 4: IYOPRO'S Simulation Process Attribute Editor

## Assigning Conditions

### Goals of this session

- Assignment of conditions for decision making at gateways

### How-to

Since we learned how to define variables in lesson [Assigning Process Variables](#), we now can define conditions at the exclusive gateway of [Model 3.a](#) (see the picture below). We will use the variables x, y and z from the [previous lesson](#).

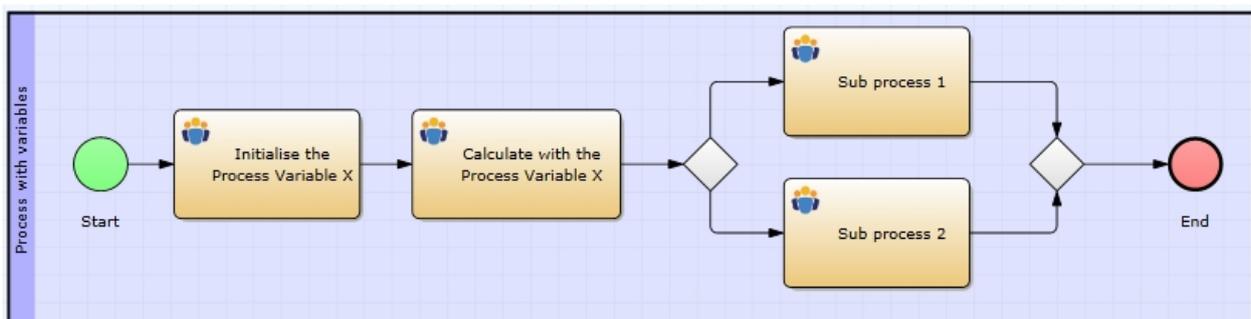


Figure 1: [Model 3a](#) from the [Simulation Tutorial](#)

## Approach

We will first define the condition of the upper sequence flow behind the branching (first) exclusive gateway:

- 1) Select the upper sequence flow behind the branching (first) exclusive gateway
- 2) Click the Edit button right of the Expression property under the Optional Simulation Properties
- 3) The Expression Editor shows up
- 4) We want to follow this sequence link when x is smaller than y. Therefore we enter the following expression: `Result=(x<y);`

Now we setup the condition for the bottom sequence link.

- 1) Repeat the steps 1 to 3 for the bottom sequence link
- 2) We want to follow this sequence link when x is greater than y. Therefore we enter the following expression: `Result=(x>y);`

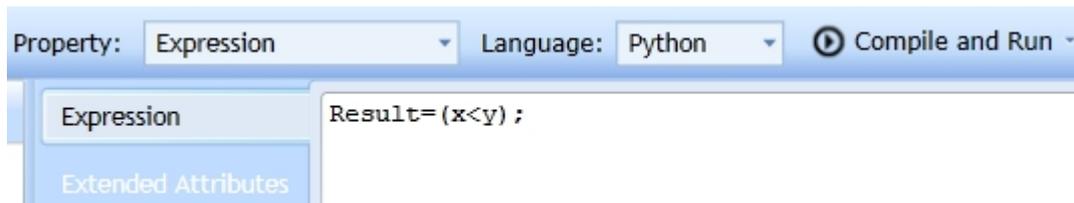


Figure 2: The condition's expression for the top sequence flow.

## Scope of Variables

### Goals of this session

- Visibility of variables - Which elements can access a defined variable?

### How-to

In this context the scope or visibility of a variable means its accessibility throughout the simulation.

- Variables are accessible / visible, when they are defined.
- The variables x and y from [Lesson: Assigning Process Variables](#) are visible from the first activity on throughout the whole process
- This means, if we would have defined them at any activity behind the exclusive gateway, they would not be visible at the gateway
  - So, the simulation would not have been able to make a decision based on those variables
- The naming of the variables is almost free, but still there are claimed or forbidden names, that cannot be used.
  - When using the Python language please refer to the [Python documentation](#) for claimed or forbidden names.
  - When using the F(x) language you will get a list of already claimed names below the expression field (Functions and Constants)

## Simulation Context Data

We provide some specific data access from the simulation context to allow you to enhance your simulation configuration.

- The data provided is accessible in **Python** scripts only
- Use the Namespace **SimulationContext** followed by a dot and the data qualifier, which you want to access
- See the table below for a list of data that is accessible in **Python** scripts

<i>SimulationContext.Time</i>	Represents the simulation time as System.DateTime (.NET type). Can be used as getter in all python scripts.  You can use the following methods:
-------------------------------	---

	<ul style="list-style-type: none"> <li>• <code>SimulationContext.Time.Year</code></li> <li>• <code>SimulationContext.Time.Month</code></li> <li>• <code>SimulationContext.Time.Day</code></li> <li>• <code>SimulationContext.Time.Hour</code></li> <li>• <code>SimulationContext.Time.Minute</code></li> <li>• <code>SimulationContext.Time.Second</code></li> <li>• <code>SimulationContext.Time.Millisecond</code></li> <li>• <code>SimulationContext.Time.DayOfWeek</code></li> <li>• <code>SimulationContext.Time.DayOfYear</code></li> </ul> <p><b>Example usage 1 as gateway condition:</b>  <code>SimulationContext.Time.Hour&gt;12</code>  will check, whether current time is 13:00 or later.</p> <p><b>Example usage 2 as gateway condition:</b>  <code>SimulationContext.Time.DayOfWeek=='Friday'</code>  will check, whether current day is Friday.</p> <p>At time-based events with scripted inter-arrival time, you can use operations like:</p> <ul style="list-style-type: none"> <li>• <code>GotoHour(x)</code>: jump to hour x (with x being an integer between 0 and 23)</li> <li>• <code>GotoMinute(x)</code>: jump to minute x (with x being an integer between 0 and 59)</li> </ul>
<code>SimulationContext.Resources</code>	Provides access to the current state of resources. Can only be used as getter.  You can use the following methods: <ul style="list-style-type: none"> <li>• <code>SimulationContext.Resources.IsIdle(ROLE_STRING)</code></li> <li>• <code>SimulationContext.Resources.Count(ROLE_STRING)</code></li> <li>• <code>SimulationContext.Resources.CountIdle(ROLE_STRING)</code></li> <li>• <code>SimulationContext.Resources.CountBusy(ROLE_STRING)</code></li> </ul> <p><b>Example usage 1 as gateway condition:</b>  <code>SimulationContext.Resources.IsIdle('executive officer')</code>  will check, whether at least one 'executive officer' is free.</p> <p><b>Example usage 2 as gateway condition:</b>  <code>SimulationContext.Resources.CountIdle('executive officer')&gt;=3</code>  will check, whether at least three 'executive officer' are free.</p>
<code>SimulationContext.Product</code>	Represents the current <a href="#">product</a> as a string. Can be used as getter or setter.  <p><b>Example usage as gateway condition:</b>  <code>SimulationContext.Product=='standard product'</code>  will check, whether the current product is called 'standard product'.</p> <p><b>Example usage as activity expression:</b>  <code>SimulationContext.Product='standard product'</code>  will set the current product to 'standard product'.</p>

Additional Information

## Differences between the languages Python and F(x)

- When using the f(x) language a semicolon at the end of a line means an assignment (the variable now holds the given value).
  - Without the semicolon the line will be seen as an expression, that will be evaluated to a result by the under-lying rule-engine.
- If you are using the Python language, then you will always need a semicolon at the end of the line (or expression).
  - A result statement in Python always looks like "Result=(EXPRESSION);"
- See the figures 1 and 2 for the same expression
  - Figure 1 in Python
  - Figure 2 in F(x)

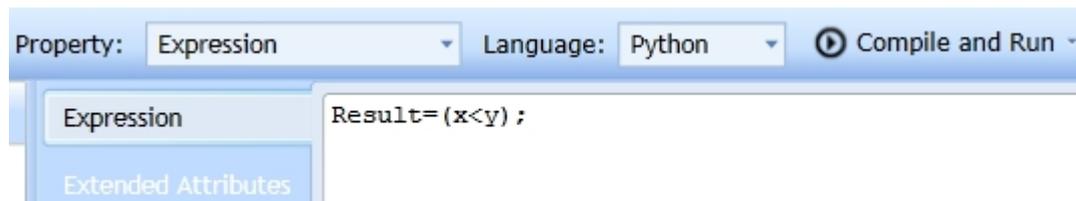


Figure 1: A condition's expression in Python language

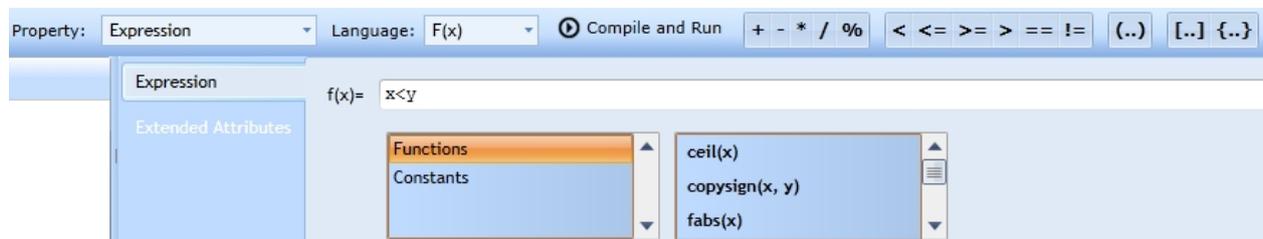


Figure 2: The same condition's expression in F(x)-language

## Difference between exclusive and inclusive gateways

- In contrast to the parallel gateways an exclusive merging gateway does not wait for sub-processes to reach the gateway.
- Instead the gateway will operate as soon as a sub-process (token) reaches it.
- An exclusive gateway will always create only one sub-process (token).
  - (Still, the BPMN allows valid models with multiple sub-processes [tokens] reaching an exclusive merging gateway, which then will forward each sub-process without delay.)

### Note

Default sequence flows do not need an expression / condition. A default sequence flow will always be determined as the next path element, if none of the other edge's conditions evaluate to true.

## Using Probabilities

### Goals of this lesson

- Learn another way to configure the Gateways' decision making

Basing the gateways' decision making on the process variables is an excellent approach. Though, often it is not necessary to undergo such an effort. E. g. if you know that "Edge 1" will be pursued in 40% of the cases and "Edge 2" in 60% of the cases, it will be unnecessary and time-consuming to define a process variable and the conditions at the edges. Therefore we offer you the possibility to simply put weights (or probabilities) on each edge at the exclusive and inclusive gateway.

### How-to

- Click on a sequence flow at a gateway (and make sure, you selected Simulation from the Property Set)
- You will now find the Probability property under the Optional Simulation Properties (See figure 1)

- Click into the text field and put in a floating-point number between 0 and 1
  - 0 stands for always false, which means that the sequence flow will never be fired
  - 1 stands for always true, which means that the sequence flow will always be fired
  - A value of (e. g.) 0.5 indicates that there is a chance of 50% that the sequence flow will be pursued

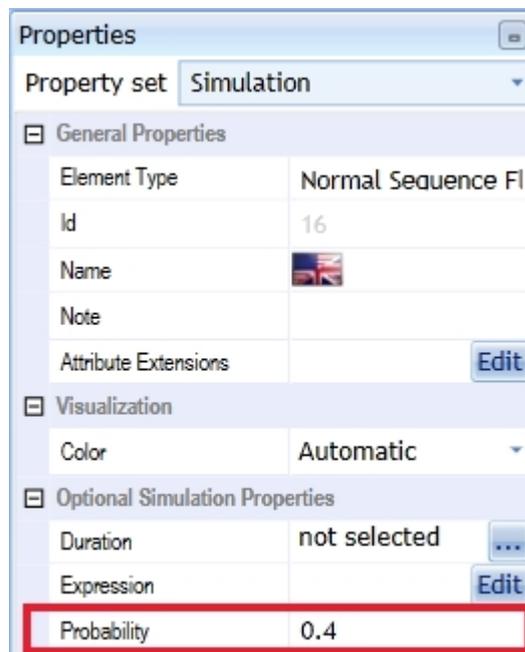


Figure 1: The Simulation Properties of a Sequence Flow - The Probability Property highlighted

### Exclusive Gateways

- At this gateway the sum of all probabilities assigned to the outgoing sequence flows must equal 1
- If this is not the case, the simulation will produce errors, which will then be shown in the report

### Inclusive Gateways

- At the inclusive gateway the sum of all probabilities assigned to the outgoing sequence flows does not matter

### Example Models

Please see the Models [2 \(parallel\)](#), [3a \(exclusive\)](#) or [Model 3b \(inclusive\)](#) from the [Simulation Tutorial](#) for examples.

### Element settings

You can find a complete description of possible simulation settings for the available BPMN 2.0 elements [here](#).

- [Events](#) - Events settings.
  - [Linking events to each other](#) - Explains how to link sending and receiving events
  - [Start Events](#) - Describes the start events' simulation properties
  - [Intermediate Events](#) - Describes the intermediate events' simulation properties
  - [End Events](#) - Describes the end events' simulation properties
- [Activities](#) - A complete list of possible simulation settings for activities.
- [Flows](#) - Sequence, Message and Association flows
  - [Sequence Flows](#)
  - [Message Flows](#)

- [Default & Conditional Flows](#)
- [Data Associations](#)
- [Pools and Lanes](#) - Settings for pools and lanes
- [Data and Artifacts](#) - How to use artefacts in simulation

## Events

In the following we will describe the events' simulation properties.

### Content

- [Linking Events to each other](#): Learn how to link events to each other by giving them the same name
- [Start Events](#): Describes the start events' simulation properties
- [Intermediate Events](#): Describes the intermediate events' simulation properties
- [End Events](#): Describes the end events' simulation properties

### Linking Events to Each Other

To simulate BPMN 2.0 models, we need to know, which sending event(s) trigger(s) which receiving event(s). To accomplish this, we need to give events that shall be linked to each other the same name.

### Approach

Imagine you create a model that contains two intermediate sending message events and two intermediate receiving message events. Then there are two possible ways, to identify associated (or linked) events:

1. You connect the events via message flows.
  - a) (This is not possible for other event types, therefore we also offer a second option, see below)
2. You do not want to add anymore flows to your model, because they would reduce your model's clarity.
  - a) Therefore we offer you the possibility, to avoid using message flows between sending and receiving message events by simply giving them the same name.

Please see Models [4c \(Messages without an explicit message flow\)](#), [5 \(sending Signals to Multiple Process Instances\)](#) and [6 \(Triggering Start Events\)](#) from the [Simulation Tutorial](#) for examples to this concept.

### Limitations

There are some limitations to this concept:

- Pay attention to the events' relations. E. g. one sending message event can trigger only one receiving message event, 1:1 relation. But one sending signal event can trigger more than just one receiving signal event, 1:n relation.
- Receiving events without a label (name) will catch all triggered sending events of the same type depending on the relations (see bulleting above).
- If the model contains one receiving and one sending event of the same type, it will link these two events to each other.
- It is possible to model n:1 relations (always depending on the BPMN 2.0 Specification's constraints. E. g. one receiving message event can be triggered by n sending message events).

### Start Events

A process is started by a start event, which can be either general or depending on time or on other facts (receiving a message, signal etc.). Since the latter cases always involve sending events, we only need to add simulation properties to the general and to the timer start event.

To start the process we need to add some information. We can either select one of the

- stochastic distributions (see [Stochastic Event Trigger](#) or [Model 1a](#) from the [Simulation Tutorial](#) for examples)
- or recurrent time instants from the time editor to determine the process' inter-arrival time (see [Calendar Based Event Trigger](#) or [Model 1c](#) from the [Simulation Tutorial](#) for examples)
- or we can write a Python-script (Scripted delay). See [Scripted Event Trigger](#) or [Model 1d](#) from the

[Simulation Tutorial](#) for examples

Attention: Contrary to the conceptual idea of the message receiving start event, we allow you to specify an inter-arrival time to represent the duration until the message receipt. This is occasionally necessary, e. g. if you do not want to model a black box pool, which sends the messages to this specific start event, and which solely exists to start this event.

## Simulation Properties of the General and the Timer Start Event

Optional Simulation Properties	
Inter-arrival Time	Select a stochastic distribution (which represents your process' inter-arrival time) or use <a href="#">Calendar based</a> or <a href="#">Script based Event Triggers</a> to set up the inter-arrival times. <ul style="list-style-type: none"><li>• See <a href="#">Assigning Inter-arrival Time / Duration</a> for more information.</li><li>• Or see <a href="#">Chapter 1 (Processes and Sequence Flow)</a> from the <a href="#">Simulation Tutorial</a> for an example.</li></ul>

## Simulation Properties of the Message receiving Start Event

Optional Simulation Properties	
Stochastic Event Trigger	Check this box, if you want to specify a distribution for a time span which will then represent the delay of the message receipt.
Must-Have Simulation Properties	
Duration until Message Receipt	This property will become obligatory after checking the Stochastic Event Trigger checkbox. Here you can specify a distribution, which will then represent the duration until the message will be received.

## Intermediate Events

Events generally do not possess any simulation properties. All of them use the inherent information based on the BPMN 2.0 specification.

However there are two exceptions:

- Firstly timer intermediate event: It basically does the same job as the general / timer start event. The only difference is that it can be placed "in" the flow or it can be attached to an activity.
- Secondly the message receiving intermediate event: In some occasions you will simply want to create a model, which uses the conceptual idea of a message receiving event, but which does not necessarily have a respective sender event in another pool. Therefore we offer you the possibility to specify an optional duration to represent the time consumption until the message receipt usually occurs.

Note that if you do not specify one of the below properties, the simulation will execute the event immediately when reaching the activity.

## Simulation Properties of the Timer Intermediate Event

Optional Simulation Properties	
Inter-arrival time	Specify a distribution (for a time span), a recurrent time instant or a script to represent this event's trigger delay. <ul style="list-style-type: none"><li>• This approach is equivalent to the stochastic time operations at <a href="#">Start Events</a>.</li><li>• See <a href="#">Model 12 - Scripted Delay</a> from the <a href="#">Simulation Tutorial</a> for an example.</li></ul>

## Simulation Properties of the Message receiving Intermediate Event

Optional Simulation Properties	
--------------------------------	--

Stochastic Event Trigger	Check this box, if you want to specify a distribution for a time span which will then represent the delay of the message receipt.
Must-Have Simulation Properties	
Duration until Message Receipt	This property will become obligatory after checking the "Stochastic Event Trigger" check-box. Here you can specify a distribution, which will then represent the duration until the message will be received.

## End Events

End events do not contain any simulation properties. They are used to terminate a process and to trigger start events if they are linked to those.

The number of occurrences of end events is counted and provided in simulation [results](#).

## Activities

Activities are the only BPMN entities that actively consume time. Therefore the activity's duration is the most important simulation property.

Once you click the button next to the Duration label, you can select one of the stochastic distributions described in the [Distributions](#) topic with help of the [Time Editor](#).

Several simulation properties will only be unlocked, if you add some optional Activity Properties. Currently there are three options, which are listed in the table below. By selecting one of these properties, an additional input box will appear in the Must-Have Simulation Properties area.

## Simulation Properties of Activities

Activity Properties	
Looping	Determines whether this activity is looping. This means it will be executed n times until a certain condition is being evaluated to true.
Multiple Instances: Parallel	Determines whether this activity is being executed in n parallel instances.
Multiple Instances: Sequential	Determines whether this activity is being executed n times in a row.
Must-Have Simulation Properties	
Duration	The duration of the activity. You need to select one of the stochastic distributions and parametrize it as adequately as possible for your process.
Looping Condition	<p>Defines the condition to break the loop. As long as the condition is evaluated to false, the activity will be started again.</p> <ul style="list-style-type: none"> <li>• When the expression's result is true, the loop will break.</li> <li>• You can either write a simple mathematical term like "x&gt;5;" (Language-Selection: F(x)). x must be defined somewhere else before) or you can formulate a complex Python script. <ul style="list-style-type: none"> <li>○ See Process Variables or the Models <a href="#">7a</a>, <a href="#">7b</a> and <a href="#">7c</a> from the <a href="#">Simulation Tutorial</a> for more information and examples</li> </ul> </li> </ul>
Number of Parallel Instances	This expression defines the number of parallel instances of the activity. The approach is similar to the Looping Condition. The sole difference is that you can simply put a number into this box, which indicates the number of parallel instances. Another example is "x + 1", though x must be defined or altered somewhere else.
Number of Iterations	This expression defines the number of iterations of the activity. The approach here matches with the Number of Parallel Instances.
Optional Simulation Properties	

Expression	In this text box you can define custom attributes (e. g. "x=5"), or operations that shall be executed in this activity (e. g. a Python-script).
Stochastic Attribute Assignment	In addition to Expression field, this option gives you the possibility to use one of our stochastic distributions to calculate the value of an attribute. <ul style="list-style-type: none"> <li>• E. g. "x" gets its value from the Normal Distribution with the parameters: <ul style="list-style-type: none"> <li>○ Mean-Value = 4,</li> <li>○ Standard-Deviation = 1</li> </ul> </li> </ul>
Participant	Assign participants to this pool or lane. See topic Organization Diagram for more information.
Inherit Resources	This Check-Box determines whether this activity shall inherit its superior elements' resource demands. E. g. the activity's Pool demands n entities of resource x. If you check this property, the activity will also demand n entities of resource x. See <a href="#">Resources</a> topic for more information on the resources concept.
Resource Priority	This property represents the priority of this activity in the resource allocation algorithm. <ul style="list-style-type: none"> <li>• The higher the priority is set the faster this activity (or the process instance which is currently working off this activity) will be provided with the needed resources.</li> <li>• This option gives you the possibility to prioritize your activities in the resource allocation, because in reality there are often tasks that are more important than other tasks.</li> </ul>
Report Properties	
Generate Box-Plot	Gives you the option to generate a Box-Plot diagram for this activity. Attention: <ul style="list-style-type: none"> <li>• These diagram types are very expensive in relation to the simulation time advancement.</li> <li>• The more Box-Plots you want to be generated the slower the calculation period of the simulation becomes.</li> </ul>

## Subprocess Activities

The subprocess activity also contains a property called Use Stochastic Duration. This is useful when you have not yet created the subprocess model (or did not specify any simulation specific data in that subprocess model) but still expect valid simulation results.

Subprocess Activity Properties	
Use Stochastic Duration	Determines whether this subprocess activity shall use the Stochastic Duration, which can be assigned in its Duration property. By enabling this property, the underlying model to the subprocess activity will not be simulated. It will be ignored and the simulation replaces the execution of the model with a stochastic distribution, which then represents the time that is needed to execute the subprocess model.

## Flows

This topic covers all the different BPMN flow types.

### Content

- [Sequence Flows](#)
- [Message Flows](#)
- [Default & Conditional Flows](#)
- [Data Associations](#)

### Sequence Flows

In contrast to the BPMN 2.0 Specification we added one feature, that is mostly desired by non-technical modelers: the Duration of a sequence flow.

- This can be useful when you need a delay between two activities.
- E. g. you model a process like the one shown below. Imagine the way up to the board is long and takes around ten minutes.
- You do not want to add another activity which represents the ten minutes time consumption for the walk to the board room.
- Instead you simply add a Duration in the simulation properties set.
- See [Model 1b - Time Consuming Sequence Flows](#) from the [Simulation Tutorial](#) for an example.

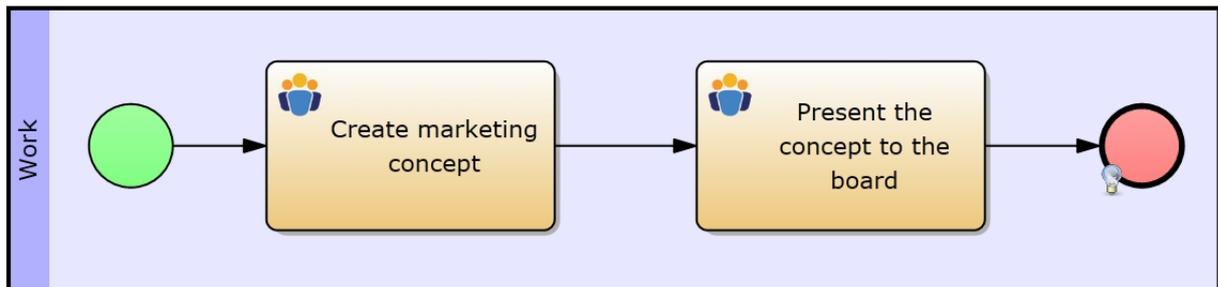


Figure 1: Example Model

## Simulation Properties

Optional Simulation Properties	
Expression	<p>This field is used to define the expressions after an exclusive or inclusive gateway.</p> <ul style="list-style-type: none"> <li>• You simply add a condition like "x&lt;y" (in F(x)) or a Python-script, that determines whether this sequence or conditional flow is to be followed next in the model.</li> <li>• See <a href="#">Gateway Decision Making</a> and especially the sub-section <a href="#">Assigning Conditions</a> for more information.</li> </ul>
Duration	<p>Add a duration for this sequence flow.</p> <ul style="list-style-type: none"> <li>• The approach is exactly the same as the duration of the activity.</li> <li>• This feature can be used to represent a time consumption, for which you would not like to add an additional activity, like the time period of walking to the CEO's office (which can be significantly far way).             <ul style="list-style-type: none"> <li>○ See <a href="#">Model 1b - Time Consuming Sequence Flows</a> from the <a href="#">Simulation Tutorial</a> for an example.</li> </ul> </li> </ul>

## Message Flows

Message flows are used to represent communication between pools. That's why we added some useful features which will give you the possibility to get more details into your communication scheme (see table below).

## Simulation Properties

Optional Simulation Properties
--------------------------------

Data	<p>In this Text-Box you can add attributes, which shall be transported along with the message.</p> <ul style="list-style-type: none"> <li>• E. g. you define the variable x in activity a in pool 1, that shall be passed to an activity b in pool 2, because it will be the data basis for a gateway decision after activity b. <ul style="list-style-type: none"> <li>○ To accomplish this, you simply write the name of variable (in this case: x) into the Data-Text-Box and it's done!</li> <li>○ If you would like to pass more than one variable or attribute, you can do this by separating them with a semicolon (e. g. "x; y;")</li> </ul> </li> <li>• See <a href="#">Model 4b - Transporting Process Variables via Messages</a> from the <a href="#">Simulation Tutorial</a> for an example.</li> </ul> <p>Note that you can assign new names to the attributes you want to be transferred to another pool.</p> <ul style="list-style-type: none"> <li>• The keyword to do this is "AS".</li> <li>• E. g. you want the attributes "revenue" and "profit" to be transferred from your food division to your enterprise's headquarters. <ul style="list-style-type: none"> <li>○ Then you simply enter the following: "revenue AS revenueFood; profit AS profitFood"</li> <li>○ Of course you can mix "to be renamed" and "not to be renamed" attributes.</li> </ul> </li> <li>• <a href="#">Model 3e - Data Stores</a> from the <a href="#">Simulation Tutorial</a> uses the keyword AS to rename transferred data between pools for example.</li> </ul>
Transport Delay	<p>Here you can add a transport delay by selecting an appropriate stochastic distribution.</p> <p>E. g. sending a document to your supplier, which will be delivered by a courier in approximately one hour.</p>

### Default & Conditional Flows

You can use these elements in simulation without any additional settings. These flow types do not contain any customizable simulation properties.

### Data Associations

Data Associations are used to link Data Objects and Data Stores to the BPMN model's entities. E. g. if you want an activity to access the data of a previously created data object, then you will use a data association between the activity and the data object. Depending on the data associations direction, the activity will either read data from the data object or write data to the data object.

When using data associations with data stores, you have to specify the data, which is read or written by any model's entity. This is being done in the Data property.

Optional Simulation Properties	
Data	<p>This property is used to determine the data, which will be read or written from or to the data store. Simply insert the name of the data entries. E. g. "x;" will transfer a variable called x to or from the data store.</p> <p>Use a comma if you need to pass more than just one variable. F. e. "x, y;"</p> <p>You can also rename the variables with the "AS" identifier.</p> <ul style="list-style-type: none"> <li>• Imagine two pools using the same data store. <ul style="list-style-type: none"> <li>○ The data entity x is available to both pools.</li> <li>○ But due to some constraints it is necessary that pool A calls the variable y</li> </ul> </li> </ul>

	<p>instead of x, and pool B calls it z.</p> <ul style="list-style-type: none"> <li>○ So, the data property of pool A's data associations looks like this: <ul style="list-style-type: none"> <li>▪ "x AS y;" (reading)</li> <li>▪ "y AS x;" (writing)</li> </ul> </li> <li>○ and those from pool B are: <ul style="list-style-type: none"> <li>▪ "x AS z;" (reading)</li> <li>▪ "z AS x;" (writing).</li> </ul> </li> </ul> <ul style="list-style-type: none"> <li>● <a href="#">Model 3e - Data Stores</a> from the <a href="#">Simulation Tutorial</a> uses the keyword AS to rename transferred data between pools for example.</li> </ul> <p>The comma separator also works for the renaming functions: E. g. "x AS a, y AS b;"</p>
--	---

## Pools & Lanes

Pools and Lanes generally only possess the inherent information based on the BPMN 2.0 specification.

You can assign one (or more) resource(s) to the Participant property of a pool or lane. All tasks lying in that pool or lane will then by default adopt the need for the resources, that have been assigned to its superior element (lane or pool).

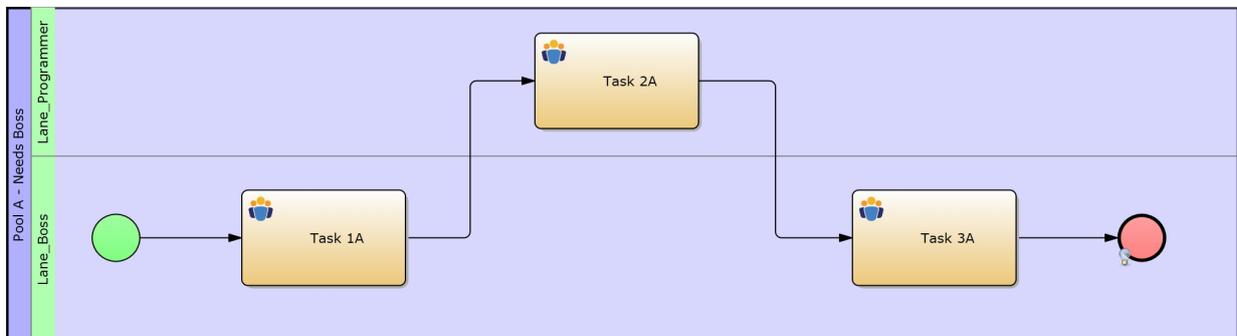


Figure 1: Example Model

- In this model we assigned the participant Boss to the pool, which means that every user task in the pool requires this particular resource.
- Further we assigned two Programmers to the Lane "Lane\_Programmer", which indicates that we need two Programmers at user task "Task 2A".

Note that:

- Resources inherit the resource needs from its superior elements (pools and lanes) by default. You will have to uncheck the checkbox Inherit Resources in the simulation properties of an activity, if you do not want this specific activity to derive its resource needs from the superior elements.

## Simulation Properties of Lanes and Pools

Optional Simulation Properties	
Participant	Assign participants to this pool or lane. See topic Organization Diagram for more information.

## Data and Artifacts

This section covers the simulation settings of the elements:

- [Data Object](#)
- [Data Store](#)

## Data Object

Data Objects are used as Data Input or Output. You can stick them to activities, and depending on the Association flow's direction, the Data Objects act either as an input (association flow in) or as an output (association flow out), or both.

## Simulation Properties

Optional Simulation Properties	
Expression	In this text box you can define custom attributes (e. g "x=5"), or operations that shall be executed in this activity (e. g. a Python-script) .  See <a href="#">Model 3d - Data Objects</a> from the <a href="#">Simulation Tutorial</a> for an example.
Stochastic Attribute Assignment	In addition to Expression field, this option gives you the possibility to use one of our stochastic distributions to calculate the value of an attribute <ul style="list-style-type: none"> <li>• E. g. x gets its value form the Normal Distribution with the parameters: Mean-Value = 4, Standard-Deviation.= 1</li> <li>• See <a href="#">Model 3c - Inclusive Gateways</a> from the <a href="#">Simulation Tutorial</a> or <a href="#">Assigning Process Variables</a> for examples.</li> </ul>

## Data Store

The data store holds data, which is being written to it by the model's entities. Once a data entry is present in the data store, it can be read and manipulated.

- Use Data Associations to write / read to / from a data store
- See Data Associations for more information on transferring data
- Or see [Model 3e - Data Stores](#) from the [Simulation Tutorial](#) for an example.

## Advanced Concepts

In this section we will introduce advanced concepts, which contribute to a more realistic simulation configuration and for this reason increase the benefit of the experiment reports.

### Content

- [Resources](#): Describes how to use resources, and what prerequisites have to be considered to achieve sufficient results.
- [Costing](#): Learn how to set up the necessary settings to ...

### Resources

Resources in IYOPRO are used to map real life scarce entities to the BPMN model. A resource can be a human, a machine or something else that you consider to be a resource.

A Participant, that is part of the execution of an activity, can be assigned to every activity. Although the Participant normally represents a human resource, you are free to expand the term to machines, storage capacities, and so on.

### Prerequisites

If you want to assign resources to your models, you need to define them in organization diagrams . More information can be found in this IYOPRO documentation.

The raw process to create a model with resources is as follows:

- 1) Create a BPI-Project
- 2) Create the organization diagram like the one from Figure 1
- 3) Add a collaboration diagram
- 4) Import the roles at any activity or pool / lane from the organization diagram
- 5) Now you can chose the roles at every activity or pool / lane
- 6) Examples: Please refer to [Chapter 7](#) of the [Simulation Tutorial](#) for a how-to on Resources.
- 7) You can [access the current state of resources within python scripts](#).

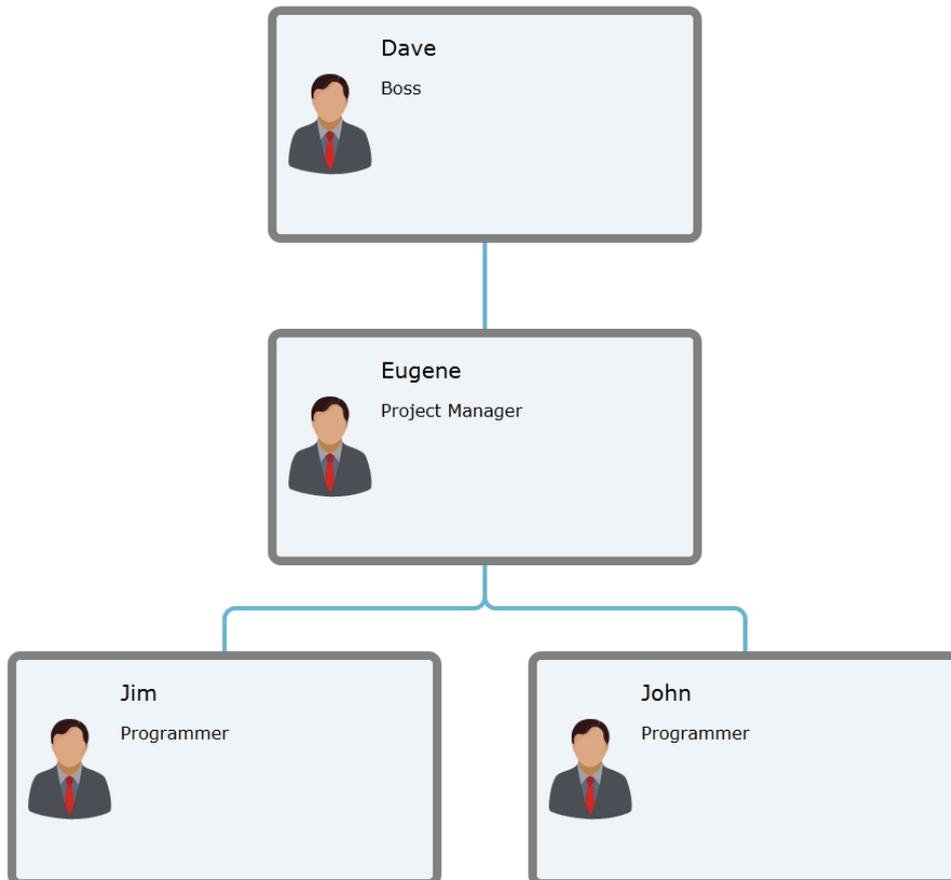


Figure 1: The Organization diagram that will be used for the following lessons

## Costing

Combining process simulation with cost accounting permits the assignment of costs to processes and/or products. The [result](#) is a table-based costing overview, where row headers represent the types of cost and column headers represent processes or cost-driving products.

Using cost accounting within simulation requires at least three steps:

- [Declaring cost types within organization diagrams](#)
- [Quantifying fixed costs within organization diagrams](#)
- [Quantifying variable costs within collaboration diagrams](#)

The activity-based mapping of resource costs (e.g. human workers or machines) to processes or products requires additional steps:

- [Allocating resources to user activities](#)
- [Determining fixed or variable resource costs](#)

If you define products, you can [map the costs on these products](#). Also the process behaviour may depend on current products.

The [result](#) section explains the expected result table, in particular the meaning of the different rows.

## Declaring cost types

### Declaration of cost types

All cost centers have to be declared within [organization diagrams](#), which must be part of your BPI Project. If you don't know how to create organisation diagrams see [BPI Project](#) section.

Figure 1 shows an organization diagram with three cost centers declared: 'FEK', 'MEK' and 'VerwGK'. Figure 2 shows the property editor for the diagram elements.

Proceed the following steps for declaring a new cost center:

1. Create or open an organisation diagram.
2. Create a new element using the big arrows inside the editor view
3. Change the element type to 'Cost center' (Figure 2, Step 1)
4. Label the cost center with a name (Figure 2, Step 2)
5. Decide whether it is driver-independent (fixed) cost type or an driver-dependant (variable) cost type (Figure 2, Step 3)

If the cost-center is driver-neutral you have to set the [fixed costs](#) within the organization diagram. If it is driver-dependent you have to assign [variable costs](#) within collaboration diagrams. The hierarchical order of cost centers inside the organigram doesn't have a meaning till now, this will change in future releases.

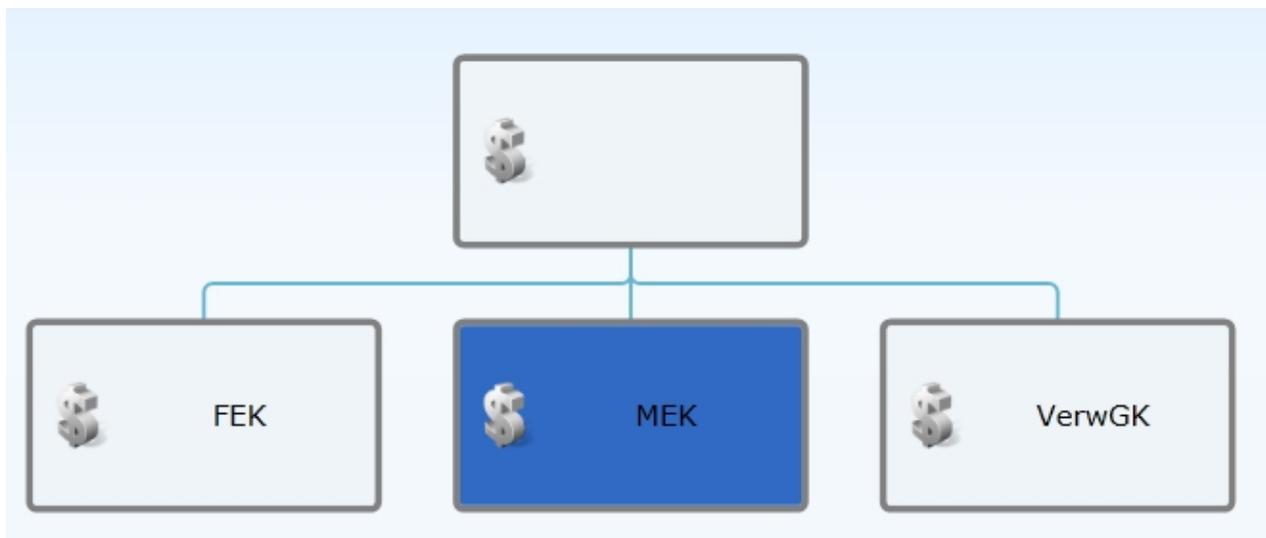


Figure 1: Cost centers declared within an organization diagram

Property set **Modeling**

General Properties	
Element Type	1 Cost Center
Id	10
Note	
Name	2 MEK
Attribute Extensions	<a href="#">Edit</a>
Layout Properties	
Alignment	Center
Cost Center Properties	
Cost Type	3 Driver Dependant Costs

Figure 2: Properties of the diagram element 'cost center'

## Neutral costs

### General settings for driver neutral and fixed costs

Fixed costs are driver-neutral costs, which means they are independent from the number of executed processes. E.g. your 'office rent,' which is 1200 Euro per month as shown in figure 3. After you have [declared](#) your driver neutral cost center, you have to set the following parameters via the property editor. Be sure that the 'Property set' is 'Simulation' if

1. The time period for exact one accounting (months, days or hours),
2. the unit of measurement (usually EUROS or Dollars),
3. and - of course - the amount to account.

Properties

Property set **Simulation**

General Properties	
Element Type	Cost Center
Id	10
Name	 rent
Note	
Attribute Extensions	<a href="#">Edit</a>
Visualization	
Color	Automatic
Must-Have Simulation Parameters	
Cost Type	Driver Neutral Costs
Costs	1200
Time Unit	Month
Unit	Euro

Figure 3: Properties to be set for driver neutral costs

### Allocation techniques for fixed costs

The total amount appearing in your [result table](#) depends on parameters given by you and the length of the

simulation run. There are two possibilities for segmenting the total amount to your processes and products:

1. If you don't make any extra settings, a well-fitting surcharge rate will be used, so that more expensive processes even get a larger part your costs. This is the classical way, which is a good choice for costs which doesn't have any relation to any of your cost drivers, e.g. your office rate. Nevertheless, it's not the best choice for every cost type.
2. An alternative is the activity-based distribution of your fixed costs. It is automatically activated, if you assign the cost-center to activities within your collaboration diagrams ([see next section](#)). Even in this case, the total amount stays the same, it's just about allocating these costs! E.g. if you are using a manufacturing machine, which generates a fixed rate of leasing costs every month, you may want to assign these costs just to manufacturing processes with an emphasis in the processes which were executed more often.

## Activity-based costs

### Activity-based costing

After you have [declared](#) your driver-dependent cost centers, you have to assign them to activities in your collaboration diagrams, which means that every time the activity is executed, costs are generated. Select an activity within your diagram, be sure to use the simulation property set, and click on 'Activity based costing' (figure 4) for opening an extra window, which will allow you to configure the individual costing.

First, you have to choose the corresponding cost center from the drop-down box to the left (see figure 5). If the drop-down box is empty, be sure that you have declared your cost centers within the same BPI projekt.

Next, you have to choose between three types of costing indicators:

- Flow-dependent means that the value in the column 'cost' will be booked one time per execution of one activity.
- Time-dependent means that the amount to be booked depends on the duration of the activity. The value in the column 'cost' will be multiplied with the individual activity duration.
- Property-dependent means that the value in column cost is multiplied with the value of the given process variable, which may be individually modified in python scripts.

You only can set a value in the 'Costs'-Column if the selected cost center is declared as driver-dependent. If you select a driver-neutral cost-center here, just the proportional assignment to different processes is influenced here, note the amount of costs itself.

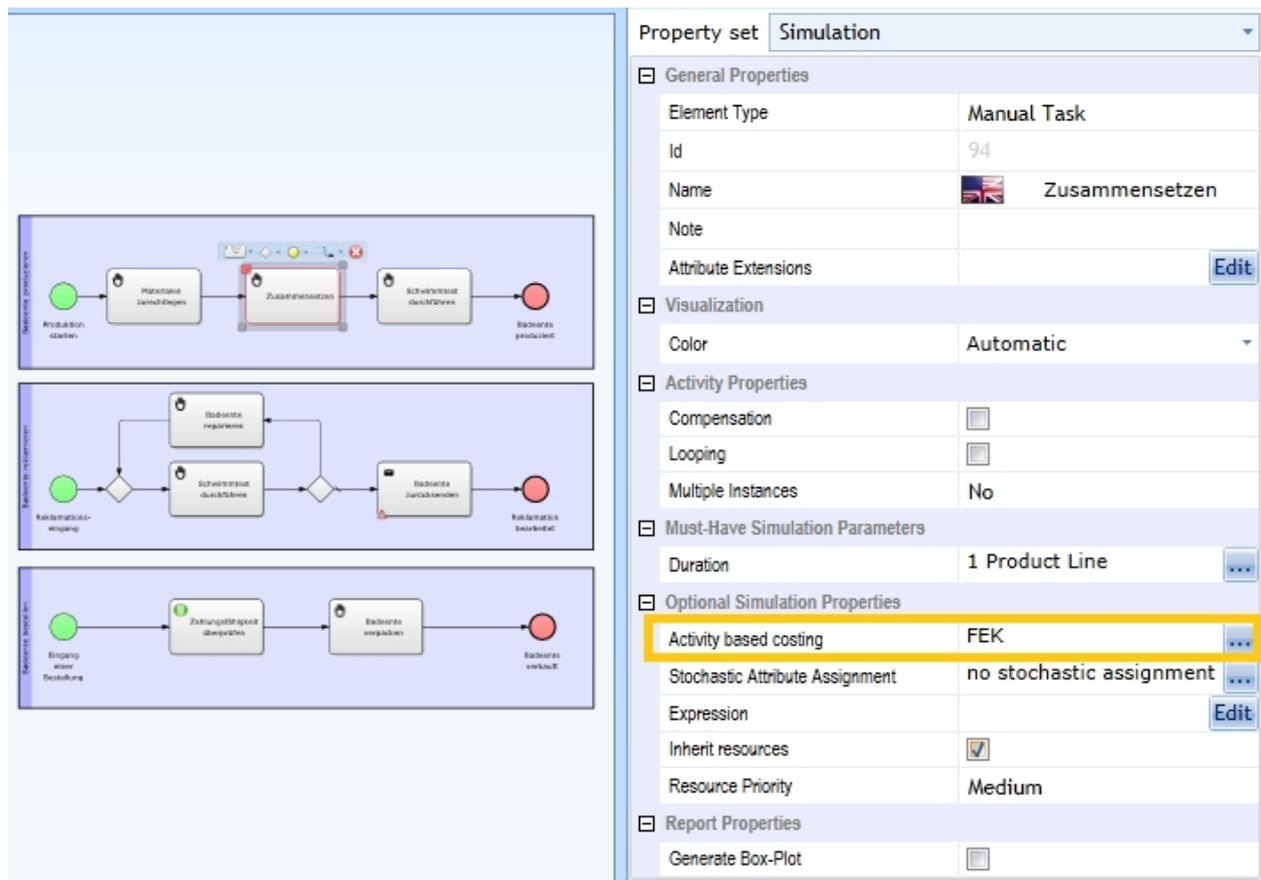


Figure 4: Open the window 'Activity based costing' for a selected activity via the property editor (be sure to be in Simulation mode)

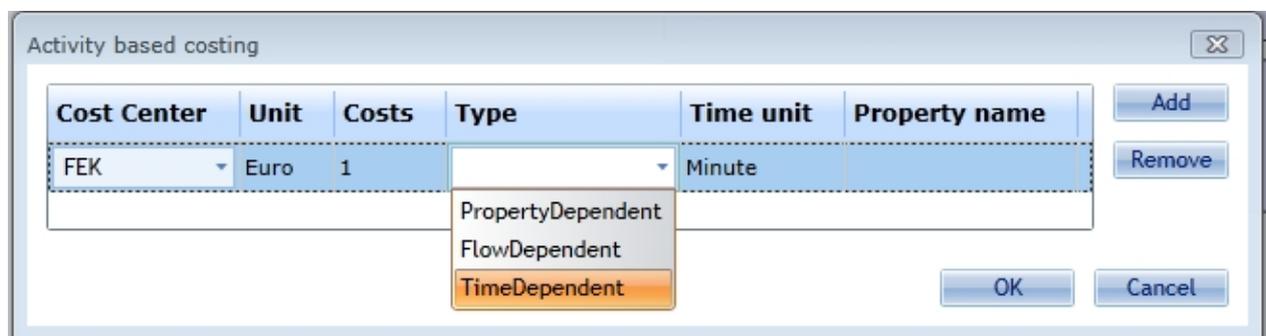


Figure 5: Configuring activity- based costs

### Resource costs

The costs generated by resources can be defined in your organisation diagrams. To work with resources, you have to [accolacte these resources to user activities](#) in your collaboration diagrams. See the corresponding [section](#) to learn how to do that. This section only deals with the cost adjustment of already allocated resources.

Open the organization diagram with your defined human resources. Be sure that you are using the simulation property set. There's a subsection called 'Costing Properties', where you have to set several parameters:

1. Costs: The amount the person earns,
2. Time Unit: The time unit in which the person earns that amount,
3. Unit: unit of measurement (usually EUROS or Dollars),
4. Payment: Indicated whether it is a fixed salary, indenpendet if person is working or not, or a variable

wage rate, where the person only gets money for time actively spend to user activities.

Property set: Simulation	
General Properties	
Element Type	Human
Id	1
Name	Anna Logica
Note	
Attribute Extensions	<a href="#">Edit</a>
Visualization	
Color	Automatic
Costing Properties	
Costs	5000
Payment	Fixed salary
Time Unit	Month
Unit	Euro
Optional Simulation Properties	
Has Basic Load	<input type="checkbox"/>
Post-processing Time	not selected <a href="#">...</a>
Setup Time	not selected <a href="#">...</a>

Figure 6: Open the window 'Activity based costing' for a selected activity via the property editor (be sure to be in Simulation mode)

## Products

Products are cost-drivers, since the number of your executed processes depends on number of products handled. A process definition can deal with a set of products, while a process instance deals with exactly one product or is completely product-neutral. Determining which process instances deal with which products allows a product-based costing allocation.

## Defining products

Create an orgigram to define your products. Change the element type of the orgigram element to 'Product' and give your product a name. Be sure that the orgigram is part of your BPI project. If you once have declared your products, you can use them inside your collaboration diagrams.

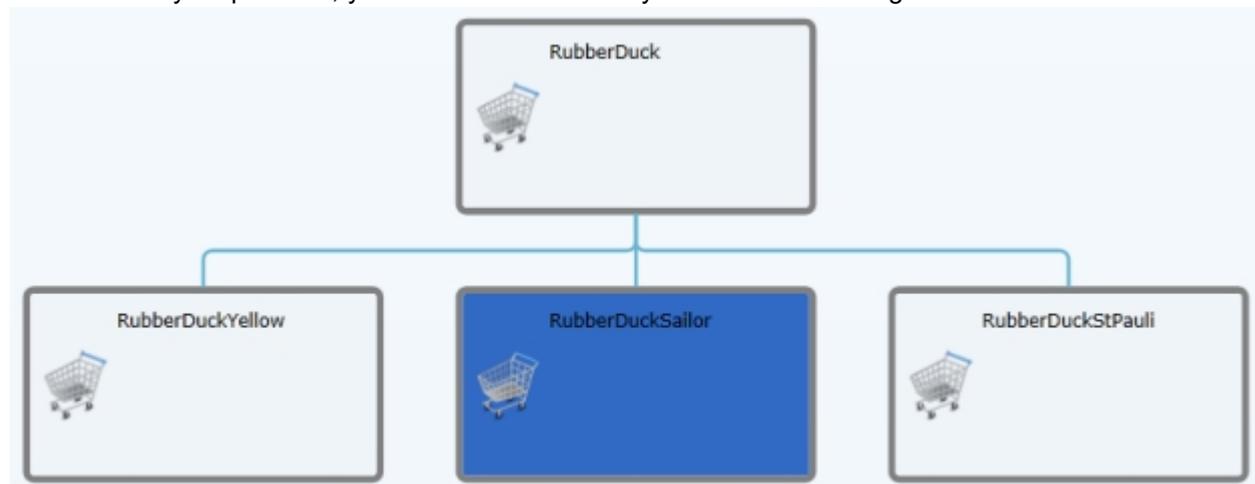


Figure 7: Orgigram with products

## Start-events

The products defined in the organigram will appear in the time editor for events and activities. In the upper right there is drop-down box, which's default is 'default'

At a start-event 'default' means, that the whole process will be product-neutral. If a product is set at a start-event, the whole corresponding process will deal with that product. Set one interarrival-time for each product which is processed in that process.

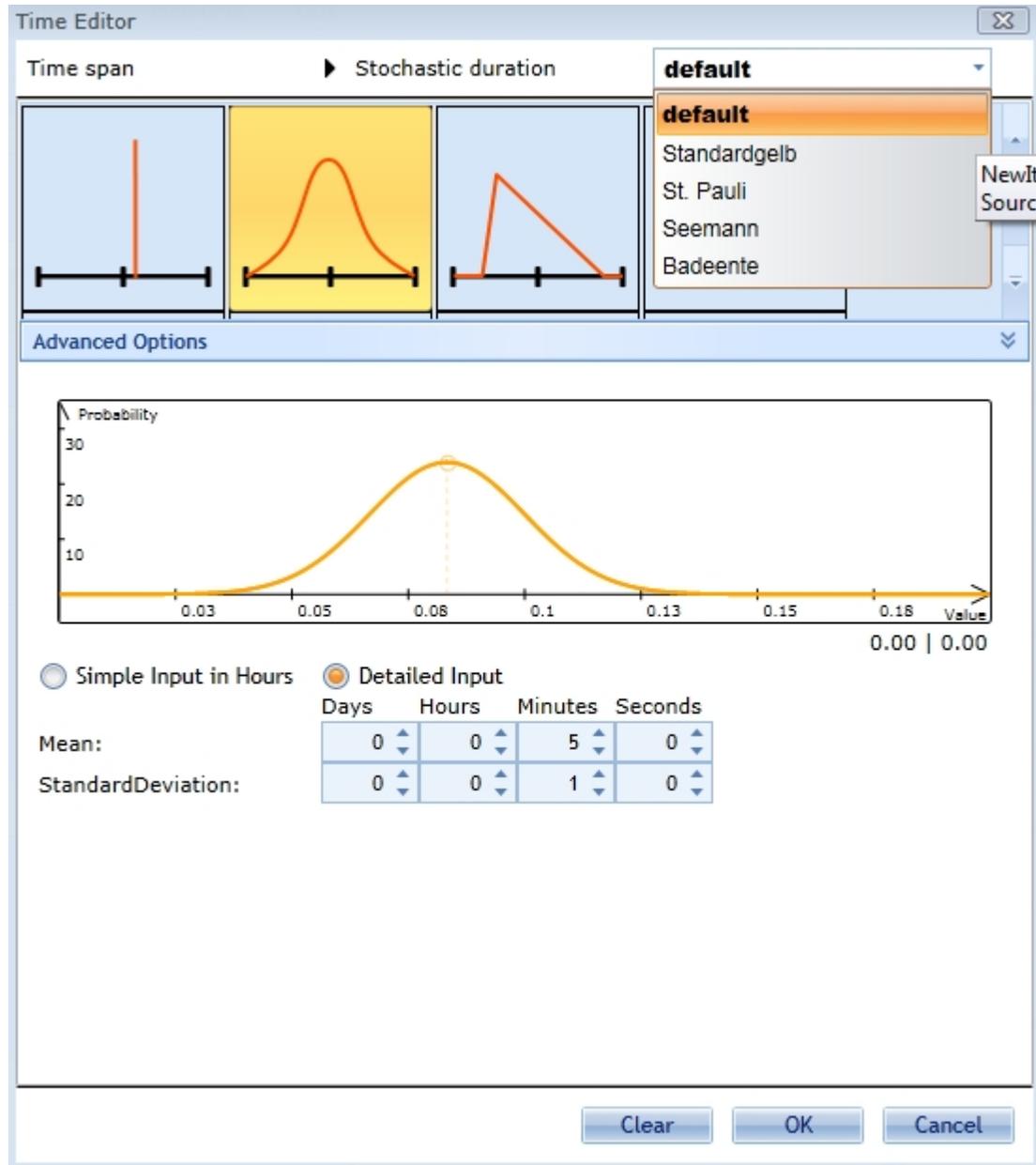


Figure 8: Time-Editor with drop-down box to set product specific parameters

## Activities

Once a process processes a specific element, the duration of activities may differ corresponding to the current product. That's why you can set a specific duration for each product used in a process. If there are no specific duration parameters for a current product, the 'default' duration parameters will be used.

## Gateways

Decision making at splitting exclusive or inclusive gateways may depend on the current product. You either can use python scripts (see next subsection) or product-dependent probabilities. Select a sequence link after an splitting gateway and click on 'Probability' in the property editor to open the window showed in figure 9 (be sure using simulation property set).

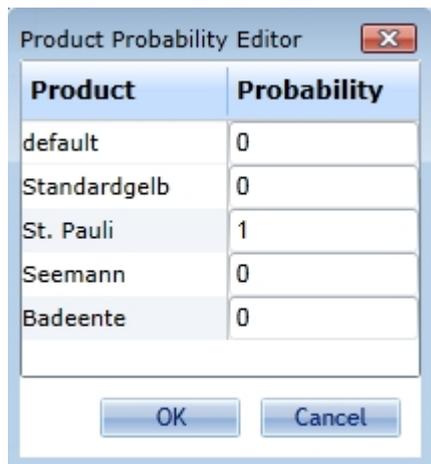


Figure 9: Setting different probabilities for each product and sequence links behind splitting exclusive or inclusive gateways

### Python Scripts

You can access the current product of a process instance within Python scripts, e.g. the following conditions, checks whether the current product is "RubbeDuckYellow":

```
SimulationContext.Product=='RubberDuckYellow'
```

The following expression shows how to set a product with help of python scripts within an already running process instance.

```
SimulationContext.Product='RubberDuckRed'
```

### Result

After running a simulation experiment, the report will be opened automatically. If you have declared cost centers, your report will contain a section called 'Costing Details'.

It will contain one process table per measurement unit. E.g. you can do a monetary costing calculation, the accounts will be held e.g. in Euro. Since you can freely set the measurement unit (e.g. to EURO, Dollars, CO2 or kWh) you also can use the calculation technique for ecological auditing like life cycle assessment.

If you have defined products, additionally, you will get one product table per measurement unit. When you click on a column header representing a product, a corresponding process table will be opened, which apportions the costs of the process instances used for that product.

- The upper half of the table (up to the row 'Total deduced costs') represents variable, driver-dependen costs.
- The following rows up to 'Total activity costs' represent fixed, driver-neutral costs, which are apportioned to the product-/process-columns by the activity-based technique.
- The last lines up to but exclusive 'Total costs' represent driver-neutral costs, which are apportioned by a fix surcharge rate. This technique is chosen, if you didn't allocate your neutral costs to activities at all.
- The last line 'Total costs' is the sum of all costs for a process, a product or the combination of a process and a product.

### Process Costs Euro

	Pool: Badeente bestellen#1	Pool: Badeente reklamieren#1	Pool: Badeente produzieren#1
Versandkosten	1182	156	0
MEK	0	228	8305
FEK	0	404.08	8304.93
Total deduced costs	1182	788.08	16609.93
Fixed salaries (activity based ratio)	0	105.68	894.32
Total neutral costs (based on utilization)	0	105.68	894.32
Total activity costs	1182	893.76	17504.25
Beckenheizung (5.1%)	60.37	45.65	893.99
Total costs	1242.37	939.41	18398.24

Figure 10: Process Costs in Euro

## Experiment Results

After every simulation run a graphical result overlay and a detailed report are generated to present experiment results.

The [Experiment Results Perspective](#) is a graphical overlay for collaboration diagrams which provides a quick overview of simulation results. It contains information about

- element iterations
- total costs
- average waiting times
- average processing times

The report grants a detailed view on simulation results. It is divided into different sections:

- Resource Details: Here you will find detailed information about the resources, like usage (idle, busy or waiting time).
- Models: Every simulated model will appear here. By clicking the + (plus) symbol the following sections will open up.
  - Process Details: Here you will find all the necessary information about the process itself, like lead times.
  - Activity Details: Here you will find detailed information about each activity.
  - Event Details: Here you will find detailed information about each event.
  - Gateway Details: Here you will find detailed information about each gateway, like the overview of the chosen paths at conditional gateways.

## Content

We will use two models to describe all the report's statistics.

1. The [first model](#) will cover the Process, Activity and Event details
2. The [second model](#) will describe the queue statistics
3. The [third model](#)'s report will introduce you to the Gateway and Resource details

## Experiment Results Perspective

Experiment results are displayed automatically after every simulation run. The graphical overlay provides an overview of simulation results regarding process costs, iterations, processing and waiting time. It consists of the following overlay components

- [Histograms](#)
- [Counters](#)
- [Sankey Style](#)
- [Overlay Options](#)

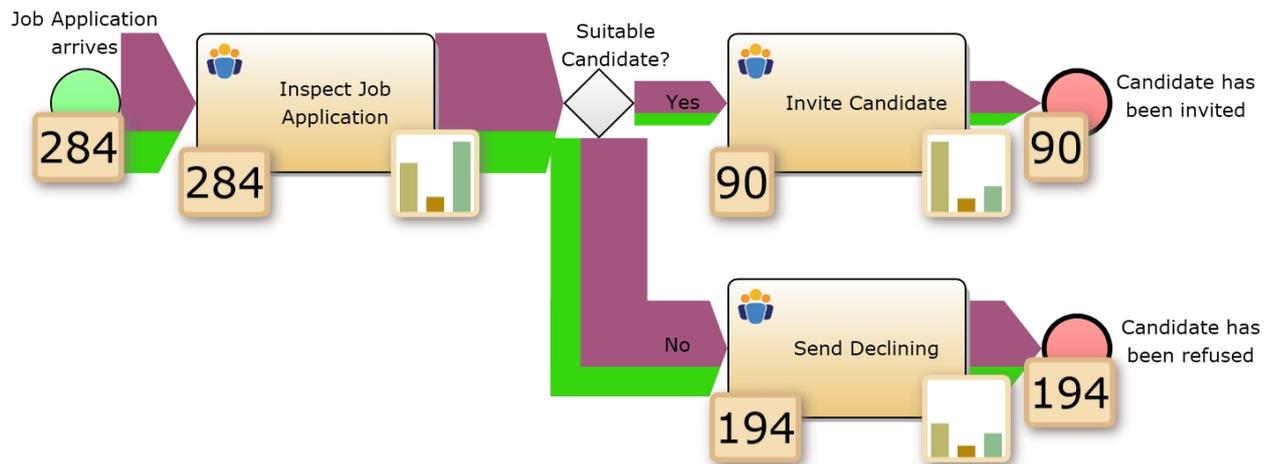


Figure 1: Example Model with activated Counters, Histograms and Sankey Style

### Histograms

A histogram is a graphical representation of simulation results which contains information of total costs, average waiting and processing time of each activities. The bar length represents the proportion of the activity results in comparison to the maximal results of the specific simulation run. Concrete numerical results for activities can be displayed by moving the mouse pointer over a histogram.

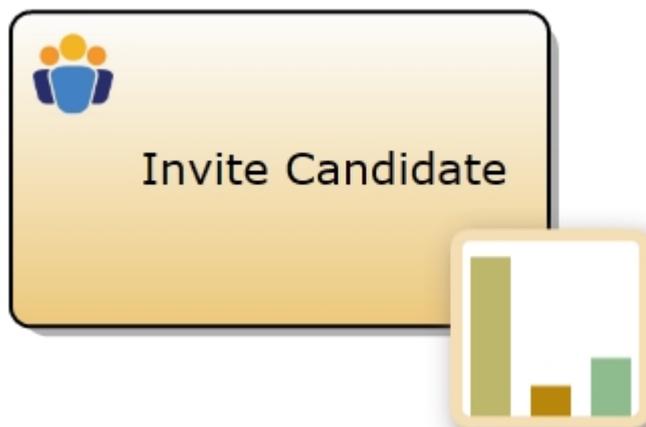


Figure 1: Histogram example (bars from left to right: average processing time, average waiting time and total costs)

### Counters

Counters represent the absolute frequency of an element for a specific simulation run. They can be displayed for events and activities.



Figure 1: Activity with counter

### Sankey Style

The Sankey Style illustrates the quantity of iterations for a single sequence flow in proportion to the total amount of process runs for a specific simulation run. Sequence flows with a higher number of iterations are displayed proportionally wider than those with a lower number of iterations.

### Sankey Style for different products

A simulation which includes more than one product will result in a Sankey Styles with different colors - one for each product. The width of each color string is influenced by the number of iterations for each product in proportion to the overall count of the sequence flow. The concrete numerical iterations for each product can be displayed by moving the mouse pointer over a Sankey Style.

### Overlay Options

Overlay Options provide an opportunity to customize the complexity of provided experiment results shown on [collaboration diagrams](#).

The following settings are available

Colorize	Activate and deactivate the colorization of activities. Activities can be colorized referring to their number of iterations, total costs, average waiting and processing time. A red color signals a high proportion and green a low proportion in comparison to the maximum result of the particular characteristic.
Color Count	Define the number of different colors for Sankey illustrations. It can be set between 1 and 16.
Event Count	Activate and deactivate the event counter.
Activity Count	Activate and deactivate the activity counter
Activity Histogram	Activate and deactivate <a href="#">histograms</a> for activities.
Sankey-Style Sequence Flow	Activate and deactivate <a href="#">Sankey Style</a> .

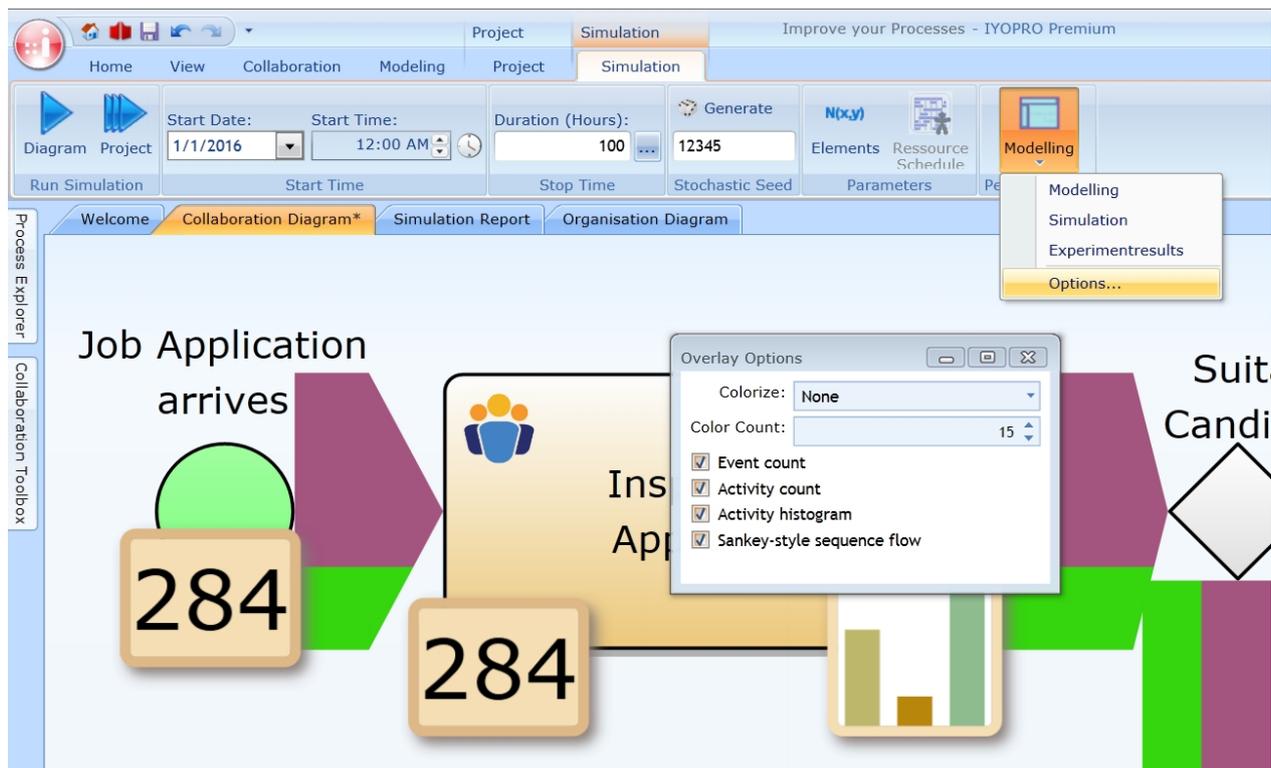


Figure 1: Opened Overlay Options window

### Model for Process, Activity and Event Details

#### The Model

We executed a simulation run on the model depicted below. It is quite simple and consists of one start event, two activities and one end event.

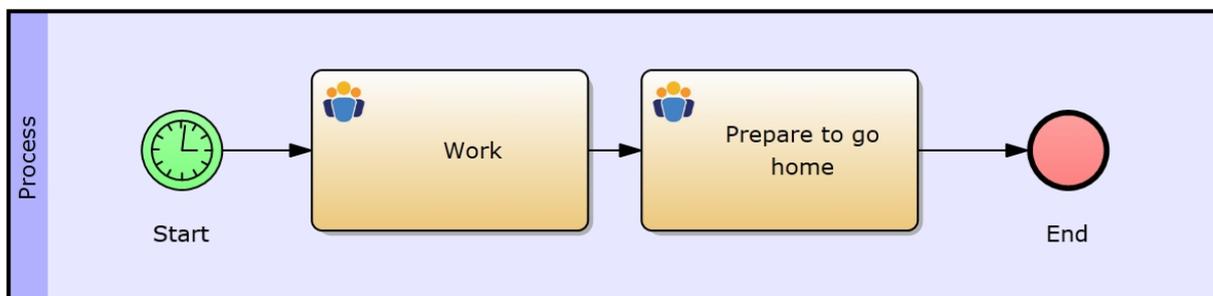


Figure 1: Example Model

We urge you to reproduce the simulation report described below to get a better understanding, if you are looking at this topic for the first time. The following table shows the parameter settings of the model. We left all the other parameters unchanged (default).

BPMN Element	Simulation Properties
Start event	Inter-arrival Time: Constant Distribution with Constant Value = 2
Activity "Work"	Duration: Constant Distribution with Constant Value = 12
Activity "Prepare to go home"	Duration: Normal Distribution with Mean = 12 and Standard Deviation = 3
General Simulation Properties	
Stochastic Seed	12345
Simulation stop time (hours)	240

## Process Details

### Process run-times

The first section of the report (after the model description) deals with the general statistics of the process (es). The information gives you an overview of the process(es)' lead time(s), which are visually enhanced by histograms or box-plots.

### Histogram

The first figure in the process details depicts a histogram for the processing time of one of the processes (in this case each pool is a process). It simply shows the amount of process instances (Y-axis), which finished in a certain interval (X-axis).

- In our example the length of the process lead time concentrates in the range 22 until 27 hours. (See the histogram in Figure 1)
- So, this first glance at the report tells us, that we need to find out, where the process instances out of range originate from.
- The process instances with a lead time less than 22 hours are not our problem, but they might be the key to shift our cluster (22-27 hours) to the left.
  - Let us just assume that the shorter the lead times of the process the better.
- The process instances on the right side of our cluster are the problem that need to be dealt with. So let's go further into it!

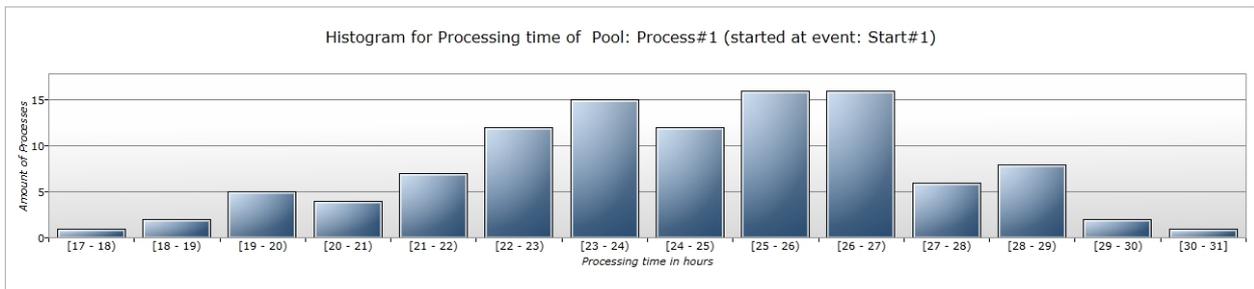


Figure 1: Report: Process details - Histogram for the processing time

Figure 2 below shows the statistics for the process(es)' lead time(s).

See the extra table for more information on the statistical elements.

Process runtimes

Title	Obs	Mean	Std.Dv	Min	Max	Lower Quartile	Median	Upper Quartile
Process	107	1d, 0h, 24m and 1s	0d, 2h, 42m and 13s	0d, 17h, 50m and 17s	1d, 6h, 53m and 26s	0d, 22h, 32m and 17s	1d, 0h, 44m and 54s	1d, 2h, 32m and 59s

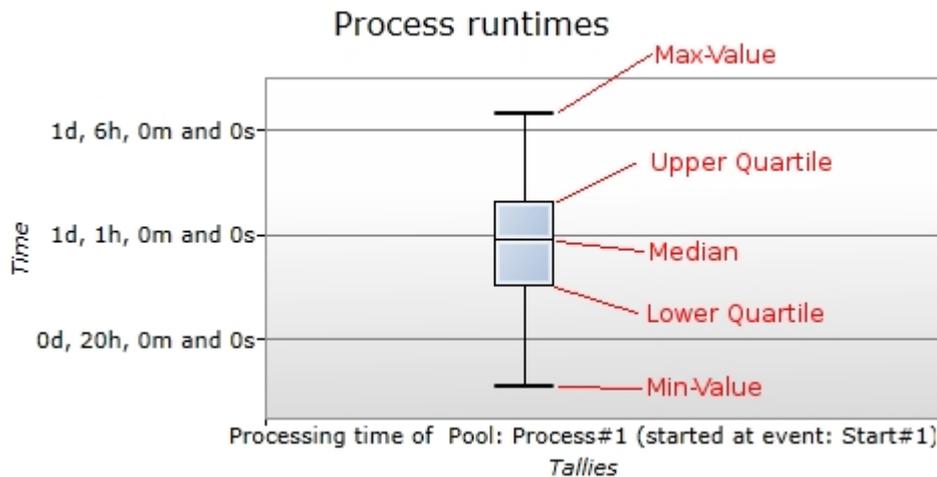
Figure 2: Report: Process details - Process run-times (in table representation)

Process runtimes (see table above and the box-plot below)	
Title	Name of the process.
Obs	The number of finished process instances.
Mean	The mean value of the process' lead time.
Std.Dv	The standard deviation of the process' lead time.
Min	The minimum value of the process' lead time.
Max	The maximum value of the process' lead time.
Lower Quartile	The lower quartile of the process' lead time.
Median	The median of the process' lead time.

Upper Quartile	The upper quartile of the process' lead time.
----------------	---

### Box-Plot for the process run-times

Additionally to the histogram we provide a box plot that represents the table above (Process runtimes). Notice that the cluster we discovered in the histogram is also illustrated within this box plot (the range from lower quartile to upper quartile).



Report: Process details - Process run-times (in Box-Plot presentation)

### Concurrently existing processes

Finally, there is a further statistic table, that reveals the concurrently existing instances of a specific process.

We learn that there approximately 12 process instances of our model have been active at the same time.

#### Concurrently existing processes

Title	Obs	Mean	Std.Dv	Min	Max	Period
Coexistent instances of processes in Pool: Process#1	226	11.6338	2.21605	1	15	9d, 22h, 0m and 0s

Report: Process details - Concurrently existing processes

Concurrently existing processes (see table above)	
Title	Name of the process
Obs	The number of started activity instances.
Mean	The mean value of the concurrently existing process instances. In this case there were 11.6338 process instances active on average.
Std.Dv	The standard deviation of the concurrently existing process instances.
Min	The minimum value of the concurrently existing process instances. In this case there was at least one process instance active throughout the simulation.
Max	The maximum value of the process' lead time. In this case there were at most 15 process instances active throughout the simulation.
Period	The period in which the process instances occurred. Note that we set the simulation stop time to 240 hours, which are ten days actually. Hint: The difference of two hours between that and the period information originates in the fact that we set the process' inter-arrival time (start event) to constantly two hours.

## Going deeper

- In a lot of cases the histogram and the box-plot can already provide a starting point for improvements.
- Nevertheless we need to go deeper into the process' details to fully understand its dynamics.
- That's why the further sections of the report deliver information on the process' elements to comprehend the process' local dynamics for the benefit of the entire process efficiency / structure, etc.
- Advance to the report's [Activity Details](#)

## Activity Details

This section of the report deals with activity details. These statistics are part of the process' local dynamics, whose understanding has a crucial impact on the comprehension and improvement of the entire process.

## Concurrently existing activities

The table from Figure 1 depicts the concurrently existing activities. We learn that there have been approximately 6 instances of each of the activities active at the same time.

Title	Obs	Mean	Std.Dv	Min	Max	Period
Coexistent instances of activity: "Work#1"	232	5.87395	0.66805	1	6	9d, 22h, 0m and 0s
Coexistent instances of activity: "Prepare to go home#1"	220	6.06569	1.14778	1	9	9d, 10h, 0m and 0s

Figure 1: Report: Activity details - Concurrently existing activities

## Run-times and waiting times

Figure 2 shows the activity run-times and the waiting times of activities.

- The runtime or lead time of an activity is the time between starting (reaching) an activity and completing (leaving) it.
- The table contains
  - the observations of finished activities,
  - its lead time's mean,
  - standard deviation,
  - minimum and maximum values
  - Just like the table for the process' lead time.

## Hint

- A waiting time only occurs when an activity has to wait for a message or a resource.
- Since we did not add any message flows or resources to the our simulation model, there are no waiting times in the table below.

## Explanation

- In our example process the activity "Work" lasts constantly twelve hours.
  - This is not a surprise, because we set a constant distribution with the value twelve hours for this activity.
- The activity's "Prepare to go home" duration has been assigned a normal distribution with a mean of twelve and a standard deviation of three hours.
  - As you see the actual mean value of the duration is 12 hours 24 minutes and 1 second.
  - This does not match twelve hours, which we set to be the mean value.
  - Still, this is right. Since we simulated only 240 hours, this value cannot be exactly twelve hours.
  - It will converge towards "our" mean value the longer we simulate.
  - Also, if we set another seed this value will be different.
  - That is the reason, why it is extremely important to simulate the model several times\* with different seeds, before making any decisions.

Activity runtimes and waiting times						
Title	Obs	Mean	Std.Dv	Min	Max	
Processing time of activity: "Work#1"	113	0d, 12h, 0m and 0s	0d, 0h, 0m and 0s	0d, 12h, 0m and 0s	0d, 12h, 0m and 0s	
Processing time of activity: "Prepare to go home"	107	0d, 12h, 24m and 1s	0d, 2h, 42m and 13s	0d, 5h, 50m and 17s	0d, 18h, 53m and 26s	

Figure 2: Report: Activity details - Activity run-times and waiting times

## Distributions

Finally the table from Figure 3 shows the activities' distribution statistics.

In this case we get statistics for the two distributions we have specified in the model (constant distribution and Normal distribution).

## Note

- The distributions' seeds are not equal to the general seed (12345).
- These local seeds are generated from the general seed
  - which means that they will always be the same as long as you always use the same general seed.

Distributions						
Title	Obs	Type	Parameter 1	Parameter 2	Parameter 3	Seed
Duration for element: Work#1	119	Cont Constant	12			7015950
Duration for element: Prepare to go home	113	Cont Normal	12	3		51113926

Figure 3: Report: Activity details - Distribution statistics

Distributions (see table above)	
Title	Name of the process
Obs	The number of started process instances.
Type	An abbreviated denomination of the distribution's type. E. g. Cont Normal for the Normal distribution
Parameter 1*	The value for the distribution's first parameter.
Parameter 2*	The value for the distribution's second parameter.
Parameter 3*	The value for the distribution's third parameter.
Seed	The distribution's seed value.

## Also note

The amount of parameters changes depending on the distribution type. E. g. the normal distribution needs two parameters, the Constant distribution though needs only one.

## Event Details

This report topic deals with the event details.

## Inter-arrival times

The table in Figure 1 shows the events' inter-arrival times. In this case we have got the inter-arrival times of the start and the end event from our simulation model.

- The inter-arrival time between the start events was calculated 118 times, and it is constantly two hours.
  - Since we assigned a constant distribution with a value of two hours, this is not really a surprise.
- The inter-arrival time between the end events has been calculated 106\* times.
  - As you can see, this statistic is dependent on the activity "Prepare to go home", because it is

- the only stochastic influence in this model
- \* We have got 107 observations of the process lead times.
  - To calculate an inter-arrival time we need two finished end events, that is why we have got one instance less in the inter-arrival time statistic of the end event than in the process lead times.

Interarrival Times						
	Title	Obs	Mean	Std.Dv	Min	Max
	Interarrivaltime for start event: "Start"	118	0d, 2h, 0m and 0s	0d, 0h, 0m and 0s	0d, 2h, 0m and 0s	0d, 2h, 0m and 0s
	Triggered event "End#1" Interarrivaltime	106	0d, 2h, 2m and 16s	0d, 1h, 33m and 36s	0d, 0h, 0m and 12s	0d, 6h, 20m and 3s

Figure 1: Report: Event Details - Inter-arrival time statistics

### Event counter

The table in Figure 2 depicts the statistics for the events' counters. A counter is a simple statistic data type which represents the number of executed events.

In this case the end event has been executed 107 times, which is exactly the number of finished process instances from the process details view.

Event counter						
	Title	Type	Obs	Current Value	Min	Max
	Triggered event "End#1"	Count	107	107	0	107

Figure 2: Report: Event Details - Counter statistics

### Distributions

Finally, Figure 3 shows the statistics for the events' distributions.

In our simulation model we have defined only one distribution at the start event, which has been a constant distribution (value = 2).

Distributions							
	Title	Obs	Type	Parameter 1	Parameter 2	Parameter 3	Seed
	Interarrival time at Element: Start#1	120	Cont Constant	2			82730829

Figure 3: Report: Event Details - Distribution statistics

### Model for Queue statistics

This section covers the queues' statistics.

Queues are used to represent specific model dynamics, in which an entity has to wait for another entity.

A pictorial example would be:

- Imagine the daily operation of a barber shop.
- There are two barbers that are being entered into the waiting Barber-Queue at the beginning.
- In the morning there are no customers, the barbers are just sitting around and waiting (in the queue).
- Suddenly at 10 O'clock the first customer arrives and inserts himself into the customer-queue.
- Since there are no other customers, this customer gets removed from the queue instantly along with a barber from the barber-queue.
- Ten minutes later two additional customers enter the barber shop and the customer-queue.

- There is only one barber available, so the first customer from the customer-queue gets served.
- The other one must remain in the queue until one of the barbers is available again.
- This goes on until the shop closes.

Our queue statistics get updated every time an entity enters or leaves a queue. See the following model and its report for a detailed example.

## The Model

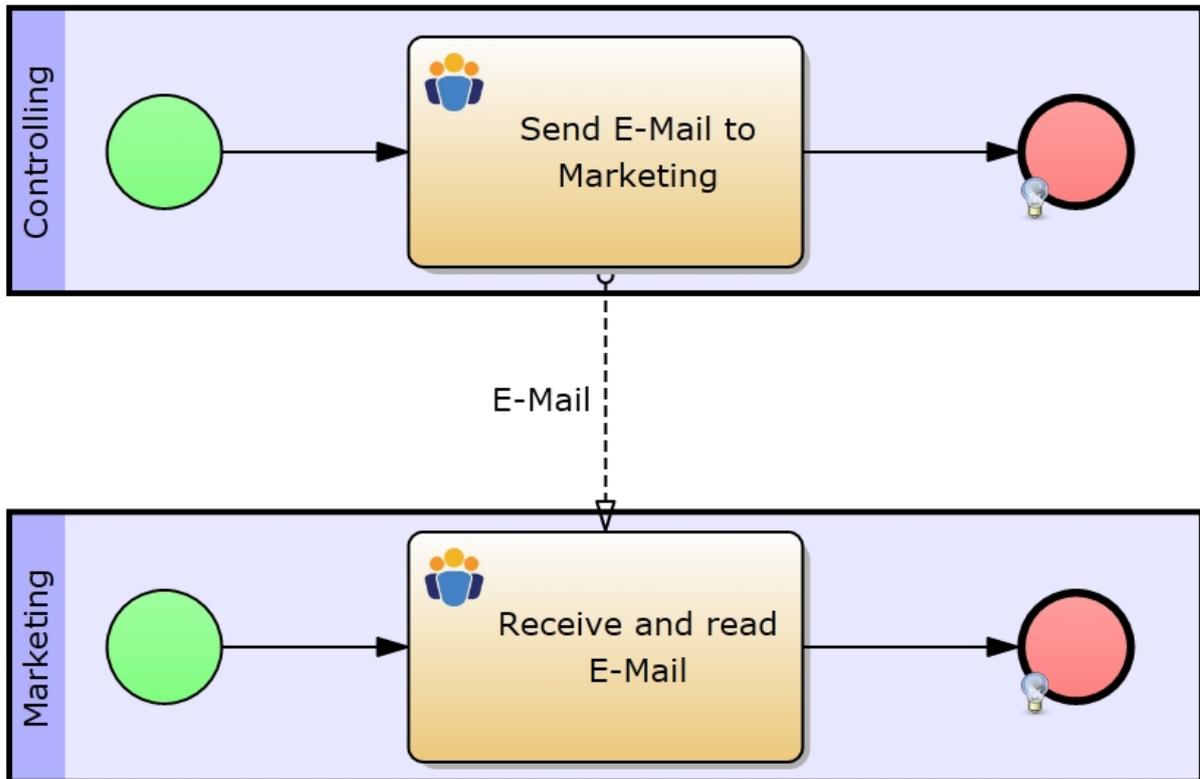


Figure 1: Example Model

## Process description

- The "Controlling" process starts constantly each hour.
  - It contains one activity that lasts between one and two hours.
  - Additionally the activity sends a message to the "Marketing" process and terminates after that.
  - Note that messages are sent once the activity has been reached.
- The Marketing process is as trivial as the one mentioned before.
  - However, we constructed its simulation parameters in a way, so that we get significant queue-statistics in any case.
  - To achieve this, we made the "Marketing" process start every two hours (Constant Distribution).
  - Thus, the "Controlling" process always sends a message to the bottom pool's activity, before any process instance can reach it.
  - This means the message is being queued to be handled by a to-be process instance, that starts in the future (simulation time).

See the [report](#) for the description of the resulting queue statistics.

BPMN Element	Simulation Properties
Start event	Inter-arrival Time: Constant Distribution with value = 1

"Controlling"	
Start event "Marketing"	Inter-arrival Time: Constant Distribution with value = 2
Activity "Send E-Mail to Marketing"	Duration: Discrete Uniform Distribution with minValue = 1 and maxVale = 2
Activity "Receive and read E-Mail"	Duration: Poisson Distribution with a mean value of 2
Messageflow "E-Mail"	This element does not contain any simulation information in this model.
General Simulation Properties	
Stochastic Seed	12345
Simulation stop time (hours)	240

## Queue statistics

Figure 1 depicts the statistics table on the queues:

- Note that only the message receiving activity possesses queues.
- There are two queues.
- This is based on the following fact:
  - Whenever a message or a process instance reaches an activity (that has an incoming message flow) there are two possibilities:
    - Either the message is the first entity to arrive or the process instance.
      - If the process instance arrives first, it will have to wait for a message to work off the activity.
      - Otherwise if the message arrives first, it will have to wait for a process instance to pick it up.
- The latter construct is depicted in the table above.
- In our example model, the messages always arrive first.
- Therefore the message queue is the only one with significant statistical information for this model.
- As you can see, most of the process queue's values are zero, though we have got eleven Observations (of removing a process instance from the queue).
- But, let us check out the Zero column.
  - Its value is also eleven.
  - This means each of the eleven process instances (that have been put into the queue) have been removed instantly, because they did not have to wait for a message.

Queues											
Title	Qorder	Obs	QLimit	Qmax	Qnow	Qavg.	Zeros	max.Wait	avg.Wait	refus.	
Message-Queue at receive-event: "Receive and read E-Mail#1"	FIFO	11	unlimit.	12	12	6.26087	0	0d, 11h, 0m and 0s	0d, 6h, 0m and 0s	0	
Process-Queue at receive-event: "Receive and read E-Mail#1"	FIFO	11	unlimit.	1	0	0	11	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0	

Figure 1: Queue statistics

## But...

- On the other hand the message queue shows us the bottle-neck of this model.
- Here we have also got eleven observations, which is correct, because every time a process instance is being reactivated, it consumes one message (which will be removed from the message queue).
- Check out the other values:
  - Currently there are twelve messages (Qnow) waiting to get picked up.
  - This value matches the maximum number of elements (observed throughout the simulation).
    - This indicates, that if we increase the simulation runtime, the maximum waiting time and with that the average waiting time will also increase.
    - We can explain this correlation by having a look at the general simulation properties.

- The "Controlling"-process starts every time unit (Inter-arrival time = constantly 1),
- and the "Marketing"-process starts every second time unit (Inter-arrival time = constantly 2).
- So, we are sending twice as many messages as there are "Controlling"-process instances, which can receive the messages.

## Note

- Of course there are models, where this principle of "always arriving first" is not valid and "gets mixed up" due to the stochastic influences on the model.
- E. g. change the inter-arrival time of our example model's both start events to
  - a Normal Distribution with a mean value of 2 and a standard deviation value of 0.5.
- Check the report and you will notice, that both queues possess significant waiting times now.

See the table below for a description of the statistical elements of queues.

Queues	
Title	Name of the queue.
Qorder	The Queuing order: FIFO (First in first out) is the default value and cannot be changed.
Obs	Observed removals of entities.
QLimit	The queue's limit (maximum number of entities this queue can contain). The default value is unlimited and cannot be changed.
Qmax	The maximum number of entities, this queue contained throughout the simulation.
Qnow	The current number of elements contained in this queue at the simulation stop.
Qavg	The average number of elements contained in this queue throughout the simulation.
Zeros	An element that is instantly removed from the queue without any time consumption is called a zero. (Zero for: spent 0 time units in the queue.)
max. Wait	This value represents the maximum waiting time of an element before getting removed from the queue.
avg. Wait	This value represents the average waiting time of an element before getting removed from the queue.
refus.	The number of elements that have been refused to be inserted into the queue.

## Activity run-times and waiting times of this model

The picture from below shows the model's activity run-times and waiting times.

Activity runtimes and waiting times

Title	Obs	Mean	Std.Dv	Min	Max
Processing time of activity: "Send E-Mail to Marketing#1"	21	0d, 1h, 34m and 17s	0d, 0h, 30m and 25s	0d, 1h, 0m and 0s	0d, 2h, 0m and 0s
Processing time of activity: "Receive and read E-Mail#1"	10	0d, 1h, 42m and 0s	0d, 1h, 3m and 33s	0d, 0h, 0m and 0s	0d, 4h, 0m and 0s
Waiting time at receive-event: "Receive and read E-Mail#1"	11	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s

Figure 2: Activity run-times and waiting times

- Contrary to our previous example models, we now have got a waiting time in the bottom row of the table.
- Its values (except the Obs) are 0.
- There is no time consumption, because our processes do not wait at this explicit activity.
- There is always a message available whenever a process instance arrives at the message receiving activity (see Zero column in the Queue statistics table).
- If we switch the inter-arrival time values of the two start events (like described above), there will be a non-zero waiting time.
- Since this model does not possess any further model dynamics, the activity's waiting time value should

be equivalent to the process queue's waiting time values. Just try it out.

### Model for Gateway and Resource details

- We did not cover gateway and resource details of the report, because our example model did not use any gateways or resources.
- That is why we extend our simulation model by gateways and resource demand now, see the figure below.
- The organization diagram (resources) is shown next to the model.
  - Dave is the boss, Eugene is the project manager, and Jim and John are the programmers.

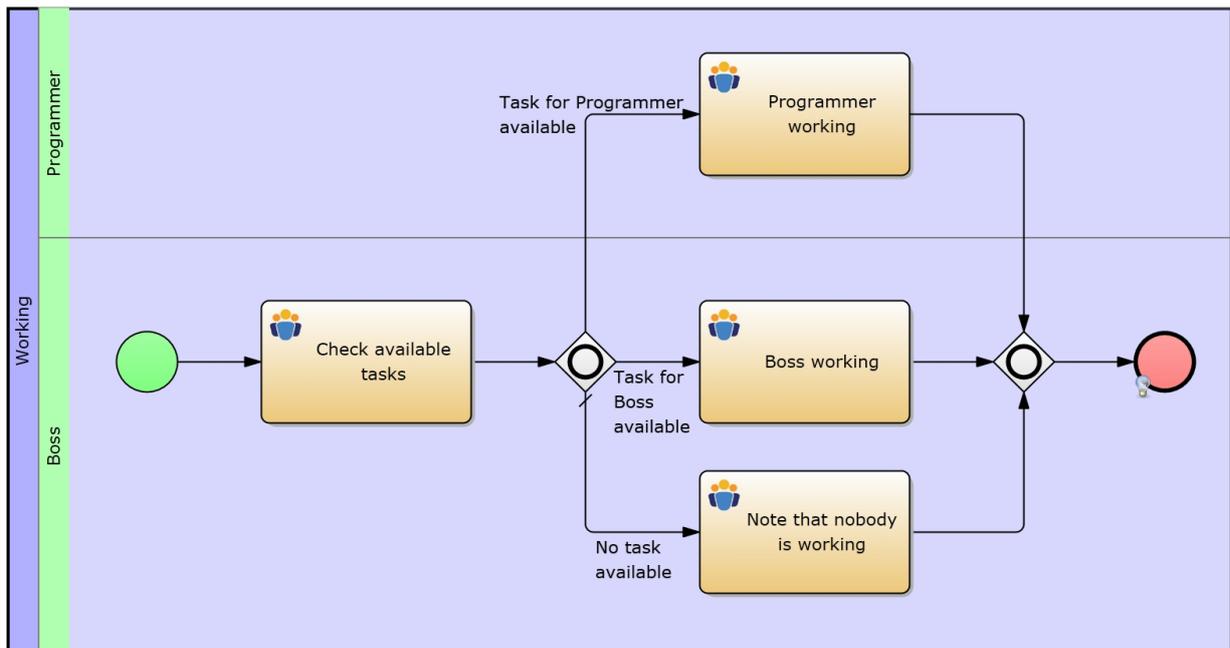


Figure 1: The example model



Figure 2: The

**Process description:**

- Our process starts at 8 o'clock every day.
- The first task is to check whether a task for a programmer and a boss is available.

- We use [stochastic variables](#) to accomplish this.
- We assume: there is a chance of 50% that a task for each of the resources is available.
- So all we need is a stochastic variable using a discrete uniform distribution with
  - the minValue = 0 and the maxValue = 1.
  - "0" stands for "not available"; "1" stands for "available".
- At the inclusive gateway we want work to be done depending on the task availability for both human resources.
  - Therefore we add an expression ([condition](#)) to the non-default sequence flows behind the gateway.
  - At the top sequence flow we check whether our "taskProgr"-variable equals "1".
    - If it does, there is a task for a programmer available
    - otherwise there is no task for a programmer.
  - We do the same for the boss' task.
  - If both expressions do not evaluate to true, the default sequence flow will be executed.
- Depending on the evaluation of the expressions at least one of the activities after the gateway will be carried out.
- The activities "Programmer working" and "Boss working" have also been assigned one participant each: Programmer and Boss.
- In the end the second inclusive gateway waits for all "live" tokens to reach it.
- The process finishes after this.

Check the Model parametrization from below and then advance to the [Gateway](#) and [Resource](#) details. Since we have already covered the first three sections of the report in the topics before, we will neglect those now. Instead we only focus on the new sections.

BPMN Element	Simulation Properties
Start event	Scripted Delay: "Result=time.GotoHour(8);" (enter this Python-script without the "") This scripted delay starts the process every day at 8 o'clock.
Activity "Check available workers"	Duration: Constant Distribution with value = 0.25 Stochastic Attribute: <ul style="list-style-type: none"> <li>● progr = Discrete Uniform Distribution with the values for min = 0 and max = 1</li> <li>● boss = Discrete Uniform Distribution with the values for min = 0 and max = 1</li> </ul>
Activity "Programmer working"	Duration: Poisson Distribution with a mean value of 8 Participant: Programmer
Activity "Boss working"	Duration: Continuous Uniform Distribution with LowerBorder = 6 and UpperBorder = 8 Participant: Boss
Activity "Note that nobody is working"	Duration: Constant Distribution with value = 0.25
Edge "Task for Programmer available"	Expression: f(x)= taskProgr==1;
Edge "Task for Boss available"	Expression: f(x)= taskBoss==1;
Edge "No task available"	Just a default sequence flow.
General Simulation Properties	
Stochastic Seed	12345
Simulation stop time (hours)	240

## Gateway Details

The figure below depicts the statistics on the chosen paths at gateways.

In this case we have got only one gateway: our (splitting) inclusive gateway. In the first table of Figure 1 we discover that:

- the gateway has fired 13 times
- the edge "No task available" has been executed least
- the edge "Task for Programmer available" has been executed most

The second table shows a detailed breakdown of the chosen paths. We note that:

- "Task for Programmer available" has been chosen six times which results to a 46% ratio of all choices
- "No task available" has been chosen two times which results to a 15% ratio of all choices
- "Task for Boss available" has been chosen five times which results to a 38% ratio of all choices

The fifth column represents the cumulated ratio.

Furthermore we added a pie chart to represent the values from the table graphically.

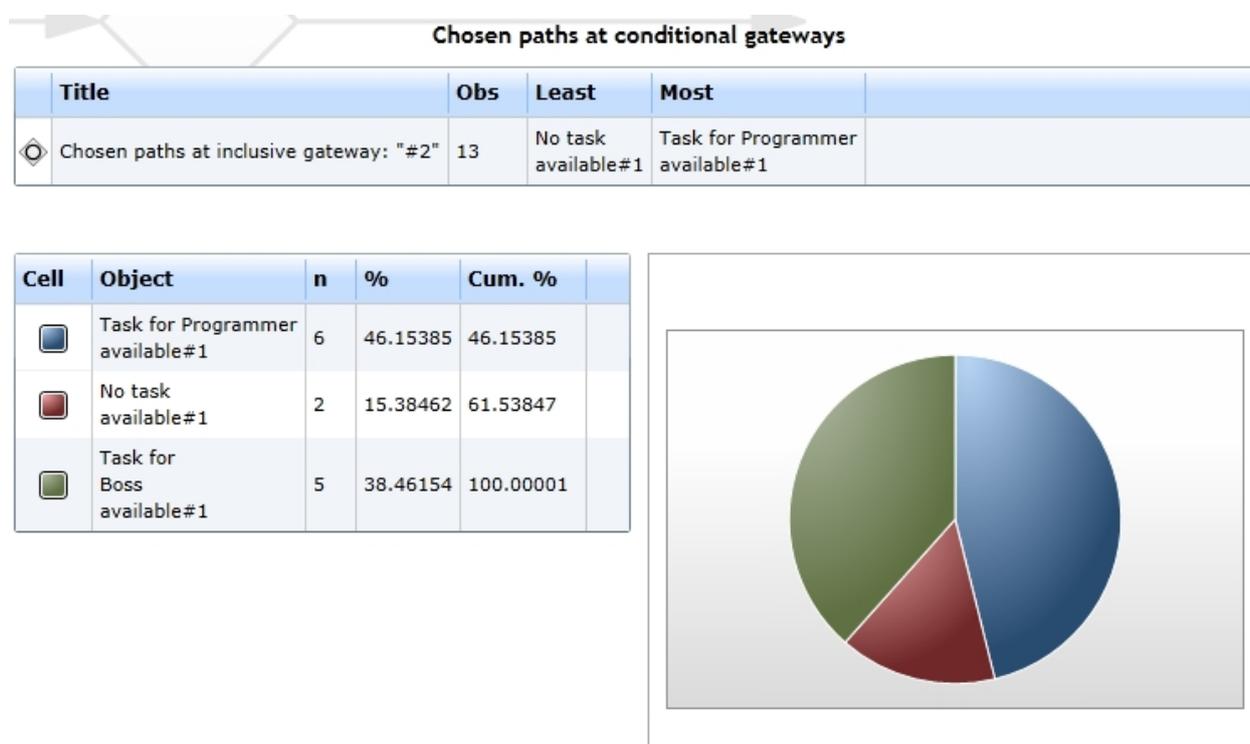


Figure 1: The Report's Gateway statistics

## Resource Details

This section covers the resource details. (Remember that we assigned one participant to the "working" activities.)

## Resource Usage

The first table in this section illustrates the resources' utilization. A resource can be set (by the simulation) to four different states (See chapter [Resources](#) from the Simulation Tutorial for more information):

- Idle: The resource is idle.
- Waiting: The resource is waiting. E. g. if an activity requires one Programmer and one Boss resource, it might get the Boss, but will possibly have to wait for a programmer. This "waiting" time will be added to the Boss' "Waiting time" after the programmer is finally available.
- In use: The resource is being used by any activity.

- Other Role: The time a resource spent in another role. E. g. when resources are assigned more than one role.
- Setup time: The time a resource spent in the setup status.
- Post-processing time: The time a resource spent in the post-processing time.
- Basic Load: The time a resource spent in the basic load status.

The values in the table are the cumulated status-time values of each resource. You might wonder why the cumulated time (waiting + in use) is 240 for the boss and 480 for the programmer. This is due to the fact that we defined two programmers in our organization diagram (see [here](#)).

- As you can see the boss is idle for 205 hours and in use for 35 hours.
- The programmer is idle for 441 hours and in use for 39 hours.
- Due to the model parameters the waiting time is always 0.
- Our resources do not have to wait for other resources, because we did not specify combined resource usage at any activity.
- The Figures 2 and 3 (bar chart and pie chart) map these values graphically.

Resource Pools			
Title	Idle	Waiting	In use
			
Resource Pool for Boss	204.550858393312	0	35.449141606685
Resource Pool for Programmer	441	0	39

Figure 1: Resource Pools statistic

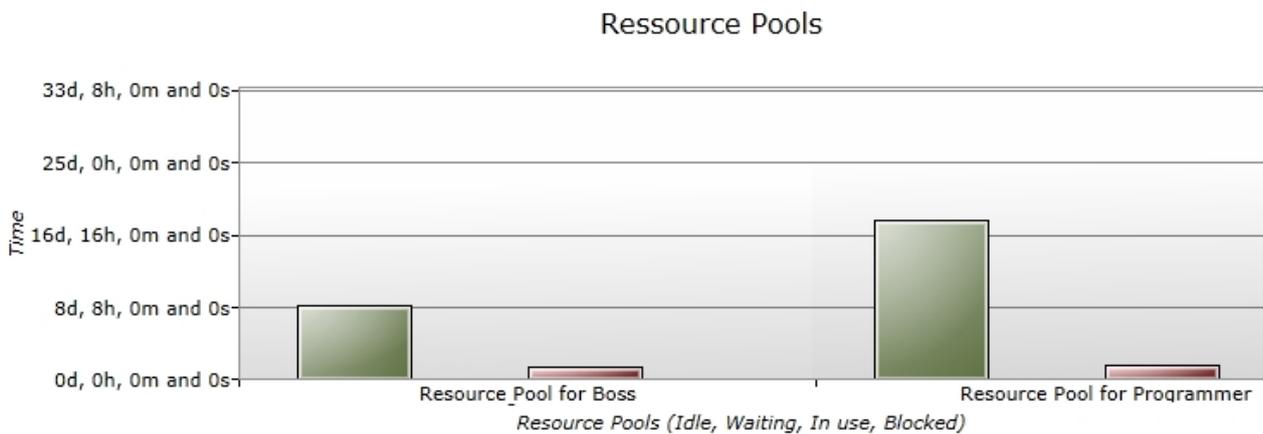


Figure 2: Histogram for the resource usage

## Resource Pool for Programmer Resource Pool for Boss

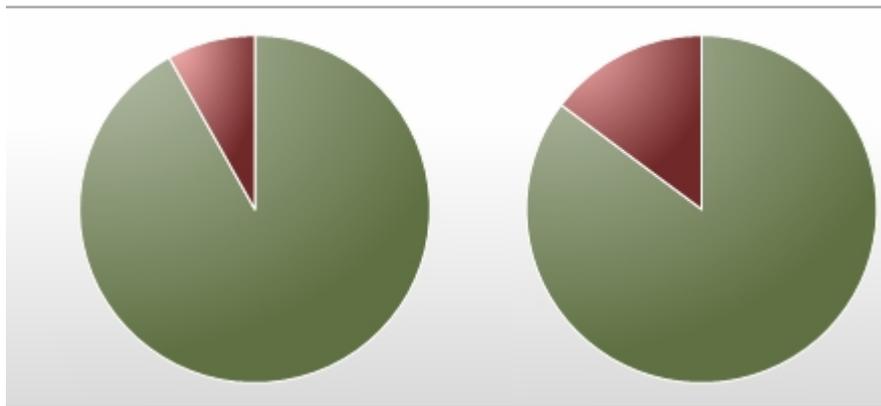


Figure 3: Pie charts for the resource usage

### Queue statistics

- Finally you can find a table in Figure 4 with the statistics for the queues,
  - which are used at each resource pool.
  - Each token that arrives at an activity (which requires a resource) inserts itself into the queue at the resource pool of which it needs one or more resource(s).
  - The queues are always set to First In First Out (FIFO) order.
- In this simple example we do not have any waiting time for a resource.
- But if you change the inter-arrival time of our model's start event to constantly one hour, you will get some significant changes in this table's values. Just try it out.

Queues										
Title	Qorder	Obs	QLimit	Qmax	Qnow	Qavg.	Zeros	max.Wait	avg.Wait	refus.
Waiting processes at resource pool: Resource Pool for Boss	FIFO	5	unlimit.	1	0	0	5	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0
Waiting processes at resource pool: Resource Pool for Programmer	FIFO	6	unlimit.	1	0	0	6	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0

Figure 4: Queue statistics of the report's Resource Details

## Simulation Tutorial

### Instructions for Simulation of BPMN 2.0 models with IYOPRO

This instruction shows how to accomplish probability depending simulation experiments with the help of IYOPRO BPMN 2.0 models. There are models for 15 consecutive lessons in the directory „Simulation Lessons“ of the IYOPRO on-line repository, that are specified in this documentation. Even though this documentation contains illustrations of the models, it is recommended to open the models with IYOPRO (use the given links to the models) and execute the specific steps for an efficient familiarization. Please note that this instruction requires good BPMN 2.0 knowledge.

The BPMN models in the directory „Simulation Lessons“ are in a consecutive order. They begin with the basic principles. Later they go on to configure more complex models that deliver meaningful data. The models and their configuration is explained below.

### Approach of this Tutorial

- We will only explain the new features, which are part of the respective lesson
  - Already explained or introduced features will be taken as given and will not be covered again
- Each lesson contains a model and a description of the purposes, which we want to achieve.
  - After explaining the lesson's new features, we will discuss the report that is produced by this model and a certain simulation configuration.

## 1. Processes and Sequence Flow

In this section we will configure simple models to give you an introduction to the simulation feature of YOPRO.

### Content

- [Model 1a - Processes & Activities](#): Get to know the basics of YOPRO's simulation feature
- [Model 1b - Time consuming sequence links](#): Learn how to use time consuming sequence links
- [Model 1c - Time Instants for start events](#): Check out another approach to schedule start events
- [Model 1d - Scripted Time Instant for Start Events](#): Learn how to set up a scripted inter-arrival time

### Model 1a - Processes & Activities

#### Goals of this lesson

- Getting to know YOPRO's simulation feature
- Setting up a simple simulation configuration
- Running a simulation

#### How-To

Have a look at our first to-be-simulated model. It consists of a Timer-Start-Event, two successive activities and an End-Event.

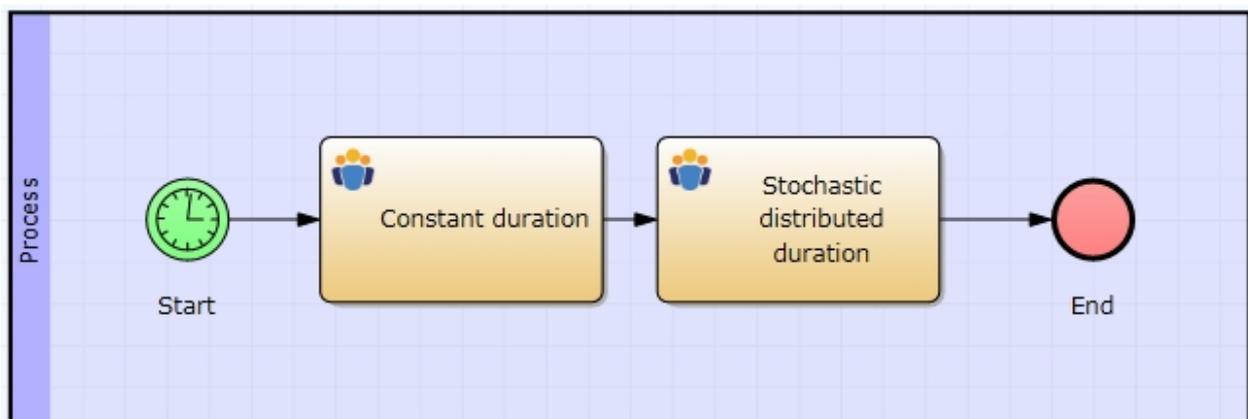


Figure 1: Model 1a - Processes & Activities / [YOPRO-Link](#)

#### First Step: The Inter-arrival Time of the start event

The simulation needs an inter-arrival time, which specifies the starting intervals between new process instances:

1. Make sure, you selected the Simulation option from the Property Set drop-down-box (See highlighted area in Figure 5 below)
2. Click the start event
3. Now click the button right of the empty field of the Inter-arrival Time option under the Must-Have-Properties (See Figure 3 below)
4. The Time Editor will show up (See the Figure 2 below and see [Assigning Inter-arrival Times](#) for detailed information)
  1. Select the Constant Distribution
  2. Set a constant value of 1
  3. Close the Time Editor by clicking OK

The Inter-arrival Time of this specific process is now set. It will always start by an interval of 1 time unit.

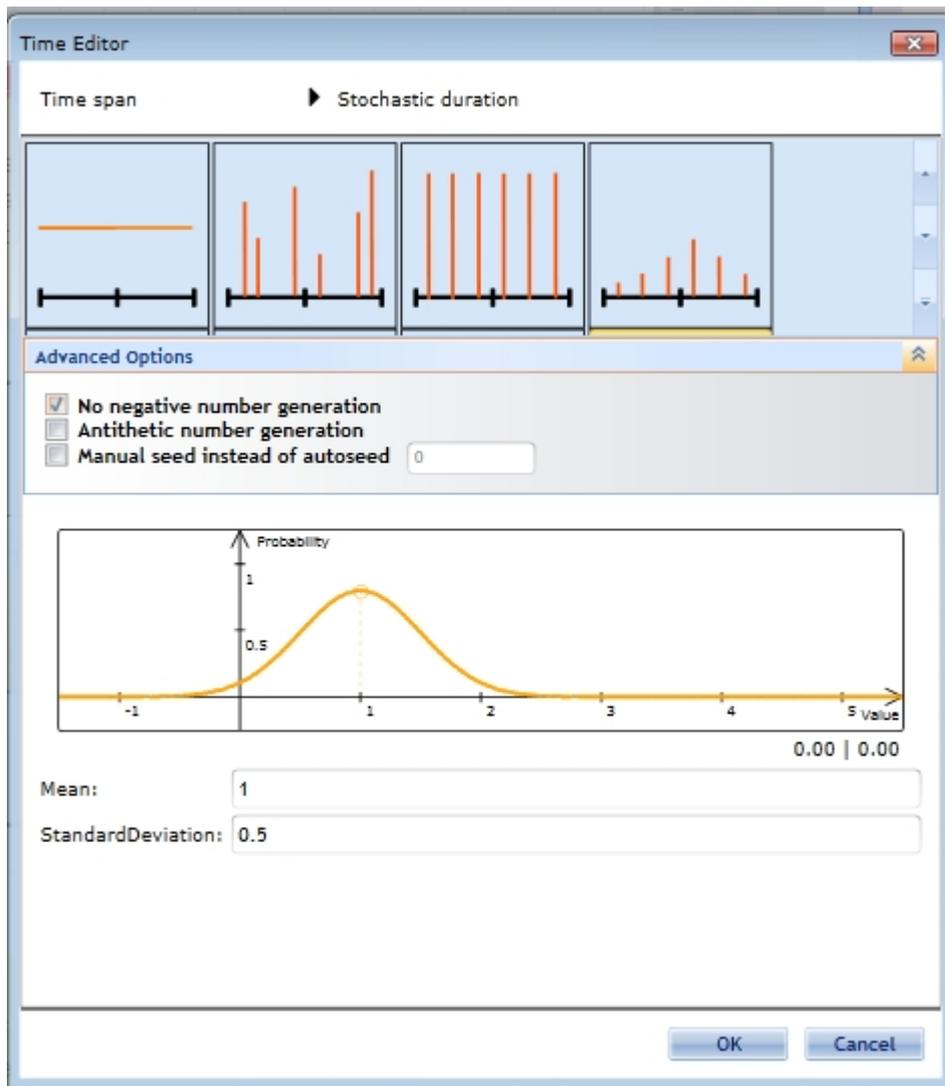


Figure 2: IYOPRO's Simulation Time Editor (in Stochastic Distribution view)

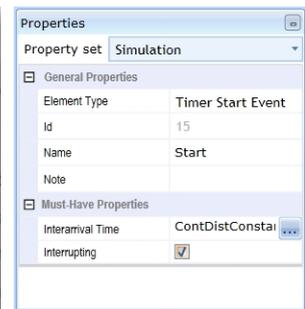


Figure 3: Simulation Properties of the Timer Start Event

## Second Step: The activities' durations

Since activities are time consumptive, they need a stochastic distribution that represents this time consumption in the simulation.

1. Make sure, you selected the Simulation option from the Property Set drop-down-box
2. Click the first activity in the sequence flow ("Constant duration")
3. Now click the button right of the empty field of the Duration option under the Must-Have-Properties (See Figure 4 below)
4. The Time Editor shows up
  1. Select the Constant Distribution
  2. Set a constant value of 1
  3. Close the Time Editor by clicking OK

As you might notice, the functionality of setting up an Inter-arrival Time and a Duration works alike.

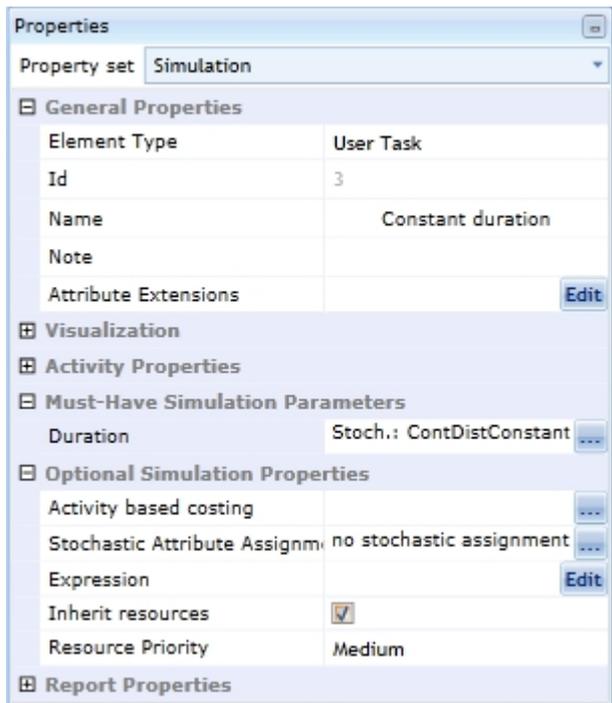


Figure 4: Simulation Properties of an activity

So far we parametrized the simulation with constant values, which mostly do not resemble the circumstances in reality. Therefore, we will bring in some dynamics in our model by setting up a real stochastic distribution for the second activity's duration.

1. Repeat the steps 1 to 3 from above for the second activity
2. The Time Editor shows up
  1. Select the Normal Distribution
  2. Set a Mean value of 1
  3. Set a Standard Deviation value of 0.5
  4. Close the Time Editor by clicking OK

### Third Step: Starting the simulation

Since the end event does not contain any simulation related options (see Figure 6 below), we now have set up all necessary properties to start the simulation. Now just click the green Analyze button (in the middle of IYOPRO's top menu bar) and wait for the Progress Bar to finish.

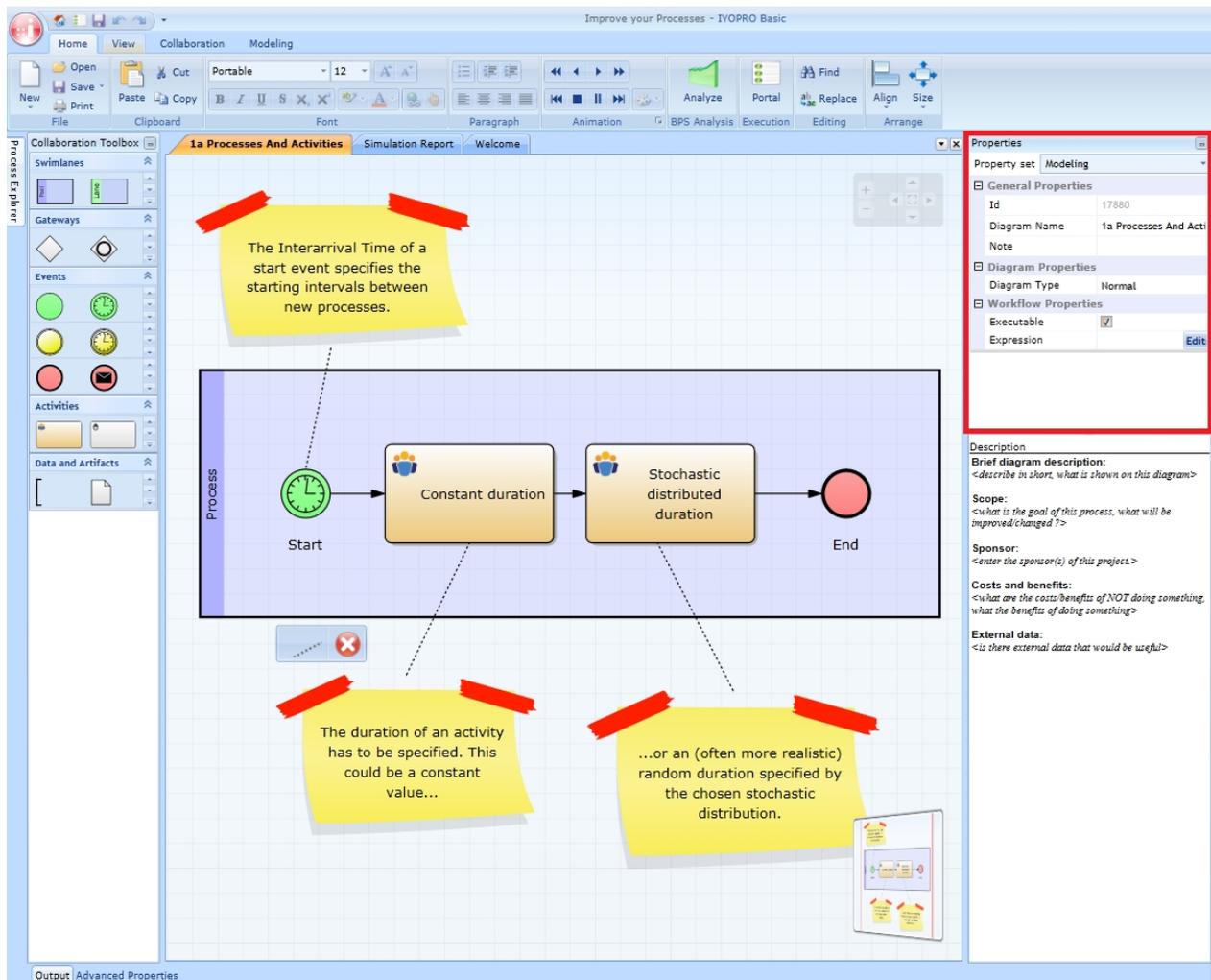


Figure 5: IYOPRO - The properties in the highlighted area

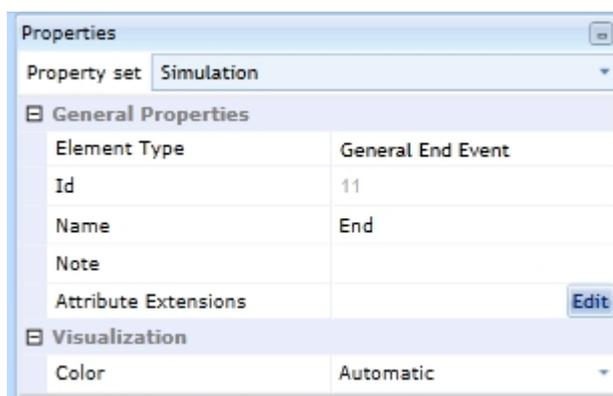


Figure 5: Simulation Properties of an end event

## Starting the Simulation

### Goals of this lesson

- More detailed insight into the starting procedure of the simulation
- Knowledge of the concept of stochastic seed

### How-to

- Prior to starting the simulation, the basic settings have to be considered.

- Click on the free space in the drawing board
  - (Note: Ensure that the Property Set Simulation is selected.)
- Now the Properties window shows general simulation properties.
- Let's have a look at the Must-Have Properties first.

## Two different views

There are two possibilities to access the general simulation properties, but both are only available if you made sure that the Property Set Simulation is selected.

1. Click on the free space in the drawing board (See the first picture).
2. Click the Simulation tab at the top of IYOPRO (See the second picture).

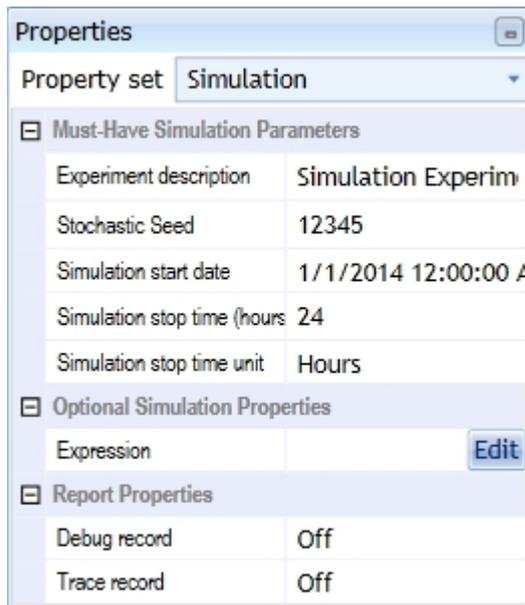


Figure 1: General Simulation Properties

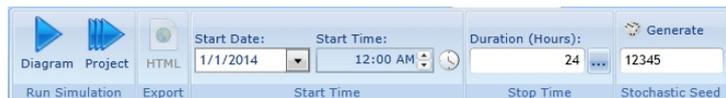


Figure 2: Another View of the General Simulation Properties

## Explanation

Both views contain the same information and options, although the simulation buttons (Diagram and Project) are only available in the latter view.

Anyways, in both views you can set a Start Date and the simulation's duration or stop time (click the button beneath the Distribution caption. You should go on and prolong the duration, and then check how the report's statistics have changed. Choosing the "right" simulation duration is crucial to the informative value of the report. Of course this "right" duration is depending on your purposes. Generally: The more the simulation's duration is increased the more the report's significance increases. But it obviously does not make sense to increase the simulation's duration, if the circumstances in the reality will not allow it.

## Stochastic Seed

We use the concept of stochastic seeds for the simulation being able to reproduce the results of experiments. The stochastic seed is a basic value for all stochastic distributed values in the simulation from where the random numbers are generated. Running the same experiment twice without changing any parameters will result in two identical results. By changing the stochastic seed between two simulation runs (and by keeping all other parameters unchanged), the results will be different. You can set the Stochastic Seed as an integer in the Must-Have Properties. You should just try it and then compare the reports. It is indispensable to run multiple simulations on a model, each with different general seeds.

## Local Stochastic Seed

Additional Information: You can assign an individual stochastic seed for every stochastic distribution. The individual seed is independent of the general seed. For example, giving the distribution of the second activity of Model 1a an individual seed will result in the same values taken from the distribution in every simulation, even though the general seed could have changed.

Please have a look at the section General [Simulation Properties](#) for more information on the other properties.

To start the simulation in IYOPRO you will either have to click the (green) Analyse button or the Play Diagram button in the simulation tab of the IYOPRO tab bar.

## Report

When the simulation has finished, you will get a report with a set of values for your experiment. See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 48
- Simulation stop time unit: Hours

## Report statistics

The report of this model consists of three parts:

- Process details
- Activity details
- Event details

## Note

Actually the report consists of four parts. The first part contains the model description and the settings of the experiment run. This enables you to reproduce the report by reconfigure the simulation with the given information.

### Process details

Here the report provides statistics for the processes' cycle times. They are visualized by histograms (Figure 1) and Box-plots (Figure 3). But there are also tables with the exact statistics (Figures 2 and 4).

Histogram for Processing time of Pool: Process#1 (started at event: Start#1)

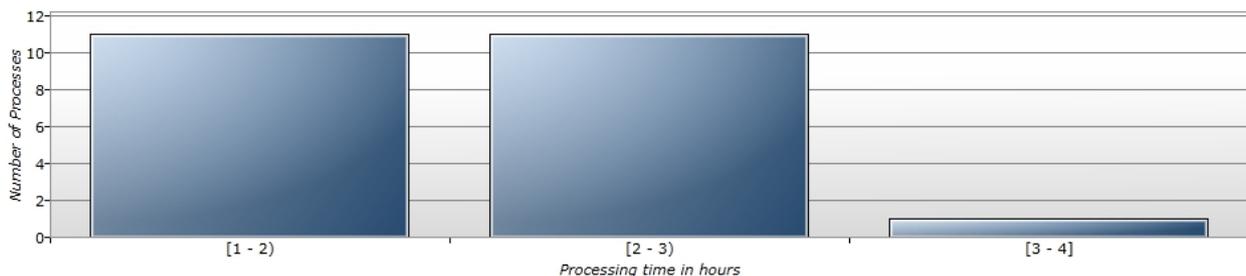


Figure 1: Histogram of the process' processing time

## Result

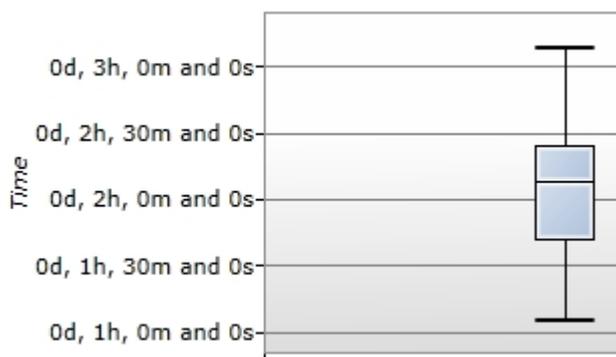
- The process cycle times from figure 2 deliver the statistics for the process' cycle time
  - We notice, that the cycle time's mean value is at around 2 hours and 2 minutes
    - with a standard deviation of about 29 minutes
  - The lowest cycle time amounts to 1 hour and 6 minutes (Min-Value)
  - The highest cycle time amounts to 3 hours and 9 minutes (Max-Value)
    - This is also represented in the histogram
    - There is only one process instance whose cycle time was higher than 3 hours

- The histogram classifies the finished process instances into time intervals
  - As you see, 22 process instances have a cycle time between 1 and 3 hours
  - There is only one outlier
- Additionally the box-plot from figure 3 tells us that the process instances' cycle times concentrate in a range between 1 hours and 30 minutes and 2 hours and 30 minutes
  - Also see the last 3 columns of the table from Figure 2 (Lower and upper quartile and median).
- The table from Figure 4 illustrates statistics about the number of simultaneously active process instances.
  - On average there was only one process instance active (Mean)
  - There have been at most 2 process instances active at the same time (Max)

Process cycle times

Title	Obs	Mean	Std.Dv	Min	Max	Lower Quartile	Median	Upper Quartile
Processing time of Pool: Process#1 (started at event: Start#1)	23	0d, 2h, 2m and 1s	0d, 0h, 29m and 10s	0d, 1h, 6m and 7s	0d, 3h, 8m and 54s	0d, 1h, 41m and 8s	0d, 2h, 7m and 12s	0d, 2h, 23m and 19s

Figure 2: The process' cycle time statistics



Title	Mean	Std.Dv	Min	Max
Coexistent instances of processes in Pool: Process#1	1.01683	0.44115	0	2

Figure 3: Box-plot visualization of the process' cycle times

Figure 4: The concurrently active process instances

## Activity details

In this part of the report you will find all the statistics of the activities.

### Activity run-times and waiting times

- We notice that "Constant duration" activity really was constant (1st line of the activity run-times and waiting times table from figure 5)
- Check out the "Stochastic distributed duration". It almost reaches our configuration of a normal distribution with a mean value of 1 and standard deviation of 0.5
  - Note that, the longer the simulation runs the more this statistic will converge to our actually set normal distribution
- Do you notice something else?
  - Yes, if you add up both activities' mean value you will get the process' mean cycle time as a result
  - The same goes for the standard deviation, minimum and maximum values
  - Unfortunately this only that simple, because our model is also very simple

### Concurrently existing activities

- We notice that there has been 0.5 activities of "Constant duration" active at the same time
  - How is this even possible? You cannot have half an activity active!?
  - These values are still just statistics for a certain period of time
  - Explanation
    - Our process starts every 2 hours
    - The "Constant duration" activity works constantly for 1 hour

- So, there is a difference of 1 hour
  - The constant activity finishes and the next process instance starts after 1 hour and will therefore reach the constant activity in 1 hour
  - That is why - in statistical matters - this activity is only active half of the time
- The same explanation applies for the stochastic activity

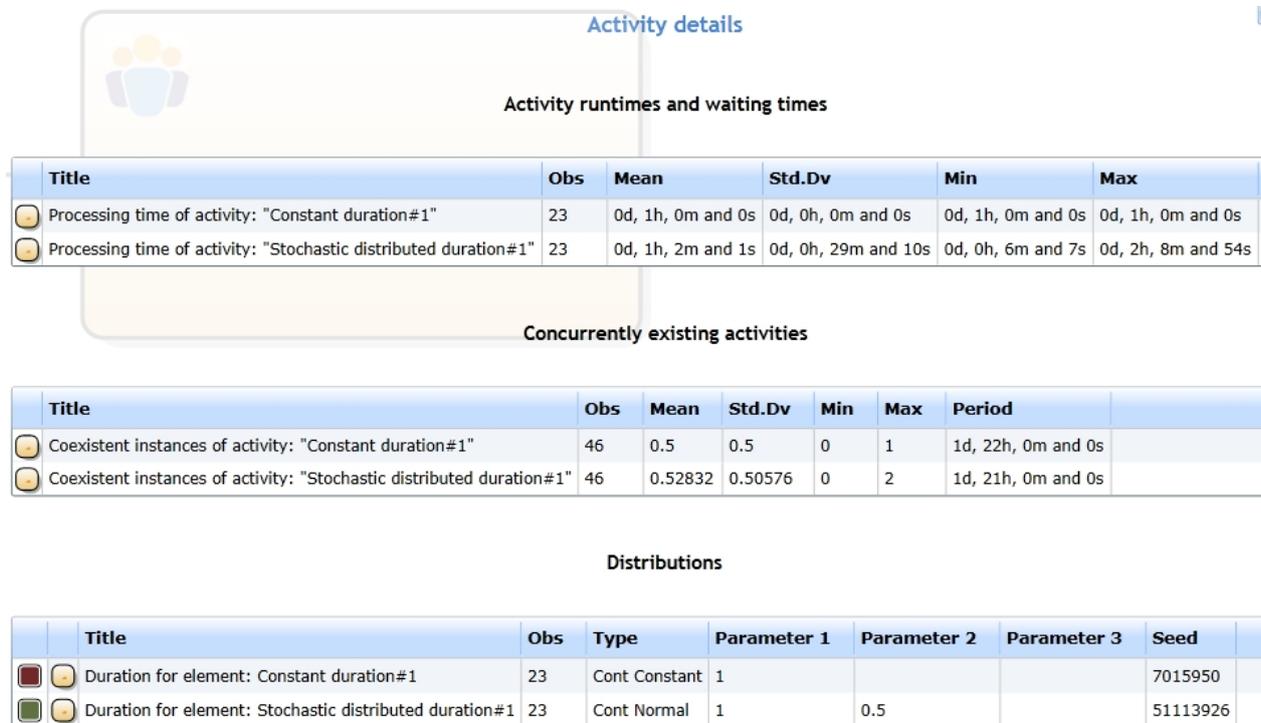


Figure 5: The report's activity details

## Distributions

- The Distributions table from Figure 5 simply shows the Distributions' configurations, which we set
- Also you can find the number of value drawings from the distribution (Obs)
- And its individual seed is also depicted in the table
  - Note that this seed depends on the general seed

## Event details

This part of the report presents the events' statistics, see Figure 6.

### Event counter

- In this table you can see how often an event has been executed (Obs)
- The end event of our process has been executed 23 times, which matches the number of finished process instances in the process details

### Inter-arrival times

- This table shows the inter-arrival times of events
  - This means, we count the time between the execution of the same event and then erect this statistic
- Our start event has an inter-arrival time of 2 hours
  - This matches with our configuration for the start event (Constant distribution with a value of 2)
- Our end event's inter-arrival time has a mean value of 1 hour and 58 minutes
  - Please do not confuse yourself by thinking that the end event's inter-arrival time should be the same as the process' cycle time
  - These times' values are similar to each other, but they are mostly not equal



	Title	Type	Obs	Current Value	Min	Max
●	Triggered event "End#1"	Count	23	23	0	23

#### Interarrival Times

	Title	Obs	Mean	Std.Dv	Min	Max
●	Interarrivaltime for start event: "Start"	22	0d, 2h, 0m and 0s	0d, 0h, 0m and 0s	0d, 2h, 0m and 0s	0d, 2h, 0m and 0s
●	Triggered event "End#1" Interarrivaltime	22	0d, 1h, 58m and 4s	0d, 0h, 43m and 59s	0d, 0h, 16m and 6s	0d, 3h, 0m and 33s

#### Distributions

	Title	Obs	Type	Parameter 1	Parameter 2	Parameter 3	Seed
●	Interarrival time at Element: Start#1	24	Cont Constant	2			82730829

Figure 6: The report's event details

## Model 1b - Time Consuming Sequence Flows

### Goals of this lesson

- Learn when and how to use time consuming sequence links

In functional models it is advantageous to hide trivial actions for clarity. This feature is shown in the picture below: using sequence links with time consumption. For example, this is reasonable for simulating the duration of the daily in-box. You could provide an additional activity, that models the in-box process. But for better clarity and to direct the viewer's focus to the actual activities you use the sequence link's duration property.

### How-to

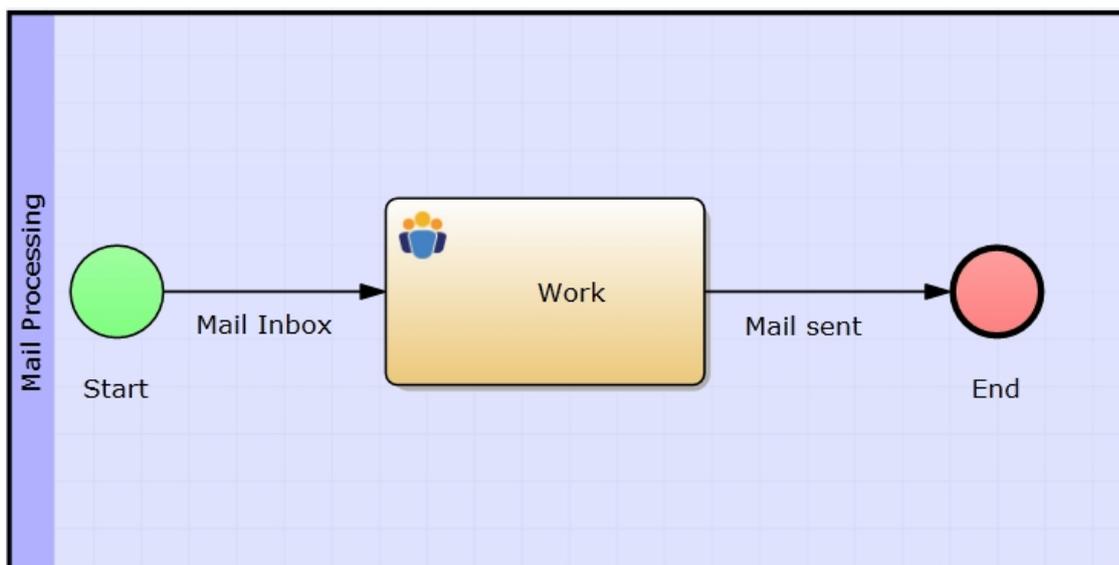


Figure 1: Model 1b - Time consuming Sequence Flows / [IYOPRO-Link](#)

### Approach

- 1) Make sure, you selected the Simulation option from the Property Set drop-down-box.
- 2) Click on the specific sequence link (in this case the one named "Mail In-box")
- 3) Now click the button right of the empty field of the Duration option under the Optional Properties
- 4) The Time Editor will show up
  - a) Select the Poisson Distribution
  - b) Set a Mean value of 2
  - c) Close the Time Editor by clicking OK
- 5) Repeat the same procedure with the "Mail sent" edge
  - a) Set up an Triangular Distribution with the values: Lower = 1, Upper = 2 and Peak 1.8

The report will show this duration in the activities section and it will provide the same statistical informations.

### Note

The General Start Event is used equivalent to the Timer-Start-Event.

### Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

### Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 48
- Simulation stop time unit: Hours

### Report statistics

Note that we will only cover the newly added statistics.

- Have a look at the activity run-times from figure 1
- We find, that each of the two time consuming sequence flows are now listed in the activity details
  - This is because the simulation supposes that such a sequence flow is an additional activity
  - Therefore you can find the same statistics as you would find for an activity



Title	Obs	Mean	Std.Dv	Min	Max
Processing time of activity: "Leadtime for edge: Mail Inbox#1#1"	14	0d, 2h, 0m and 0s	0d, 1h, 14m and 25s	0d, 0h, 0m and 0s	0d, 4h, 0m and 0s
Processing time of activity: "Leadtime for edge: Mail sent#1#1"	13	0d, 1h, 34m and 24s	0d, 0h, 14m and 53s	0d, 1h, 11m and 47s	0d, 1h, 56m and 35s
Processing time of activity: "Work#1"	13	0d, 2h, 50m and 20s	0d, 0h, 53m and 15s	0d, 1h, 12m and 14s	0d, 3h, 49m and 23s

Figure 1: Activity run-times and waiting times

### Model 1c - Time Instants for Start Events

#### Goals of the lesson

- Knowing what a recurrent time instant is and how it is being set up

#### How-to

Model 1c accords with Model 1a. The only difference is that we use Recurring Time Instants to determine the start event's inter-arrival time.

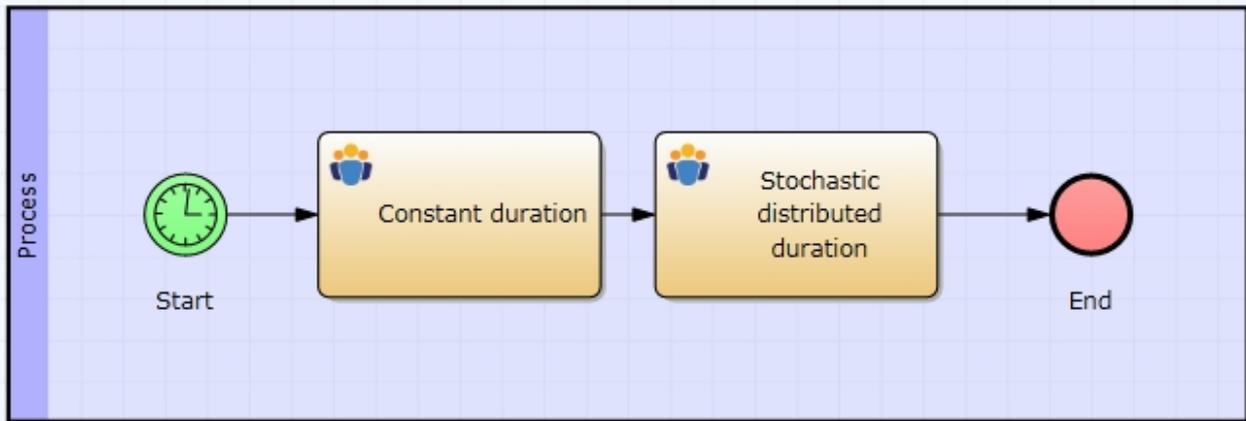


Figure 1: Model 1c - Time Instants for Start Events / [YOPRO-Link](#)

### Approach

To setup the Recurrent Time Instant for this diagram do the following:

1. Make sure, you selected the Simulation option from the Property Set drop-down-box.
2. Click the Timer Start Event
3. Click the button right of the empty field of the Inter-arrival Time option under the Must-Have-Properties
4. The Time Editor shows up. Now select Time Instant from the drop-down-box in the top left corner of the Time Editor.
5. In the drop-down-down-box right to the caption "1. Recurrence" select Hourly, which means that every hour exactly one time instant will be scheduled (respectively the time instant will recur every hour), which will then trigger the start event.
6. Now select an end (or stop) criteria. We will leave it to Open end, which signalizes the simulation, that the time instant will recur every hour until the simulation has stopped. But we could also specify a stop criteria
  - a) by amount: The time instant will stop recurring after the given number
  - b) by time: The time instant will stop recurring after the given time is exceeded

The picture below depicts the Time Editor in Recurring Time Instant mode after we configured it. Note that the Time Editor offers you a consecutive recurrence when you select one of the latter stop criteria.

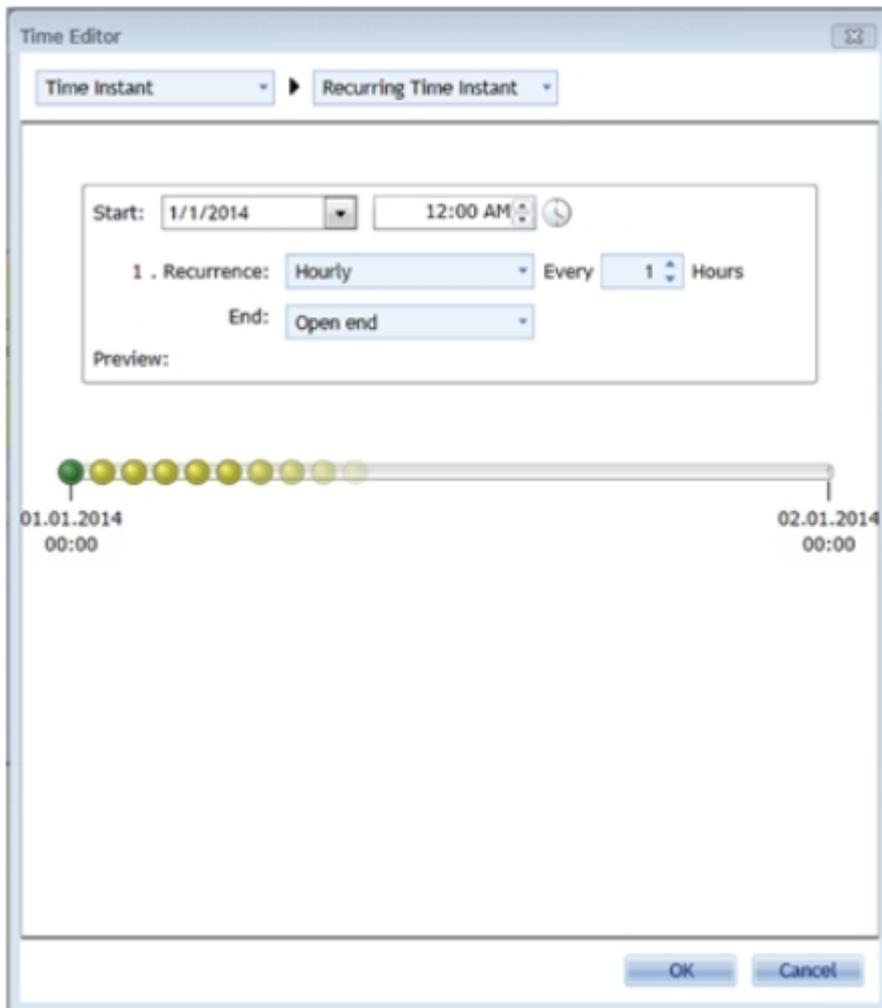


Figure 2: IYOPRO's Time Editor in Time Instant Perspective

## Report

See below for the run configuration, which we set up to receive the report statistics from below.

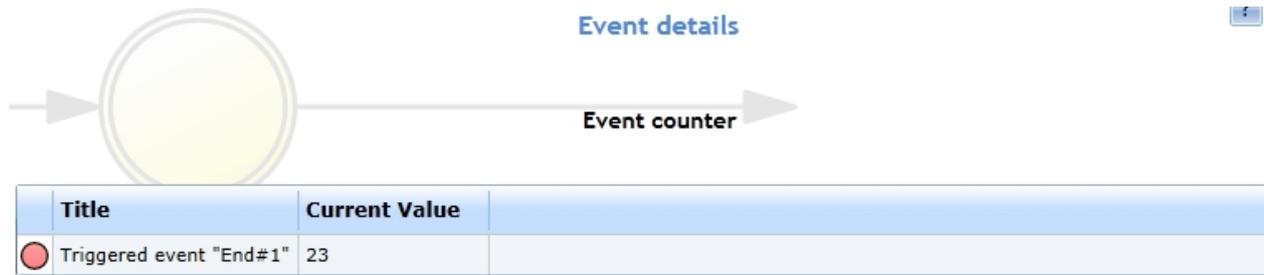
Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 24
- Simulation stop time unit: Hours

## Report statistics

- Did you notice that this report's are equal to Model 1a's report? (Except for the event details).
- Though we shortened the total simulation time to 24 hours (instead of 48 hours at Model 1a), the results are the same
  - This bases on the shorter inter-arrival time of Model 1c's start event
  - Model 1c's start event has a mean inter-arrival time of exactly 1 hour
  - This is the value we assigned to it by the calendar-based time instant which recurs every hour
  - Therefore number of processes is the same as in Model 1a



### Interarrival Times

	Title	Mean	Std.Dv	Min	Max
🕒	Interarrivaltime for start event: "Start"	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s
●	Triggered event "End#1" Interarrivaltime	0d, 0h, 58m and 51s	0d, 0h, 33m and 28s	0d, 0h, 5m and 58s	0d, 2h, 0m and 33s

Figure 1: Event details

## Model 1d - Scripted Time Instant for Start Events

### Goals of the lesson

- Learn how set up a simple scripted time instant to determine a start event's inter-arrival time
- This is only used, if timing is very complicated, e.g. it depends on complex computations. This example is very trivial, and could be achieved more easily with [Time Instants](#). But maybe, you want to substitute the '9' with a variable, which is the result on an individual algorithm, you programmed in Python language

### How-to

Model 1d accords with Model 1a. The only difference is that we use a Scripted Time Instant to determine the start event's inter-arrival time.

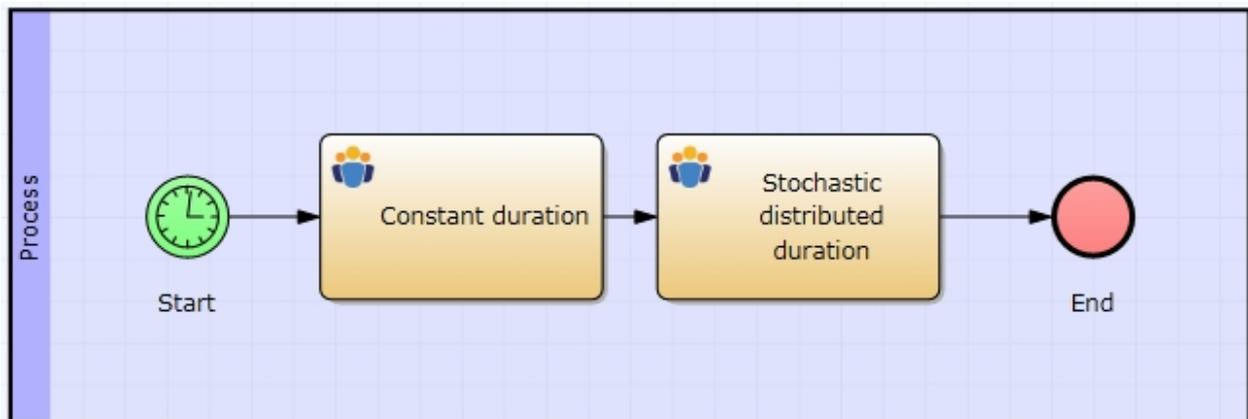


Figure 1: Model 1c - Time Instants for Start Events / [YOPRO-Link](#)

### Approach

- Click the start event (make sure you selected simulation from the property set)
- Open up the Time Editor to set up the inter-arrival time of the start event
- Now select Time Instant from the drop-down-box in the top-left corner of the time editor
- From the second drop-down-box (right to the previous one) select Scripted Time Instants
- The Time Editor now contains an input-box, in which you can write your Python code
- Put this line into the text area: `Result=SimulationContext.Time.GotoHour(9);` (see figure 2)
  - This tells the simulation to advance to 09:00 o'clock when the event is being initialized by the simulation
  - Note: If the simulation time is past 09:00 o'clock it will advance to the next day's 09:00 o'clock

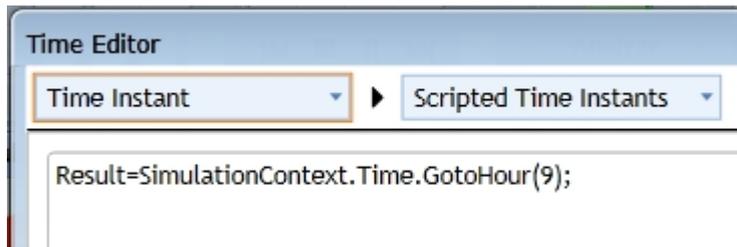


Figure 2 - Assigning the starting time for the start event

### Explanation

- As already said, this process accords to those of Model 1a and Model 1c.
- The only difference is that we used a Scripted Time Instant to determine that the process starts every day a 9 o'clock.
- Note that its inter-arrival time is 24 hours at the exact same hours each day.

### Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

### Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

### Report statistics

- Contrary to the former configurations we defined a simulation time of 240 hours (10 days)
- Our start event fires only one time per day (always at 9 a. m.)
- Therefore the process will start ten times (see Figure 1 below) within the simulation time

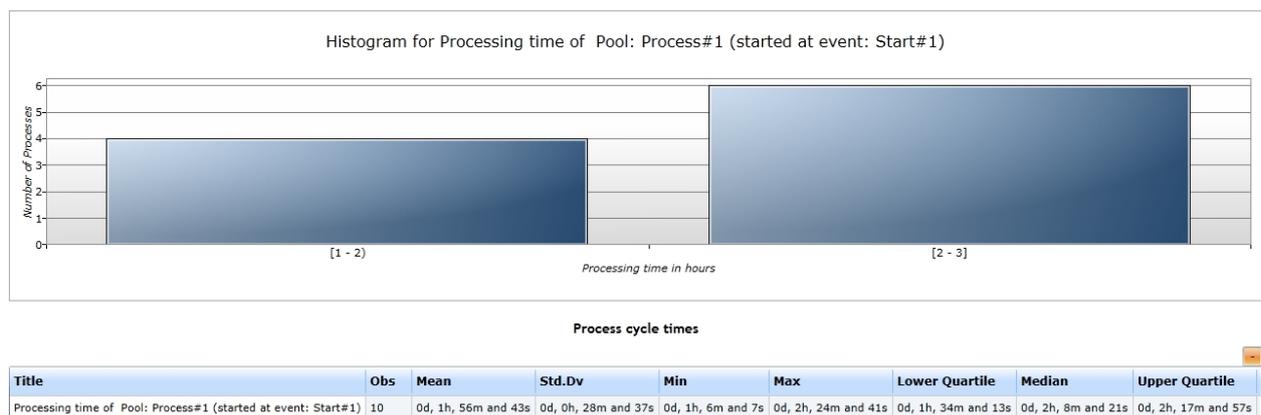


Figure 1: Process cycle times

## 2. Gateways & Process Variables

This section deals with Gateways and Process Variables, which are needed for the decision making process at exclusive and inclusive gateways.

### Content

- [Model 2 - Parallel Gateways](#): Comments to the parallel execution of activities under scope of simulation.

No special simulation settings to be made here.

- [Model 3a - Exclusive Gateways](#): Describes how conditions can be set, which specify the next sequence path.
- [Model 3b - Assignment of Stochastic Process Attributes](#): Describes how to use process variables, which change their values stochastically, for decision making at gateways
- [Model 3c - Inclusive Gateways](#): Describes how parametrize variables and conditions for decision making at inclusive gateways
- [Model 3d -Data Objects](#): Describes how data objects are used to transfer data to a specific activity
- [Model 3e - Data Stores](#): Describes how data stores can be used to store data and to transfer data across pools

## Model 2 - Parallel Gateways

### Goals of this lesson

- Getting to know how the semantics of the parallel gateway work

### How-to

This model shows a business process with parallel gateways. When the sequence flow hits a branching parallel gateway, it starts a sub-process for every outgoing link

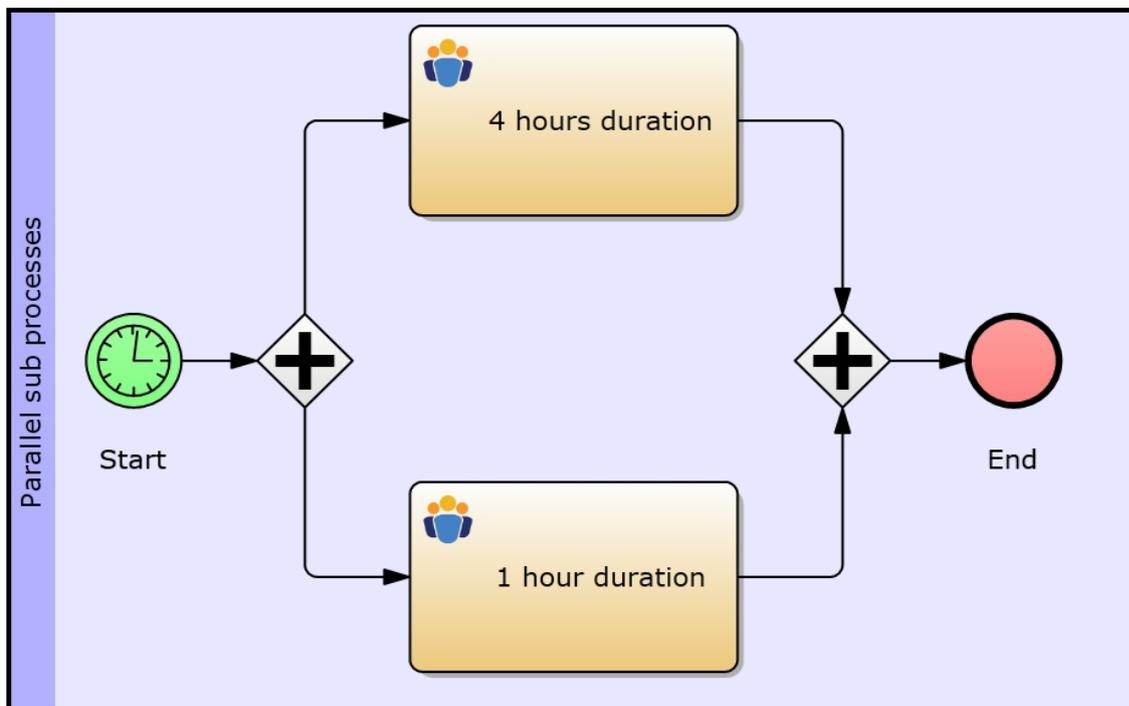


Figure 1: Model 2 - Parallel Gateway / [YOPRO-Link](#)

### Explanation

- In the model (in the illustration above) the first gateway starts two sub-processes (or analogical two tokens will be send on the path), that will be merged in the second gateway.
- Note that the merging gateway will not fire until both sub-processes (tokens) have arrived.
- Model 2 defines a constant duration of 4 time units (TU, hours in this case) for the upper activity and a constant duration of 1 TU for the lower activity.
- This means, the bottom sub-process has to wait 3 TU until it is merged by the gateway.
- The total lead time (or processing time) of the business process will always be 4 TU.

### Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

### Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

### Report statistics

- As you might notice, the process cycle time is always 4 hours
- There is no standard deviation because we set up constant durations for both activities
- And as mentioned [before](#), this model's process cycle time depends on the activity with the longest duration
  - The merging parallel gateway has to wait for all sequence flows to arrive
  - That is why the process cycle time is always 4 hours, which is the top activity's duration
- Also note: The box-plot from Figure 1 below is nearly invisible.
  - It is reduced to a simple line, because all the statistics are equal (Min-, Max-value, etc.)

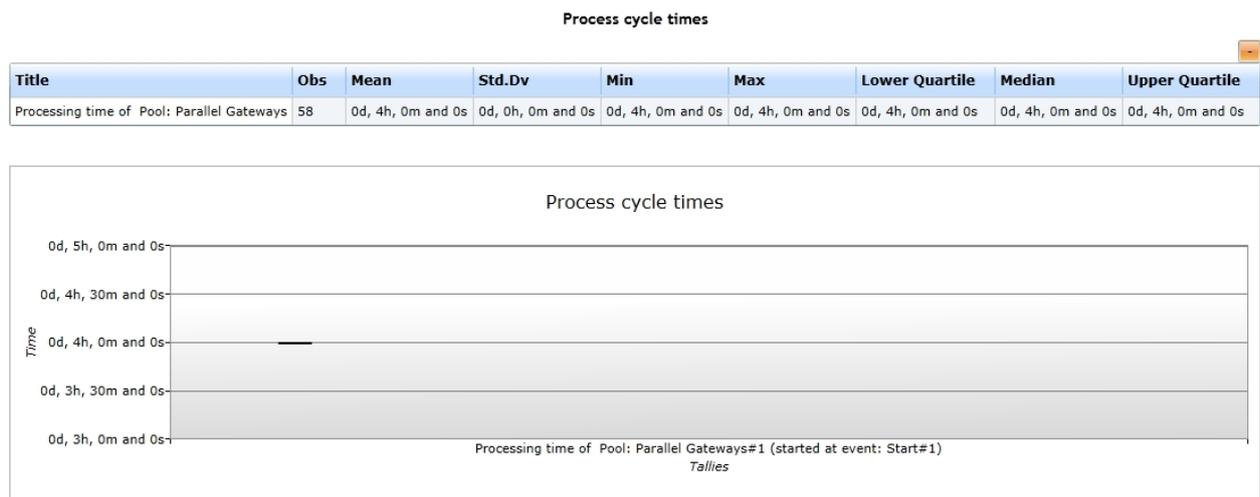


Figure 1: Process cycle times of Model 2

## Model 3a - Exclusive Gateways using Conditions

### Goals of this lesson

- Usage of exclusive gateways

### How-to

Model 3.a explains how to use exclusive gateways for a simulation. Contrary to parallel gateways a branching exclusive gateway needs a condition on every outgoing sequence flow depending on (a) process variable(s). During the simulation the conditions on the outgoing edges will be evaluated to determine exactly one edge as the next path element. You have to ensure that the conditions will determine exactly one sequence flow, otherwise (two or more edges meet the conditions) the simulation results will be useless due to this modeling fault.

Please refer to section [Variables](#) for a description of how to set up variables and conditions.

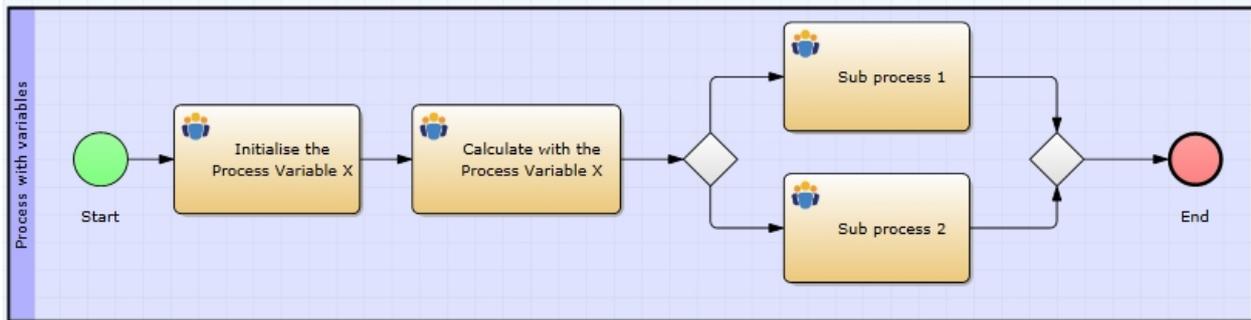


Figure 1: Model 3a - Exclusive Gateway / [IYOPRO-Link](#)

## Approach

Depending on our assignment of variables and conditions from the lessons [Assigning Process Variables](#) and [Assigning Gateway Conditions](#), our variables and conditions look like this:

- For the model above we will define the variables x and y at the first activity
  - $x=1$ ; which defines the variable x with a value of 1
  - $y=1$ ; which defines the variable y with a value of 1
- At the second activity We will now increment x by one and define the new variable z that takes its value from y
  - $x=x+1$ ; which increments x's value by 1
  - $z=y$ ; which defines a new variable z that will be given y's value
- Our variables have now these values:  $x = 2, y = 1, z = 1$
- Now declare the edge's conditions behind the branching exclusive gateway:
  - Upper edge:  $x < y$
  - Bottom edge:  $x > y$

## Explanation

This configuration is quite static and always leads to the result, that the bottom sequence flow is being followed, and therefore the activity "Sub process 1" will never be executed. Simply simulate the model with this configuration and you will notice that the activity "Sub process 1" does not contain any statistical information.

## Attention

- None of the edges' expressions test the equality of x and y.
- This is a common error in this type of condition.
- If x and y are equal, the gateway will not fire.
- It will not pass any token, which means that the process instance is trapped at the gateway.
- So, we will need to check for equality.
  - Either by adding a third edge that checks for  $x == y$
  - or by extending one of our already existing edges' condition to  $x >= y$  or  $x <= y$ .
  - Note that the latter can only be done for only one (edge's) condition, because we must impose the exclusive restriction (an exclusive branching gateway must pass exactly one token!)

## Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240

- Simulation stop time unit: Hours

## Report statistics

- In this report you will find a new section in the report: Gateway details
  - Note: When a model only contains parallel gateways, there will be no gateway details in the report
    - Because there is no active decision making, which bases on any kind of information
- Figure 1 shows the report's gateway details
  - As you see, there was only one sequence flow that has been followed after the branching exclusive gateway
  - This is precisely the circumstance, which we defined in our [decision making](#)
- Also if you look at Figure 2 further below, you will notice that the activity "Sub process 1#1" has never been reached
  - Just like we intended to
- Since this model's gateway details are not very extensive, have look at [Model 3b's report](#).

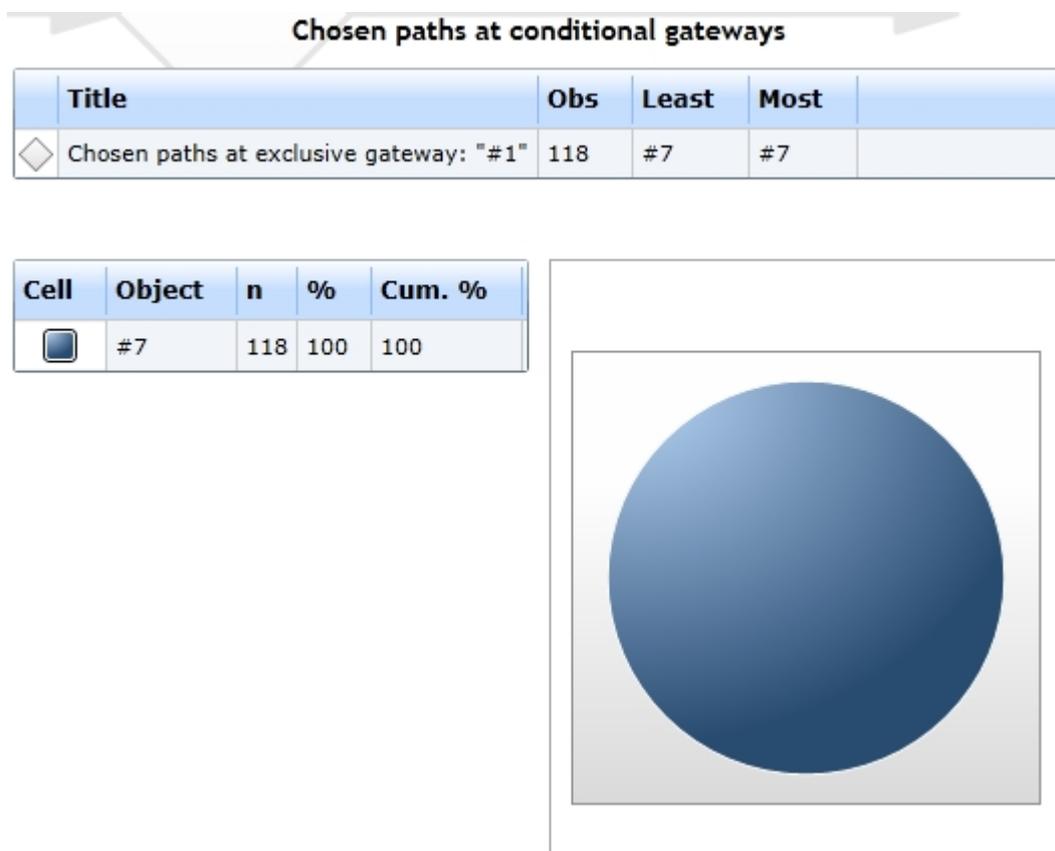


Figure 1: The report's gateway details



Figure 2: The report's activity run-times

## Model 3b - Assignment of Stochastic Process Variables

### Goals of this lesson

- Getting more dynamics into the decision making at exclusive or inclusive gateways
- Usage of stochastic process variables

## How-to

The exclusive gateway in Model 3.a from lesson [Exclusive Gateways](#) is pretty static. During the simulation every process instance will use the same path, because the evaluation will always get the same result. Models with more dynamic (and realistic) path selection can be achieved by defining stochastic attributes. Stochastic attributes are variables, whose values change (similarly to the duration of activities) by a stochastic distribution.

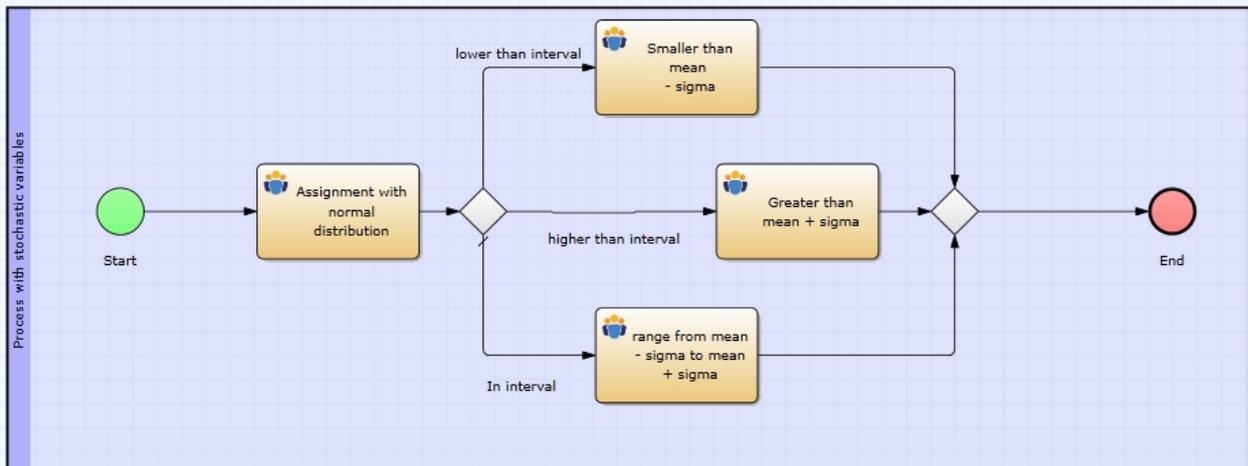


Figure 1: Model 3b - Assignment of Stochastic Attributes / [YOPRO-Link](#)

## Creating Stochastic Attributes

To use a stochastic variable, we need to define it at the activity before the gateway:

1. Make sure, you selected the Simulation option from the Property Set drop-down-box
2. Click the first activity in the model
3. Click the button right to the Stochastic Attribute Assignment property under Optional Simulation Properties
4. The Process Attribute Editor shows up (see the picture below)
5. Click the Add button to add a new stochastic attribute
6. The Distribution Editor shows up. You will notice, that it is the same Distribution Editor that you already got to know by assigning durations of activities or inter-arrival times of start events.
7. Now select the normal distribution with a mean value of 10 and a standard deviation of 3
8. The variable will be given a name (something like "v1"). You can change this by double-clicking into the Name field. Rename it to X

## Assigning Conditions with Stochastic Attributes

We are now ready to go. Simply define the conditions (see lesson [Assigning Gateway Conditions](#)) like this:

- Upper edge:  $\text{Result}=(X < 7.0)$ ; which means that the variables value is below the mean minus the sigma (standard deviation,  $10 - 3 = 7$ )
- Middle edge:  $\text{Result}=(X > 13.0)$ ; which means that the variables value is beyond the mean plus the sigma (standard deviation,  $10 + 3 = 13$ )
- Bottom edge: Put in on default by selecting Default Sequence Flow from the Element Type in the General Properties. This means the value is between the mean plus / minus the sigma ( $7 < X < 13$ ).

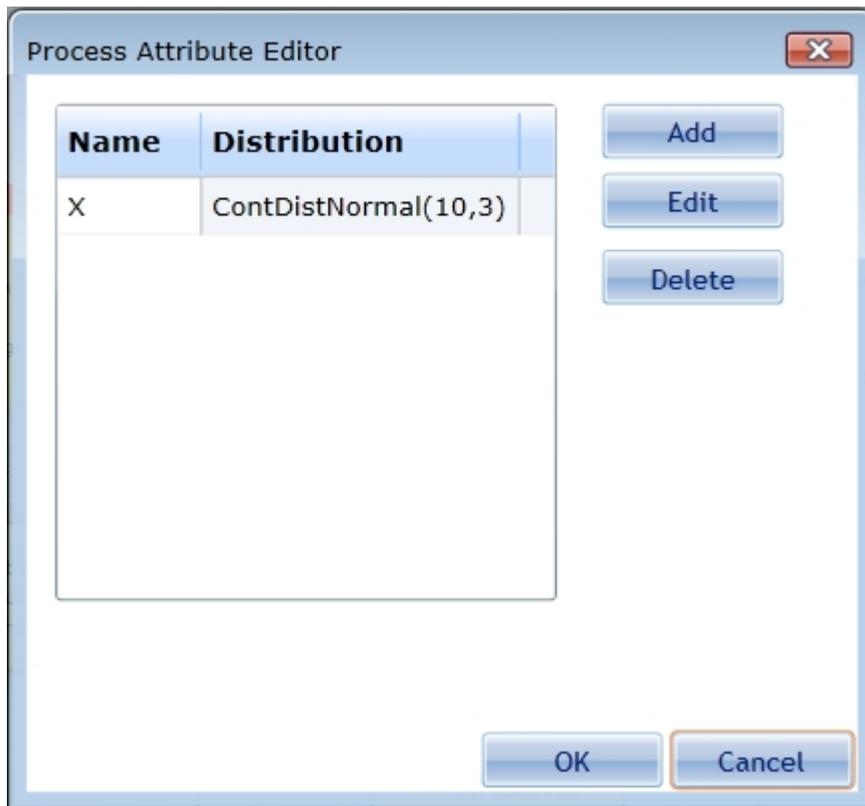


Figure 2: YOPRO'S Simulation Process Attribute Editor

## Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

## Report statistics

- This report contains a beautiful pie chart, which almost resembles the stochastic correlations of the normal distribution, which are:
  - In 70% of cases the value is within the assigned interval (mean +/- the standard deviation)
  - in 15% of cases the value is below the assigned interval
  - in 15% of cases the value is above the assigned interval
  - You can verify this by extending the simulation's run-time significantly
    - The percentages should converge to 70, 15 and 15
- In the second table you can find the statistics in numbers, which also resemble the facts from above

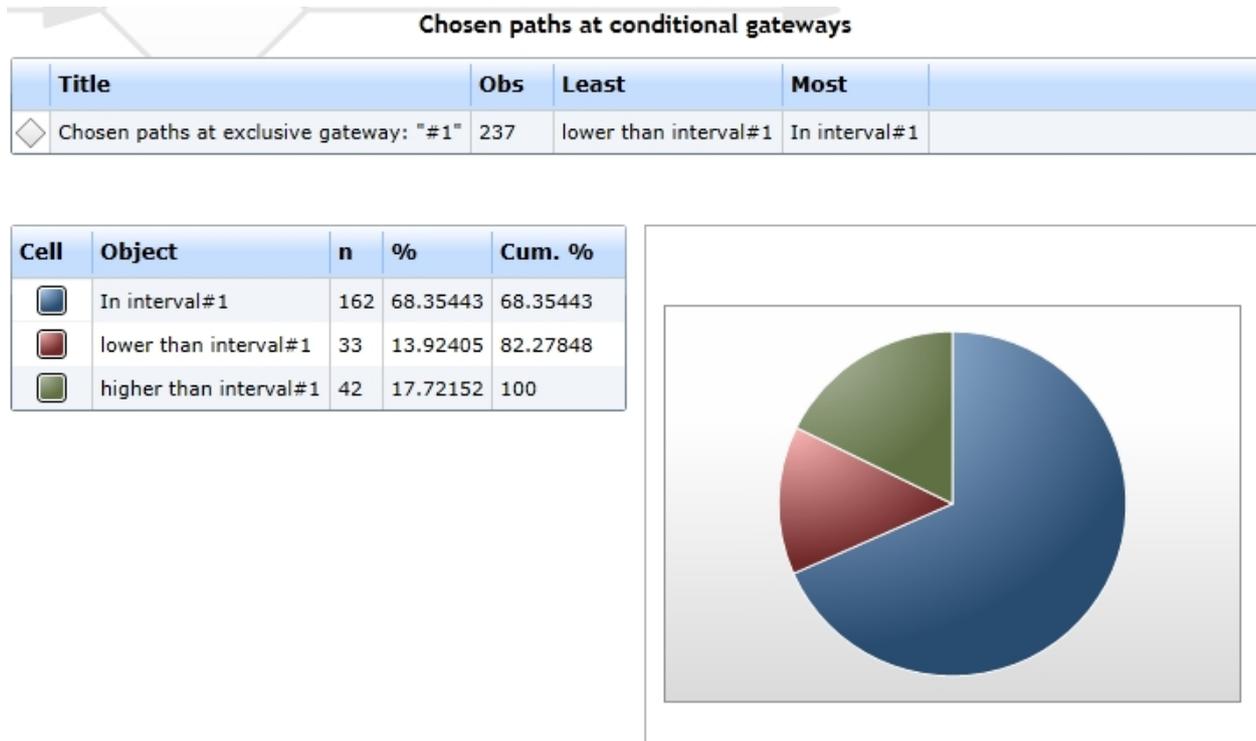


Figure 1: The report's gateway details

## Model 3c - Inclusive Gateways

### Goals of this lesson

- Usage of inclusive Gateways

### How-to

Contrary to exclusive gateways the conditions in inclusive branching gateways do not have to be exclusive.

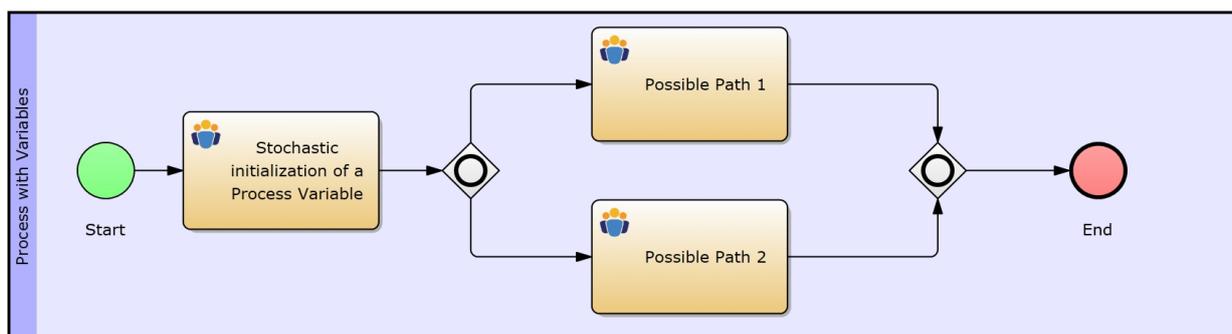


Figure 1: Model 3c - Inclusive Gateways / [IYOPRO-Link](#)

1. First create a stochastic attribute as explained in lesson [Assignment of Stochastic Process Variables](#)
  - a) Call the attribute X and assign it with a Discrete Uniform Distribution with a lower bound of 1 and an upper bound of 3
  - b) This will represent the decision basis for the inclusive gateway in the model
2. Set these conditions:
  - a) Upper edge:  $\text{Result}=(X \geq 2)$ ;
  - b) Lower edge:  $\text{Result}=(X \leq 2)$ ;

Note that (due to the equality) our conditions will both be met, when X is 2, which then will trigger two sub-processes being started on each sequence flow. The merging inclusive gateway therefore waits (similarly to the parallel gateway) until all (alive) sub-processes (tokens) arrive.

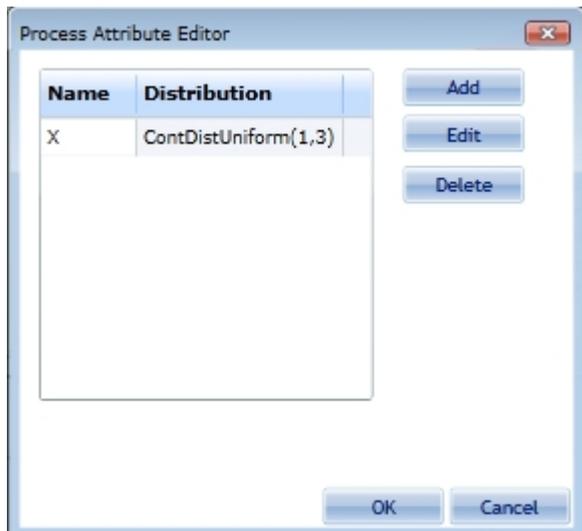


Figure 2: IYOPRO'S Simulation Process Attribute Editor

## Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

## Report statistics

- As mentioned [before](#), we configured the branching inclusive gateway such that in 33% of cases both sequence flows will be followed.
  - The process variable has been assigned a Discrete Uniform Distribution with a minimum value of 1 and a maximum value of 3
- The statistics from the figures (see below) make this correlation evident
- The process has been executed 114 times (Figure 1)
- The activity "Stochastic initialization of process variable" has been executed 118 times
- The gateway details (Figure 2) show us that the top sequence flow has been followed 78 times
  - and the bottom sequence flow 79 times
  - which sums up to 157 in total
- This means that we have got at least 39 (157 - 118) process instances, in which the gateway fires twice
- Also you can recognize from the activity details (Figure 3) that the top activity has been executed 74 times
  - and the bottom one 79 times
  - which sums up to 153 in total
- This tells us that in at least 39 (153 - 114 = 39) finished process instances the gateway fired twice
  - which is approximately 34% of all cases (just like we intended to)

Process cycle times

Title	Obs	Mean	Std.Dv	Min	Max	Lower Quartile	Median	Upper Quartile
Processing time	114	0d, 9h, 8m and 58s	0d, 3h, 53m and 33s	0d, 2h, 12m and 14s	0d, 14h, 49m and 56s	0d, 4h, 26m and 19s	0d, 11h, 18m and 51s	0d, 12h, 4m and 38s

Figure 1: Process Cycle Time

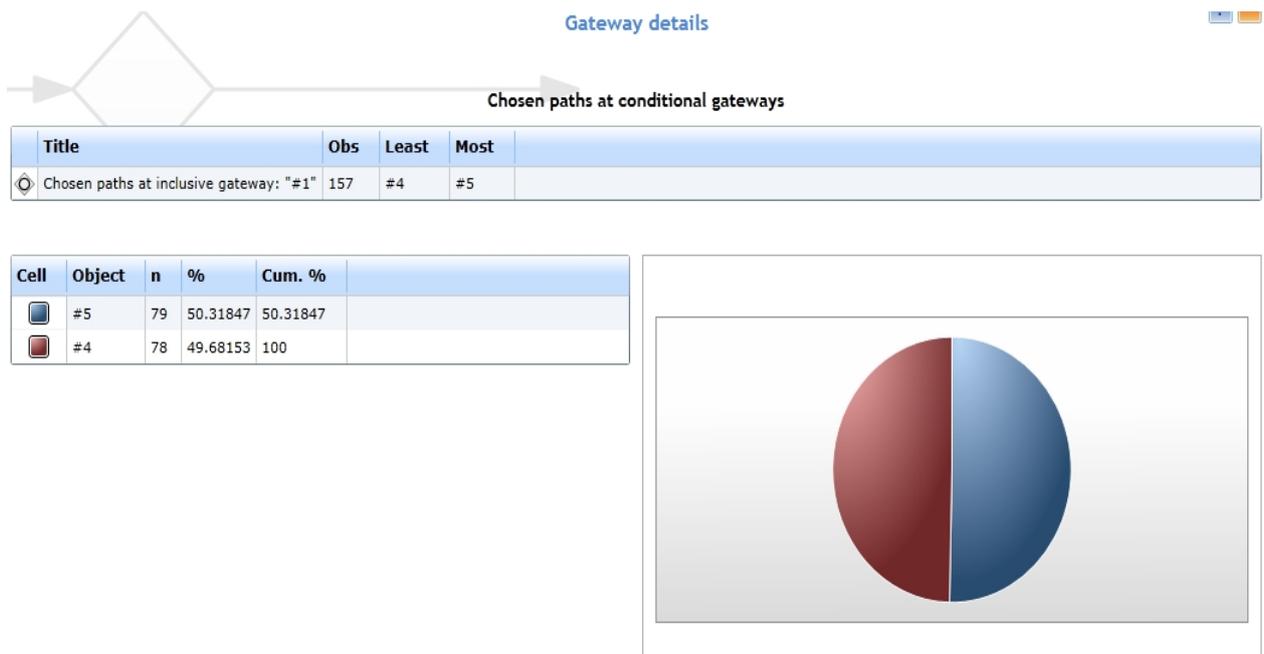


Figure 2: Gateway details



Figure 3: Activities' processing time

## Model 3d - Data Objects

### Goals of this lesson

- Usage of Data Objects (to transfer information)

### How-to

Model 3d uses a Data Object to transfer data to the process instance.

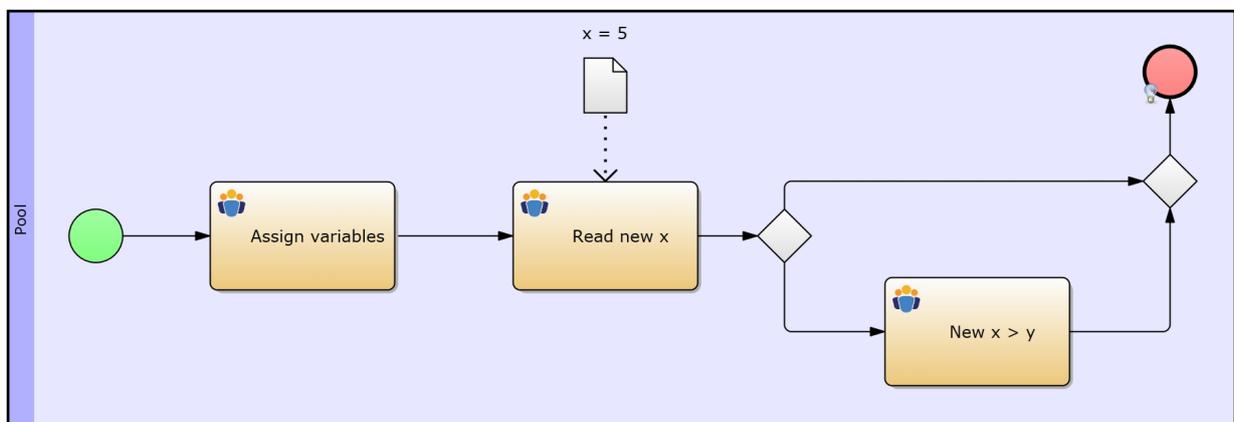


Figure 1: Model 3d - Data Objects / [YOPRO-Link](#)

### Approach

- The first activity defines the variables  $x = 3$  and  $y = 4$

- In the data object we define  $x = 5$  in the Expression field (see the highlighted area in the figure below)
- The Data Association from the data object to the second activity tells the activity to read data, that has been defined in the data object.
- Therefore the former  $x$ -value gets overwritten.
- To test this, we set up the conditions at the exclusive gateways like this:
  - Upper Edge:  $x < y$
  - Bottom Edge  $x > y$
- After running the simulation, the report will tell you, that  $x$  indeed has been overwritten.

Note: Of course you can also write data into a Data Object by inverting the Data Association's direction.

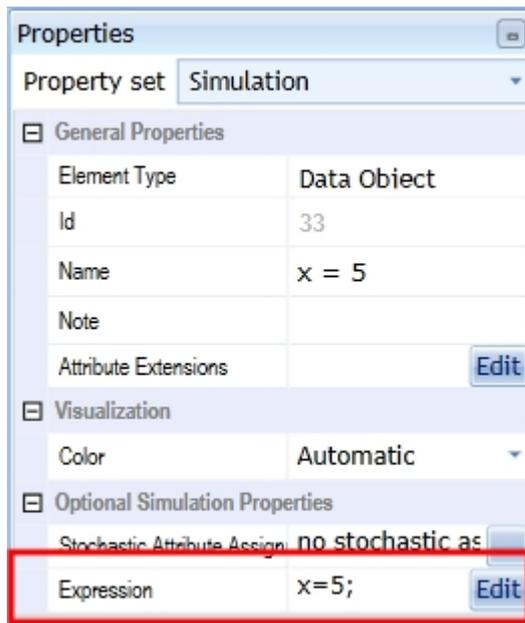


Figure 2: Simulation Properties of a Data Object (Expression property highlighted)

## Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

## Report statistics

- The process cycle times and the activity run-times (Figures 1 and 2) show us that the variable  $x$  indeed is bigger than  $y$ 
  - Also if you take a look at the gateway details (not listed here), you will notice that always the same edge has been followed behind the branching exclusive gateway
- Retrospective:  $x = 3$ ,  $y = 4$ 
  - But at the second activity we assigned  $x$  the value from the data object, which is 5
- 236 process instances have been completed throughout the simulation
- The number of completed "New  $x > y$ " also equals 236
- These numbers tell us that  $x$  is always greater than  $y$  (which is not really surprising, because we defined it that way)

### Process cycle times

Title	Obs	Mean	Std.Dv	Min	Max	Lower Quartile	Median	Upper Quartile
Processing time	236	0d, 3h, 0m and 0s	0d, 0h, 0m and 0s	0d, 3h, 0m and 0s				

Figure 1: The report's process cycle time

### Activity runtimes and waiting times

Title	Obs	Mean	Std.Dv	Min	Max
Processing time of activity: "Assign variables#1"	238	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s
Processing time of activity: "New x > y#1"	236	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s
Processing time of activity: "Read new x#1"	237	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s

Figure 2: The report's activity run-times

## Model 3e - Data Stores

### Goals of this lesson

- Usage of Databases
- Transferring data between pools by using Databases

Model 3e uses Data Stores to store data in a database and to transfer it across the pools' borders.

### How-to

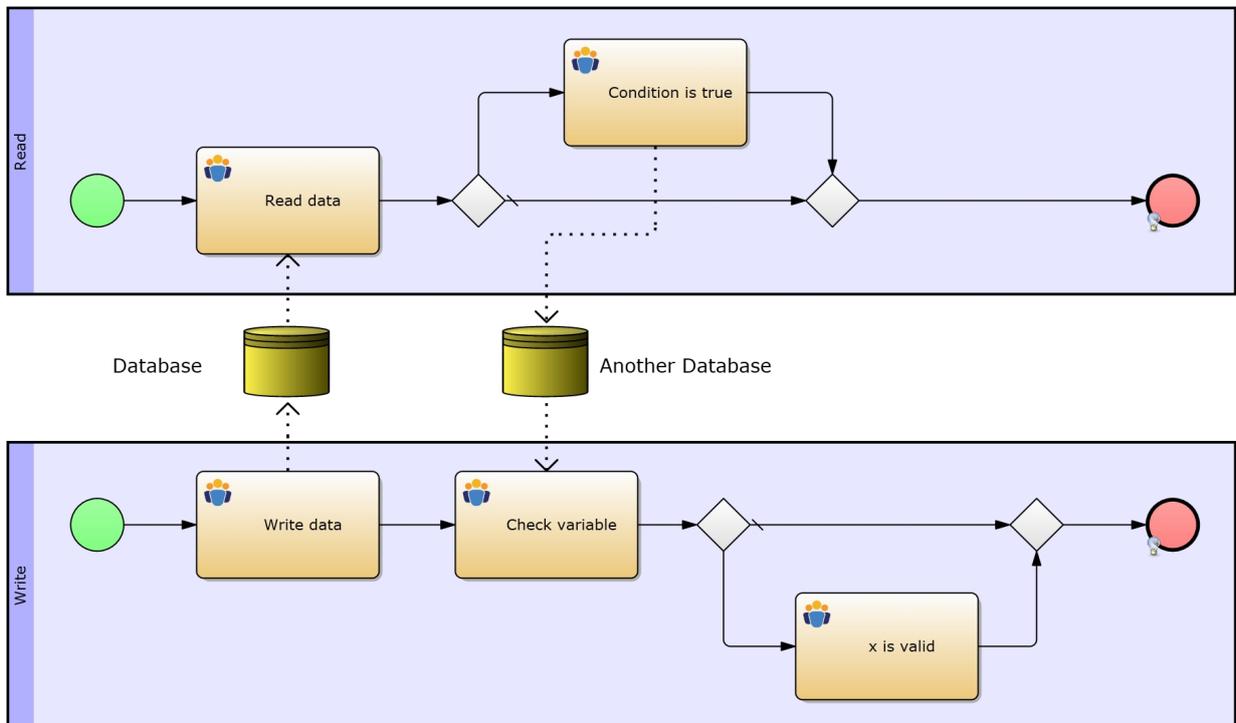


Figure 1: Model 3e - Data Stores / [IYOPRO-Link](#)

### Transferring Data between Pools

- In the first activity of the bottom pool we define  $x = 5$
- Afterwards we transfer  $x$  to the first Data Store by assigning  $x$  to the Data property of the Data Association (see the highlighted area in the figure below).

- Data Associations can (similarly to Message Flows) transport data.
- You simply need to enter the desired data's name into the Data property.
- x is now stored in the Database, and the first activity of the upper pool can read it.

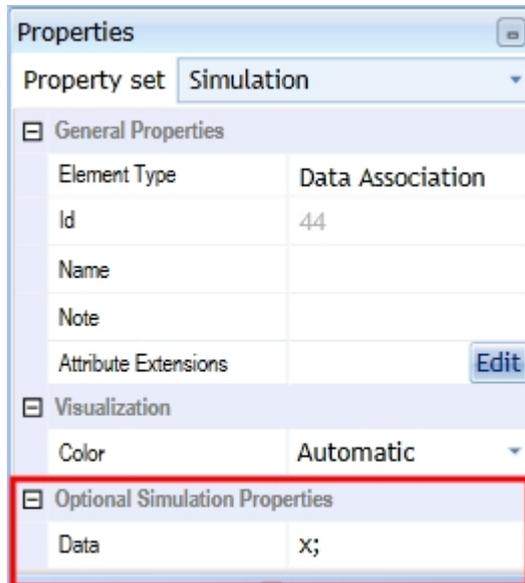


Figure 2: Simulation Properties of a Data Association - The Data property is highlighted

### Renaming data

As you might note (while examining the model on your own), we did not simply transfer x to the upper pool, but we also renamed it to y due to the AS qualifier in the Data Association's Data property (x AS y).

Transferring Data back to the original pool

- The upper edge behind the gateway has a condition defined:  $y == 5$ .
- If this is true, the upper pool's second activity sets  $y = 6$  and writes it to the second Data Store.
- Note that the variable also gets renamed back to x (y AS x).
- When the condition is not met, the process terminates.

### Using the new data

- Now the bottom pool's second activity reads the new x from the second Data Store.
- The process then uses it for decision-making at the gateway.
- The bottom edge's condition checks for  $x == 6$ . Is this the case, the third activity will be executed, otherwise the process just terminates.
- Since the new x is equal to 6, the third activity is being executed.

Extra Info: Using different Data Stores which resemble the same semantic Data Store

- By assigning Data Stores the same name, they become the same object.
- This can be very useful for extensive models.
- Imagine Model 3e to be much larger with one write / read operations at the beginning and at the end of the model.
  - By using just one Data Store you would have to draw Data Associations across the model, which could get very confusing (especially if you also use some Message Flows).
  - Therefore you can gain clearness by putting up one Data Store at the beginning and one at the end of the model, both with the exact same name.

### Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

## Report statistics

- By looking at the report, we note that our model has been simulated just fine
- Though, there is one single fact in the report, which does not add up at first glance: The bottom pool's gateway statistics
- Why did the gateway fire the default sequence flow twice?
- Please have a look at both start events' configuration:
  - The top pool's start event is being triggered every 2 hours
  - The bottom pool's start event is being triggered every hour
  - (Also check the activities' time consumptions)
  - This means
    - that the first two process instances of the bottom pool reach the branching exclusive gateway before
    - the top pool's first process instance completes the task "Condition is true"
    - and thus before it can write the new x into the data store "Another Database"
  - The first two process instances read from "Another database"
    - but they do not get any information for the variable x
    - because that data store has not been written to, yet

	Title	Obs	Least	Most
◇	Chosen paths at exclusive gateway: "#7"	238	#27	x ==6 #1

Cell	Object	n	%	Cum. %
	#27	2	0.84034	0.84034
	x ==6 #1	236	99.15966	100

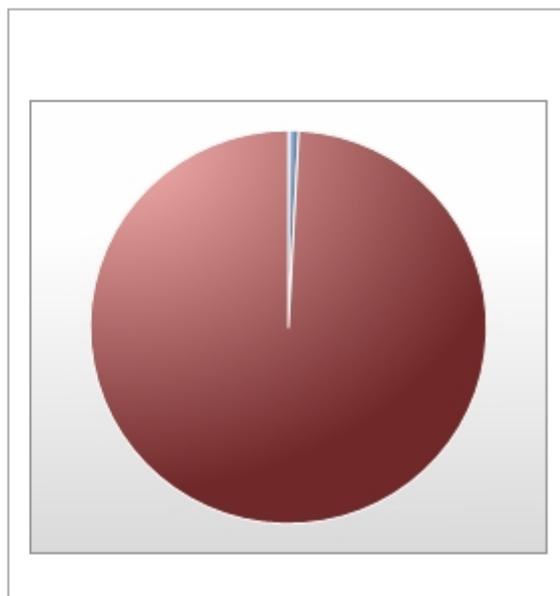


Figure 1: The Report's gateway details on the bottom pool's branching exclusive gateway

## Model 3f - Exclusive Gateway with Probabilities

### Goals of this lesson

- Usage of Exclusive Gateways with Probabilities instead of process variables

Model 3f uses probabilities to determine which sequence flows shall be pursued.

## How-to

Instead of declaring [process variables](#), we can simply use probabilities (or weighs) for each sequence flow at an exclusive gateway to determine the sequence flow to be pursued. Please have a look at the section [Using Probabilities](#) from the [Gateway Decision Making](#) chapter for more information.

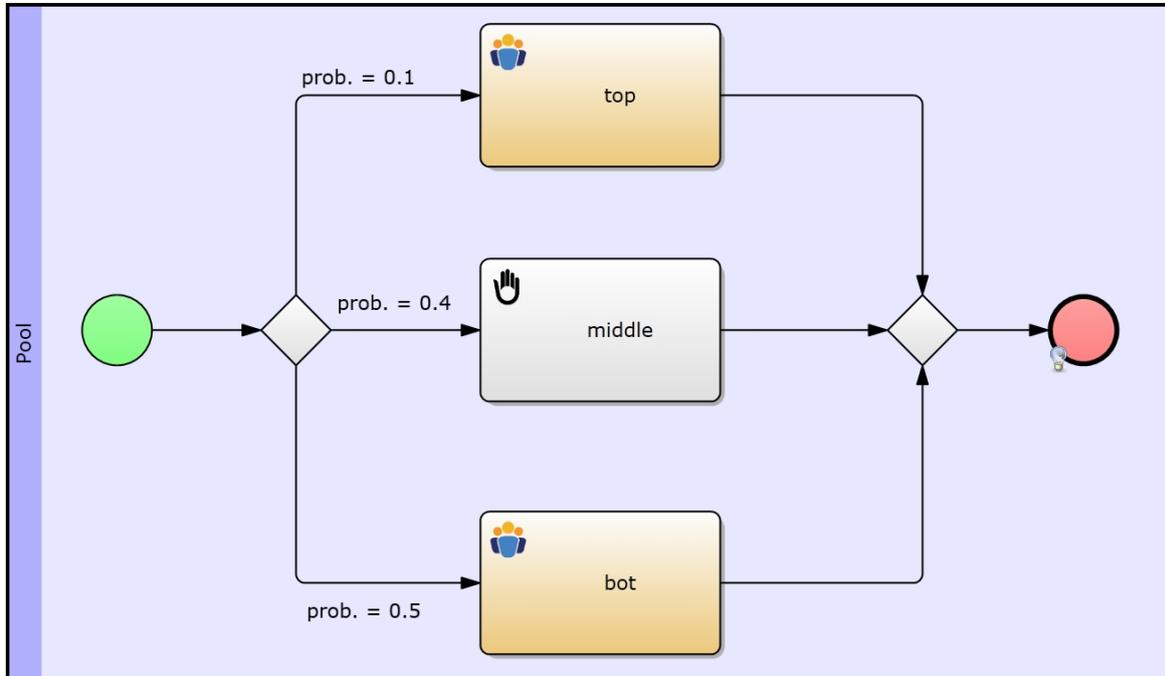


Figure 1: Model 3f - Exclusive Gateway with Probabilities / [YOPRO-Link](#)

## Approach

- Click on each of the 3 sequence flows successive to the exclusive gateway and supply the following values to its Probability property:
  - Top sequence flow: Probability = 0.,1
  - Middle sequence flow: Probability = 0.,4
  - bottom sequence flow: Probability = 0.,5
- Remember that the sum of the sequence flows' probabilities must be equal to 1

## Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

## Report statistics

- Have a look at this report's gateway details (see Figure 1)
- You will notice that the probability configuration is reflected in the gateway statistics

### Chosen paths at conditional gateways

Title	Obs	Least	Most
Chosen paths at exclusive gateway: "#2"	239	prob. = 0.1#1	prob. = 0.5#1

Cell	Object	n	%	Cum. %
	prob. = 0.1#1	25	10.46025	10.46025
	prob. = 0.5#1	123	51.46444	61.92469
	prob. = 0.4#1	91	38.07531	100

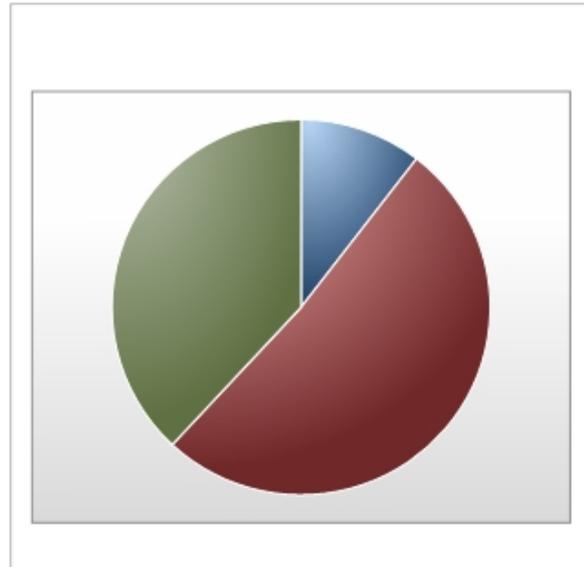


Figure 1: The report's gateway details

### Model 3g - Inclusive Gateway with Probabilities

#### Goals of this lesson

- Usage of Inclusive Gateways with Probabilities instead of process variables

Model 3g uses probabilities to determine which sequence flows shall be pursued.

#### How-to

As in [Model 3f](#) we will use probabilities (or weighs) for each sequence flow at an inclusive gateway to determine the sequence flows to be pursued. Please have a look at the section [Using Probabilities](#) from the [Gateway Decision Making](#) chapter for more information.

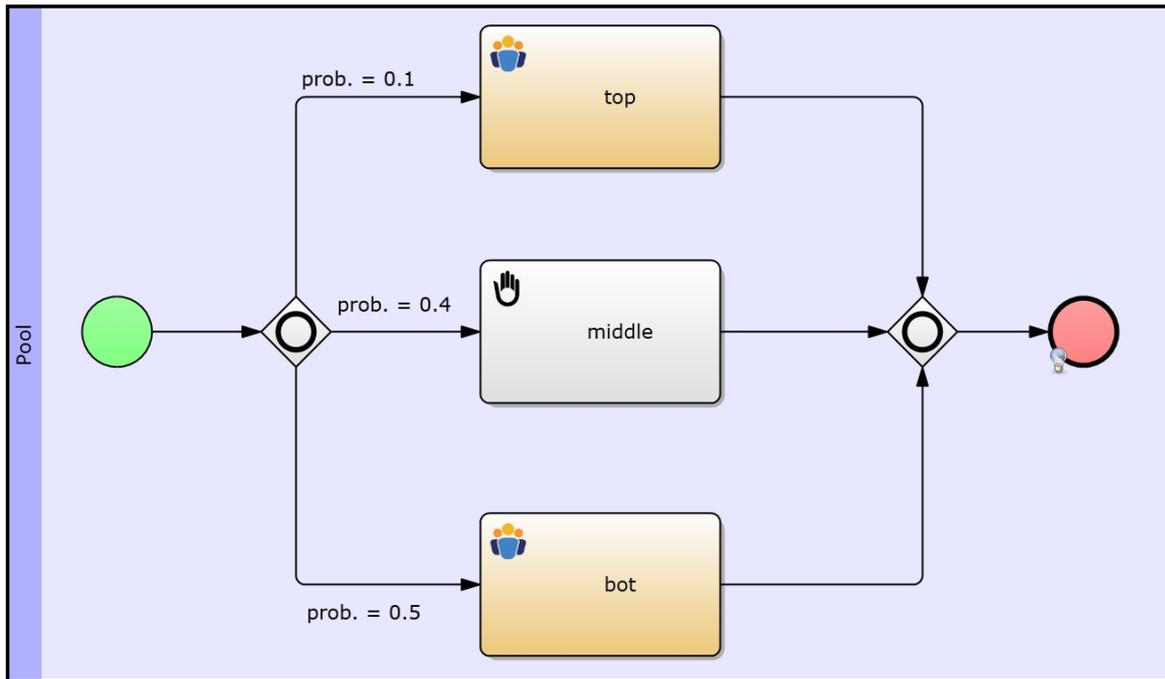


Figure 1: Model 3f - Inclusive Gateway with Probabilities / [YOPRO-Link](#)

### Approach

- Just like in [Model 3f](#) we will parametrize the sequence flows' Probability properties with the same values:
  - Top sequence flow: Probability = 0.,1
  - Middle sequence flow: Probability = 0.,4
  - bottom sequence flow: Probability = 0.,5
- At inclusive gateways the sum of the sequence flows' probabilities does not matter, because the decision of the sequence flow to be pursued is not exclusive
  - The sum can be smaller or bigger than 1

### Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

### Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

### Report statistics

- Have a look at this report's gateway details (see Figure 1)
- This report's gateway statistics resemble those from [Model 3f's report](#)
- The only considerable difference is that the number of each sequence flow (n) has increased
  - This bases on the fact, that the inclusive gateway allows multiple successive sequence flows
  - Note that the relation between the sequence flows is almost equal to those of [Model 3f](#)

### Chosen paths at conditional gateways

Title	Obs	Least	Most
Chosen paths at inclusive gateway: "#3"	333	prob. = 0.1#1	prob. = 0.5#1

Cell	Object	n	%	Cum. %
	prob. = 0.1#1	37	11.11111	11.11111
	prob. = 0.5#1	167	50.15015	61.26126
	prob. = 0.4#1	129	38.73874	100

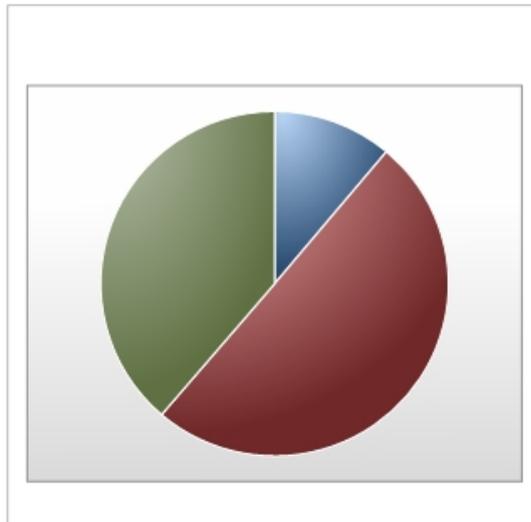


Figure 1: The report's gateway details

### 3. Messages & Signals

This section deals with the configuration of models whose pools use messages or signals to interact with each other.

#### Content

- [Model 4a - Message Exchange & Waiting Queues](#): Describes how message flows affect the simulation
- [Model 4b - Transporting Process Variables via Messages](#): Learn how to transport data via message flows
- [Model 4c - Messages without an explicit Edge](#): Describes how message events can be connected to each other without using explicit message flows
- [Model 5 - Sending Signals to Multiple Process Instances](#): Introduces the usage of signal events
- [Model 6 - Triggering Start Events](#): Explains how to trigger non-general and non-timer start events

Model 4a - Message Exchange & Waiting Queues

#### Goals of this lesson

- Understand how communication between pools affect the simulation

#### How-to

Model 4a simulates a simple communication between two processes.

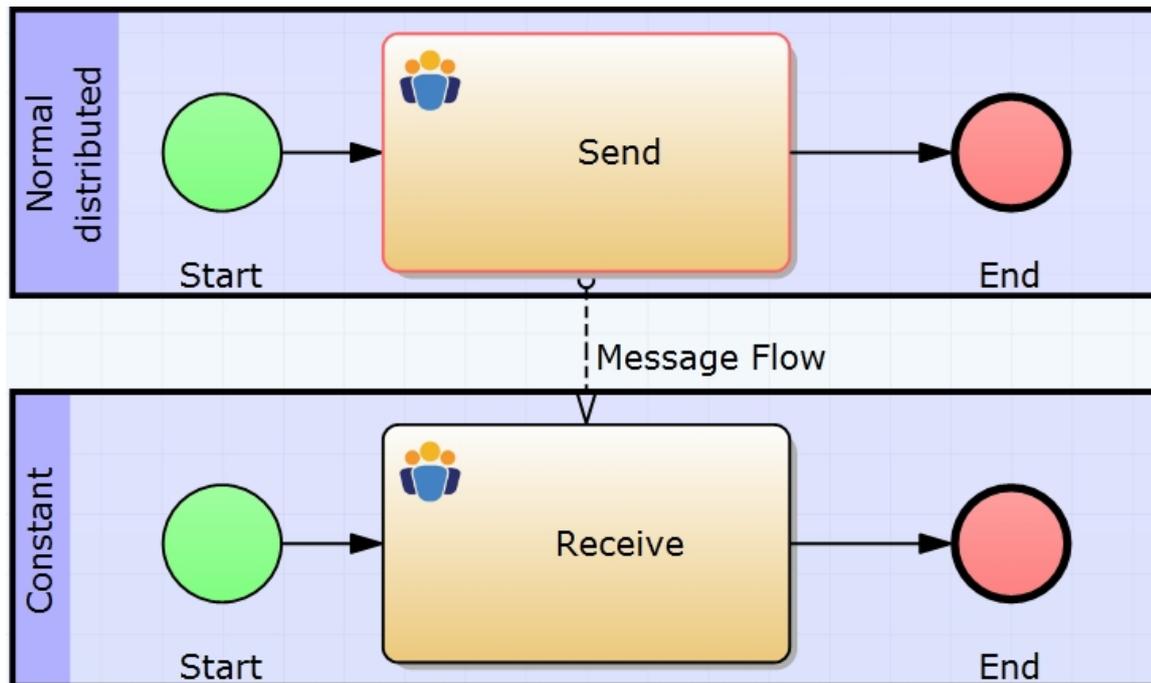


Figure 1: Model 4a - Message Exchange / [YOPRO-Link](#)

### Parametrization

- The Inter-arrival Time of the upper process is represented by a Normal Distribution with
  - a mean value of 1 TU
  - and a standard deviation of 0.75.
- The lower process has a constant Inter-arrival Time of 1 TU

### Explanation

So, sometimes the upper process will be started earlier and sometimes later than the constant (lower) process and the message will be sent to either an already started and waiting or a not yet existing process instance. This leads us to two different possibilities:

1. The lower process has to wait for the message to be sent.
2. The message waits for the lower process instance to be created and the upper process continues instantly.

Note: In addition you can add a Transport Delay for each message flow under Optional Properties, which is also represented by a stochastic distribution.

### Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

### Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

### Report statistics

- This fairly simple model introduces you to the concept of message exchange between pools
- Actually, both processes in the model should be completed in 1 hour
  - Remember that we set up constant distribution with a value of 1 for both tasks' duration

- This is true for the top pool, which sends the message
- But the bottom pool comes with a very big processing time
- How is that even possible?

Histogram for Processing time of Pool: Constant#1 (started at event: Start#2)

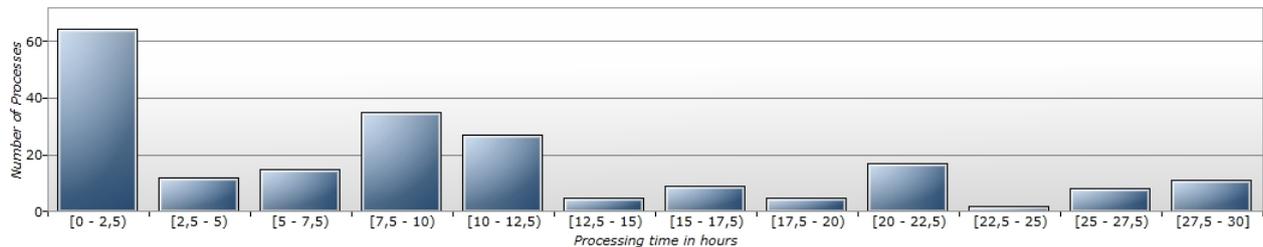


Figure 1: Histogram of the process cycle times

Process cycle times

Title	Mean	Std.Dv	Min	Max	Lower Quartile	Median	Upper Quartile
Processing time of Pool: Constant:	0d, 9h, 51m and 12s	0d, 8h, 25m and 3s	0d, 1h, 0m and 0s	1d, 5h, 36m and 50s	0d, 1h, 59m and 20s	0d, 8h, 34m and 36s	0d, 14h, 6m and 56s
Processing time of Pool: Normal d	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s

Figure 2: The process cycle times

### Explanation

- The trivial answer is: Because the top pool sends a message to the bottom pool.
- Though the answer is right, you might have some trouble accepting it
- And you are right!
- The bottom pool's processing time is not only dependent on the fact, that the upper pool sends a message to it
- It is also important that both pools' start events contain different distributions for their inter-arrival time
  - The top pool's start event has a Normal Distribution set with a mean of 1 and a standard deviation of 0.75
  - The bottom pool's start event has a Constant Distribution set with a value of 1
- This means that the bottom pool's process instance either has to wait for a message or that the message is already available
- Have a look at Figure 4, specifically at the inter-arrival time of the start event "Start Sender"
  - You will notice, that its mean value is 1 hour, 8 minutes and 9 seconds
  - It is slightly higher than the constant inter-arrival time of the bottom pool's start event
- So, the bottom pool's process instances have to wait 8 minutes and 9 seconds for a message receipt (on average)
- This waiting time accumulates to the already known value of the bottom pool's processing time.

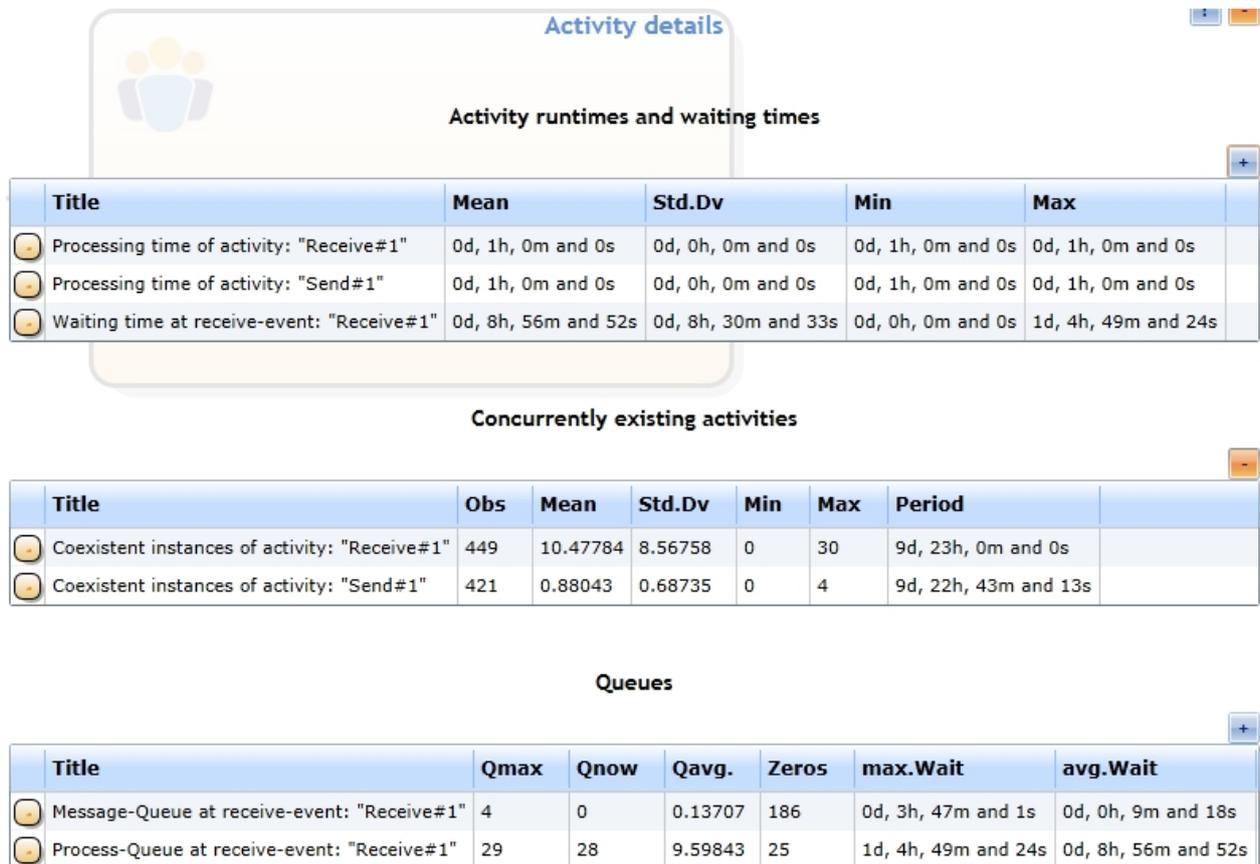


Figure 3: The activity details

## Queues

- Check out the activities' "Queues" statistics depicted in Figure 3
- You will discover that the message queue (messages waiting to get fetched by any process instance) is under-worked
  - There are no current messages in the queue (Qnow)
  - There are 186 Zeros
  - The maximum number of waiting elements is 4
- On the contrary the process queue (process instances waiting for a message) is quite busy
  - There are currently 28 process instances waiting for a message (Qnow)
    - And there are no messages to pick up (see above)
  - There are almost 10 process instances waiting for message at the same time on average (Qavg)
  - There are only 25 Zeros
- These queue statistics help you to prove or disprove your assumptions on the inefficiency of processes

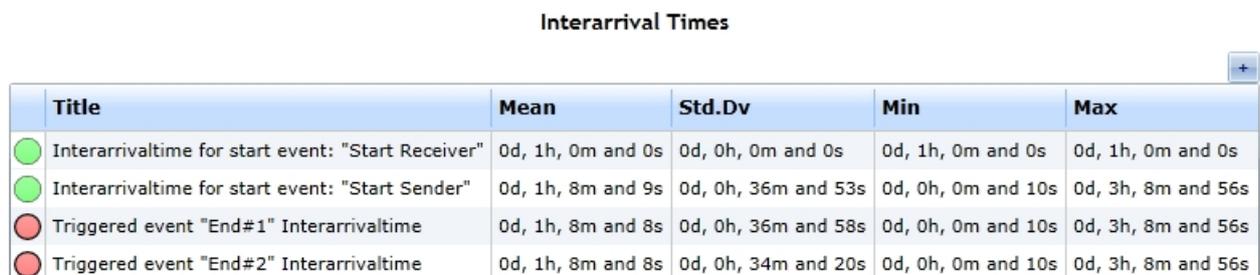


Figure 4: The inter-arrival times of events

## Model 4b - Transporting Process Variables via Messages

### Goals of this lesson

- Learn how to transport data (e. g. variables) between pools by using message flows

## How-to

The BPMN allows transportation of explicit data by messages. We will now transfer a variable called "var1" from the upper pool to the lower pool by using the message flow between the activities "Send Data" and "Receive Data".

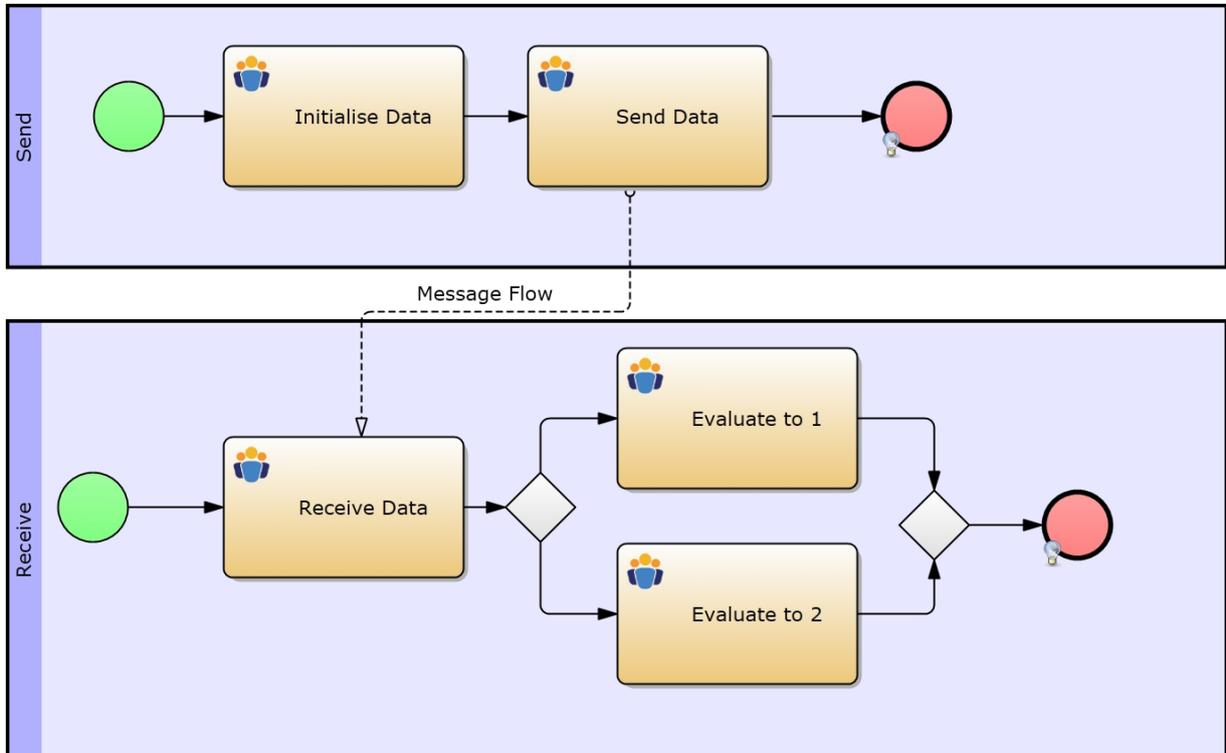


Figure 1: Model 4b - Transporting Process Variables via Messages / [YOPRO-Link](#)

## Approach

- Select the activity "Initialise Data" and define a variable called "var1" with the value 1 in the [expression property](#).
- Now click the message flow.
  - You will notice that it also contains an Expression property.
  - Click the Edit button
  - Enter the variable's name "var1" to make the message flow transport this specific variable
- Since the lower pool now also possesses the variable "var1", we will use it for the decision making at the branching exclusive gateway:
  - Upper edge condition:  $var1==1$
  - Lower edge condition:  $var1==2$

After simulating the model, you will notice that the activity "Evaluate to 2" has never been executed.

## Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240

- Simulation stop time unit: Hours

## Report statistics

- This report shows you that we really transported the variable var1 to the bottom pool
- The activity details from Figure 1 show that there are no statistics for activity "Evaluate to 2"
  - The table's line says insufficient data for activity "Evaluate to 2"
  - This happens, when the element (activity) is never reached by any process instance



Title	Mean	Std.Dv	Min	Max
Processing time of activity: "Evaluate to 1#1"	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s
Processing time of activity: "Evaluate to 2#1"	insufficient data	insufficient data	insufficient data	insufficient data
Processing time of activity: "Initialise Data#1"	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s
Processing time of activity: "Receive Data#1"	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s
Processing time of activity: "Send Data#1"	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s
Waiting time at receive-event: "Receive Data#1"	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s

Figure 1: The report's activity details

## Model 4c - Messages without an explicit Edge

### Goals of this lesson

- Learn how to use message events without drawing any message flows

### How-to

For a better overview of a more complex simulation model, you are able to hide the Message Flow. For an implicit message flow between two Message Events both have to be named equally. This is the case in Model 4c. (Have a look at [Linking Events to each other](#) for more information on this matter.)

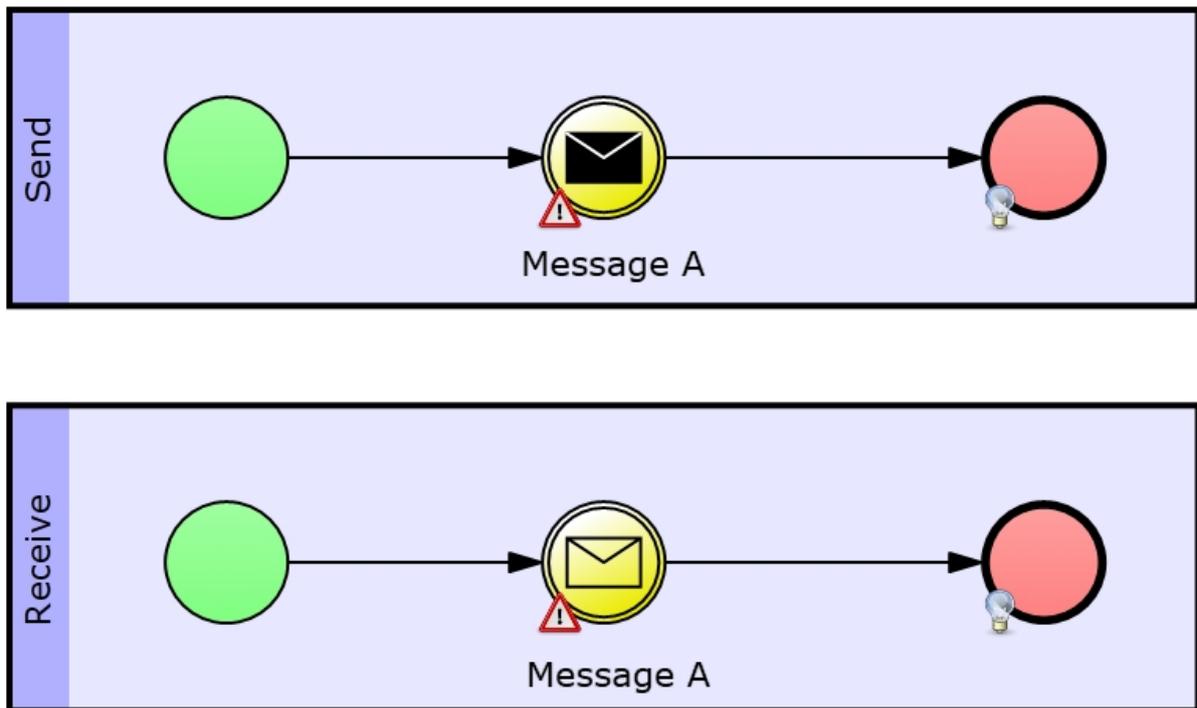


Figure 1: Model 4c - Messages without an explicit message flow / [YOPRO-Link](#)

Attention: This possibility is not yet given for activities, those always need an explicit message flow.

Note: Do not get confused with YOPRO's conformance warnings. It is right that receiving message events need an incoming message flow, but this does not matter for simulation or modeling purposes.

## Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

## Report statistics

- As you note, the simulation of this model generated a report, that is very similar to the one from [Model 4a](#)
  - Its statistics indicate the same reason for inefficiency as the one from [Model 4a](#)
  - This is not a real surprise, because the models are almost equal
  - The only difference is that we replaced the activities with intermediate message events

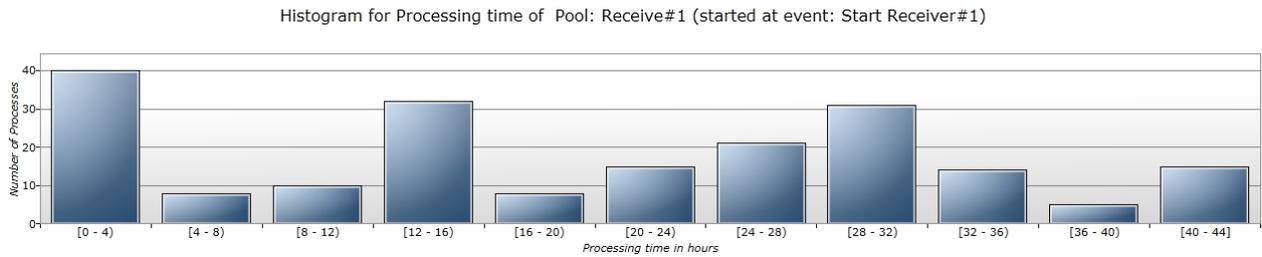


Figure 1: Histogram of the processing times

Process cycle times

Title	Mean	Std.Dv	Min	Max	Lower Quartile	Median	Upper Quartile
Processing time of Pool: Receive#1 (started at event: Start Receiver#1)	0d, 19h, 49m and 1s	0d, 12h, 29m and 7s	0d, 0h, 0m and 0s	1d, 18h, 7m and 8s	0d, 8h, 52m and 17s	0d, 21h, 35m and 56s	1d, 6h, 48m and 11s
Processing time of Pool: Send#1 (started at event: Start Sender#1)	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s

Figure 2: Processing times

## Model 5 - Sending Signals to Multiple Process Instances

### Goals of this lesson

- Learn how to use signal events

### How-to

As in the Message Events in Model 4c all linked Signal Events have to use the same name to identify them as linked. In contrast to messages, signals can be sent to multiple receivers.

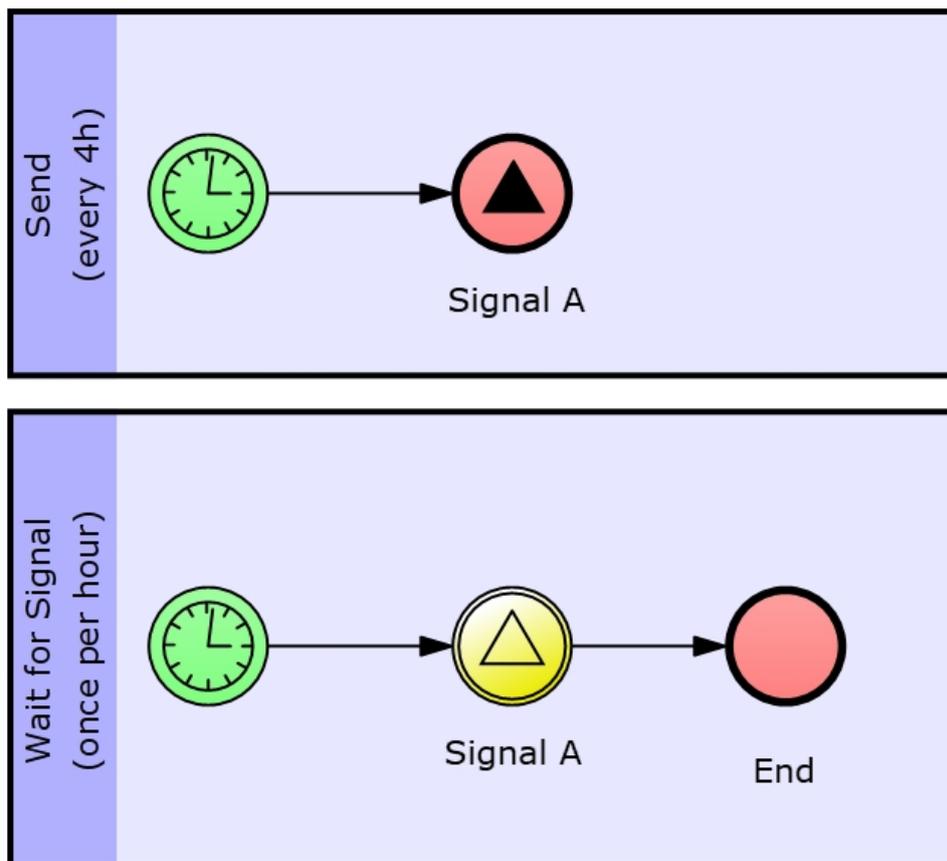


Figure 1: Model 5 - Sending Signals to Multiple Process Instances / [YOPRO-Link](#)

### Parametrization

- The model and its settings show that the upper process sends a signal every 4 TU.

- The lower process starts every 1 TU.

### Explanation

This means, that always 4 instances of the lower process wait for a signal from the upper process, and the incoming signal always re-activates these four process instances of the lower processing pool.

### Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

### Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

### Report statistics

- Retrospective:
  - The receiving process starts every hour
  - The sending process starts every 4 hours
  - Which means that there are always 4 process instances of the receiving process waiting for a signal from the top process
- This exact correlation can be found in the report
- The receiving pool's lead time is 1 hour and 30 minutes
  - The 1st process instance starts at time 1 and has to wait 3 hours for the signal
  - The 2nd process instance starts at time 2 and has to wait 2 hours for the signal
  - The 3rd process instance starts at time 3 and has to wait 1 hour for the signal
  - The 4th process instance starts at time 4 and receives the signal instantly
  - Summing up these values results in 6 hours, which must be divided by 4 (process instances) for the average of 1.5 or 1 hour and 30 minutes

Process cycle times

Title	Obs	Mean	Std.Dv	Min	Max	Lower Quartile	Median	Upper Quartile
Processing time of Pool: Send (every 4h)#1 (started at event: #1)	59	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s
Processing time of Pool: Wait for Signal (once per hour)#1 (started at event: #2)	236	0d, 1h, 30m and 0s	0d, 1h, 7m and 13s	0d, 0h, 0m and 0s	0d, 3h, 0m and 0s	0d, 0h, 30m and 0s	0d, 1h, 30m and 0s	0d, 2h, 30m and 0s

Figure 1: The processing times

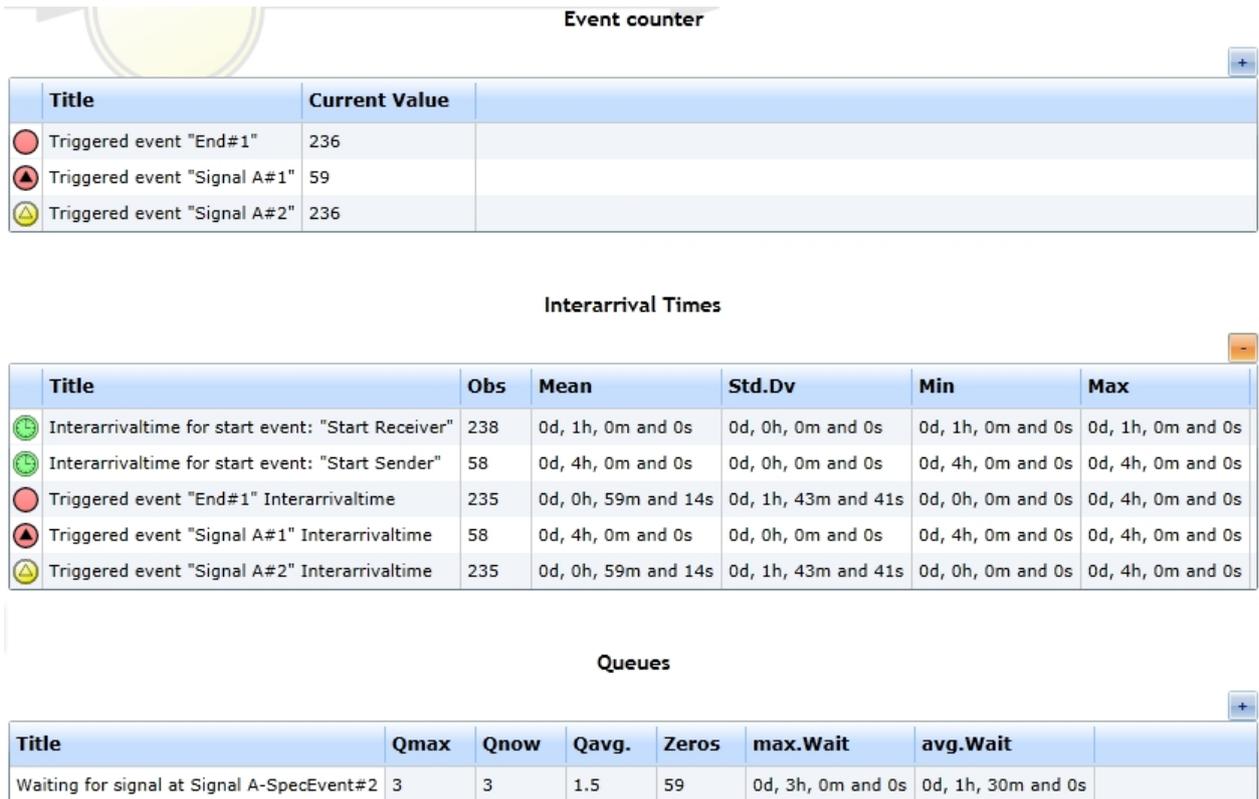


Figure 2: The Report's event details

## Model 6 - Triggering Start Events

### Goals of this lesson

- Learn how to trigger start events via messages

### How-to

Model 6 displays a process with a Timer Start Event and a process with a Message Start Event. We will use the upper pool's message sending end event to trigger the message receiving start event of the lower pool.

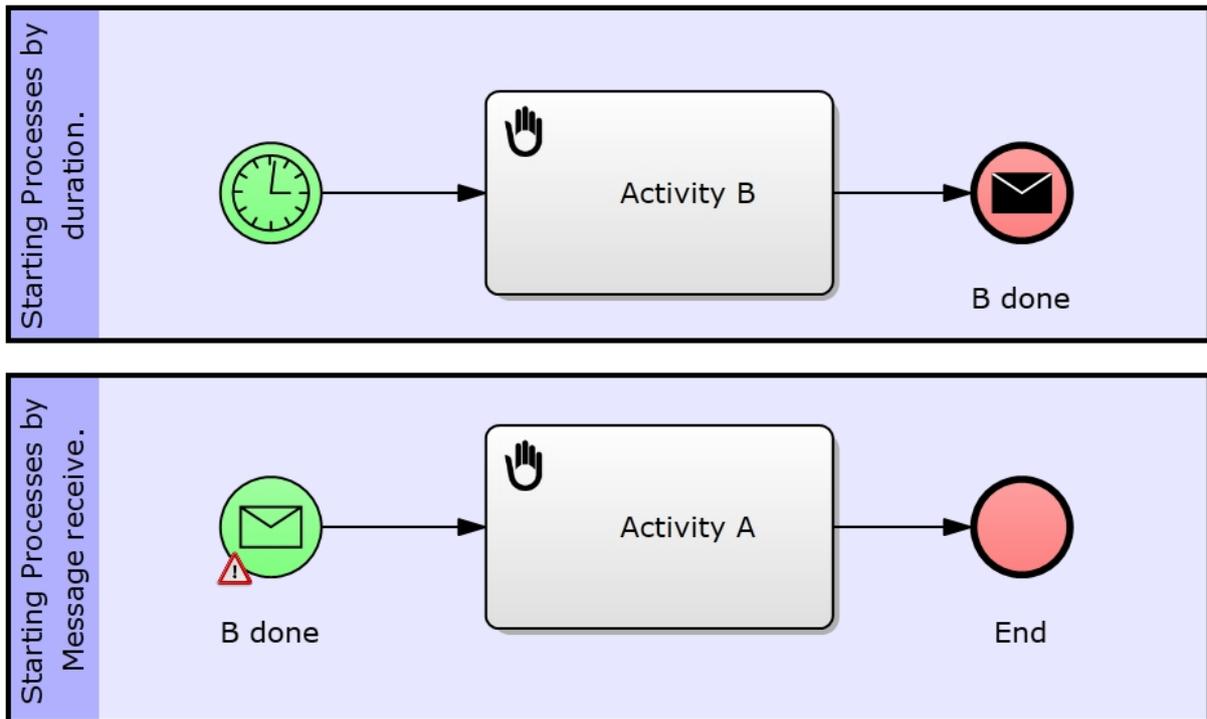


Figure 1: Model 6 - Triggering Start Events / [IYOPRO-Link](#)

### Approach

- The upper process starts after the given time span provided by the stochastic distribution.
- The lower process starts with the incoming message that is sent by the upper process' End Event.
  - As you already learned in lesson [Messages without an explicit edge](#) or in [Linking Events to each other](#) sending and (their respective) receiving events can be linked by giving them the same name.
  - Of course you could also use a message flow to link the upper pool's message sending event with the lower pool's message receiving event.
- Note: The behavior for signals is equivalent, except that they can trigger more than one process instance.

### Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

### Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

### Report statistics

- This model's sender process activates the receiving process by a message sending end event
- The report's statistics should be self-explaining

### Process cycle times

Title	Mean	Std.Dv	Min	Max	Lower Quartile	Median	Upper Quartile
Processing time of Pool: Starting Processes by duration	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s				
Processing time of Pool: Starting Processes by Message	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s				

Figure 1: The processing times

## 4. Extended Sequence Concepts

In this section we will deal with the concept of multiple activities, which can be looping, sequential or parallel. Also we give a short example for the Link Event.

### Content

- [Model 7a - Multiple Sequential Activity](#): Describes how to parametrize a multiple sequential activity
- [Model 7b - Multiple Looping Activity](#): Describes how to use looping activities for simulation purposes
- [Model 7c - Multiple Parallel Activity](#): Explains the usage of multiple parallel activities
- [Model 8 - Link Event instead of Sequence Flows](#): Introduces the usage of the link event

### Model 7a - Multiple Sequential Activity

#### Goals of this lesson

- Learn how to set up the prerequisites for multiple sequential activities

#### How-to

Model 7a represents a process which mainly consists of two activities. The first one declares a variable  $x$ , which we then use to define the number of iterations for the second activity that will be executed sequentially. This means that the number assigned to  $x$  determines how often the sequential activity will be executed.

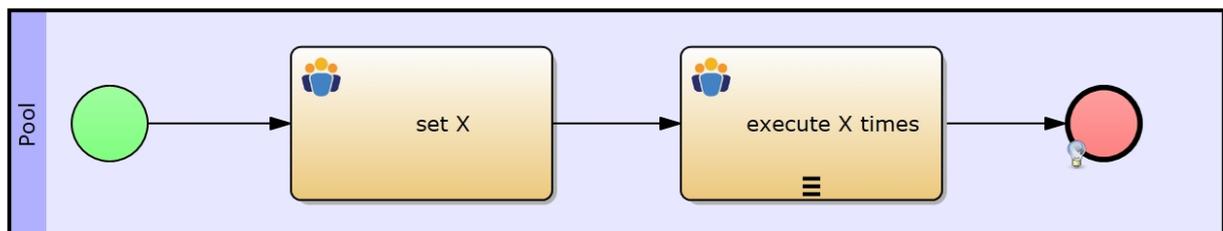


Figure 1: Model 7a - Multiple Sequential Activity / [IYOPRO-Link](#)

### Approach

- First we will define a variable called  $x$  which will serve our purposes to determine the number of executions of the sequential activity
  - Click the first activity and define the variable  $x$  with a value of 5
- Now we set the second activity to be Sequential
  - Click the second activity
  - Select Sequential from the drop-down-box next to the Multiple Instances property
- To go on make sure you selected Simulation from the Property Set drop-down-box
- Now we need to specify the Number of Iterations of our sequential activity
  - Click the Edit button next to the Number of Iterations property under the Must-Have Simulation Parameters
  - Enter the variable's name that we defined in the first activity ( $x$ )

### Explanation

We defined a variable  $x$  with a value of 5. The Variable  $x$  is used to determine the number of the second activity's iterations. This means the second (sequential) activity will be executed 5 times, before the process proceeds to the end event.

Note:

- The Number of Iterations property can contain either a variable (which holds an integer value) or an expression (that evaluates to an integer value).
- You can assign the value 5 directly or use expressions like  $x + 2$ , which would be evaluated to 7.

## Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

## Report statistics

- As we already explained in the [model description](#), the sequential activity will be executed 5 times (depending on our model configuration)
- You can recognize this correlation in the process cycle times
  - Each of the two activities last 1 hours (constantly)
  - The process' lead time is 6 hours.
- Also check out the activity details in figure 2
  - The run-times of the sequential activity is almost exactly 5 times as big as the run-times of the 1st activity
  - Also the concurrently existing activities statistics show that there have been almost 5 sequential activities executed at the same time
    - Note that this assumption is only valid, because the bottom process instances start each hour and because the sequential activity lasts one hour

Process cycle times

Title	Mean	Std.Dv	Min	Max	Lower Quartile	Median	Upper Quartile
Processing time	0d, 6h, 0m and 0s	0d, 0h, 0m and 0s	0d, 6h, 0m and 0s				

Figure 1: The processing times

**Activity runtimes and waiting times**

Title	Obs	Mean	Std.Dv	Min	Max
Processing time of activity: "execute X times#1"	1175	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s
Processing time of activity: "set X#1"	238	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s

**Concurrently existing activities**

Title	Mean	Std.Dv	Min	Max
Coexistent instances of activity: "execute X times#1"	4.95798	0.35254	1	5
Coexistent instances of activity: "set X#1"	1	0	0	1

Figure 2: The report's activity details

## Model 7b - Multiple Looping Activity

## Goals of this lesson

- Learn how to set up the prerequisites for a looping activity

## How-to

In this lesson we will introduce the looping activity. This kind of activity is being executed until a specified Looping Condition is broken (evaluated to true).

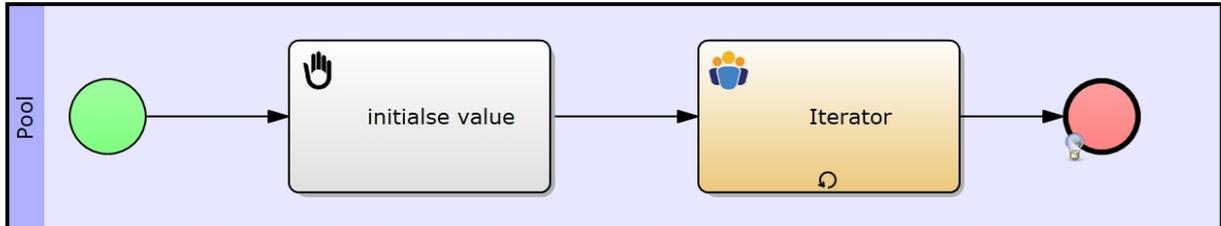


Figure 1: Model 7b - Multiple Looping Activity / [IYOPRO-Link](#)

## Approach

- First we define a variable  $x$  with a value of 0 in the first activity
- Now we need to increment  $x$  by 1 in the second activity
  - Click the activity and enter  $x=x+1$ ; in its Expression property
- Finally it is necessary to set up the Looping Condition
  - Click the Edit button next to the Looping Condition property under Must-Have simulation Parameters (of the second activity)
  - Enter the condition  $x>4$
  - Evaluating the looping condition is always the last step in the execution of a looping activity

## Explanation

The first activity initializes the variable  $x$  with the value of 0. When the process flow arrives at the second activity,  $x$  will be increased by 1 and after the activity has been worked off the condition will be checked. The activity will be executed as long as the value does not meet the condition. In result the second activity will be executed 5 times just like the multiple activity from Model 7a.

## Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

## Report statistics

- Just like in [Model 7a](#) the looping activity has been executed 5 times before the process instances proceeded to the end event
- Note that the processing time is 7 hours, because we increased the "initialise value" activity's duration to 2 hours

Process cycle times

Title	Mean	Std.Dv	Min	Max	Lower Quartile	Median	Upper Quartile
Processing time	0d, 7h, 0m and 0s	0d, 0h, 0m and 0s	0d, 7h, 0m and 0s				

Figure 1: The processing times

Activity runtimes and waiting times					
	Title	Mean	Std.Dv	Min	Max
<input type="checkbox"/>	Processing time of activity: "initialse value#1"	0d, 2h, 0m and 0s	0d, 0h, 0m and 0s	0d, 2h, 0m and 0s	0d, 2h, 0m and 0s
<input checked="" type="checkbox"/>	Processing time of activity: "Iterator#1"	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s

Concurrently existing activities

	Title	Mean	Std.Dv	Min	Max
<input type="checkbox"/>	Coexistent instances of activity: "initialse value#1"	1.99582	0.06455	1	2
<input checked="" type="checkbox"/>	Coexistent instances of activity: "Iterator#1"	4.95781	0.35327	1	6

Figure 2: The report's activity details

## Model 7c - Multiple Parallel Activity

### Goals of this lesson

- Learn how to set up the prerequisites for a parallel activity

### How-to

In Model 7c we will introduce you to the multiple parallel activity. This kind of activity is being executed x times simultaneously. The number of parallel instances is determined by an expression which must evaluate to an integer.

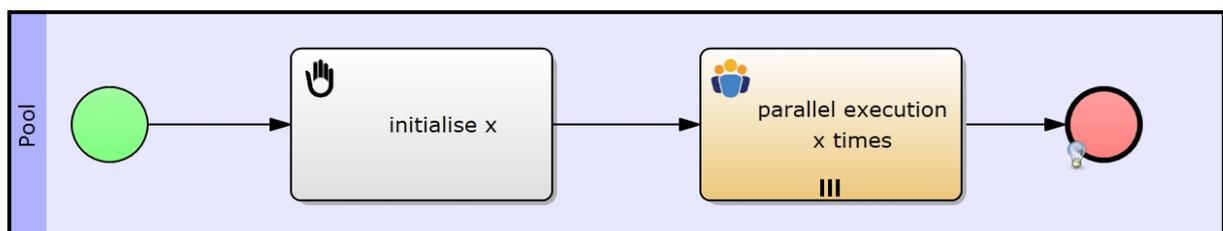


Figure 1: Model 7c - Multiple Parallel Activity / [YOPRO-Link](#)

### Approach

- We again initialize a variable called x with a value of 4 in the first activity
- At the second activity we select the item Parallel from the drop-down-box Multiple Instances
  - We now enter the above defined variable's name (x) into the expression field of the Number of Parallel Instances property under Must-Have Simulation Parameters
  - Note: You can also enter an expression like  $\text{Result}=(x+4)$ ; (which will result in 8 parallel instances) or just enter a number

### Explanation

The first activity defines the variable x with the value 4. We then use x to determine the number of parallel instances of the second (parallel) activity. This means that the second activity will be executed 4 times simultaneously.

### Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

## Report statistics

- Though the parallel activity is being executed 4 times in each process instance, you will note that this multiplicity does not impact on the processing time
  - This is what we expected, because the activities are executed parallel
- Try to change the parallel activity's duration to a non-constant stochastic distribution like the normal distribution
  - You will notice that the statistics will not be that obvious anymore
  - The parallel activity instance with the longest duration will make the hardest impact on the processing time

Process cycle times

Title	Mean	Std.Dv	Min	Max	Lower Quartile	Median	Upper Quartile
Processing time of Pool	0d, 2h, 0m and 0s	0d, 0h, 0m and 0s	0d, 2h, 0m and 0s				

Figure 1: The processing times

Activity runtimes and waiting times

Title	Mean	Std.Dv	Min	Max
<input type="checkbox"/> Processing time of activity: "initialise x#1"	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s
<input checked="" type="checkbox"/> Processing time of activity: "parallel execution x times#1"	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s

Concurrently existing activities

Title	Mean	Std.Dv	Min	Max
<input type="checkbox"/> Coexistent instances of activity: "initialise x#1"	1	0	0	1
<input checked="" type="checkbox"/> Coexistent instances of activity: "parallel execution x times#1"	119.5	68.70408	1	238

Figure 2: The report's activity details

## Model 8 - Link Event instead of Sequence Flows

### Goals of this session

- Usage of the Link Event

### How-to

Sometimes more complex models require a split in the flow to gain a certain level of cleanness. Modeling from left to right might result in a long model. By using the Link Event you can "push" the rest of the model to the next line. This saves the long edge that would be displayed across the whole model.

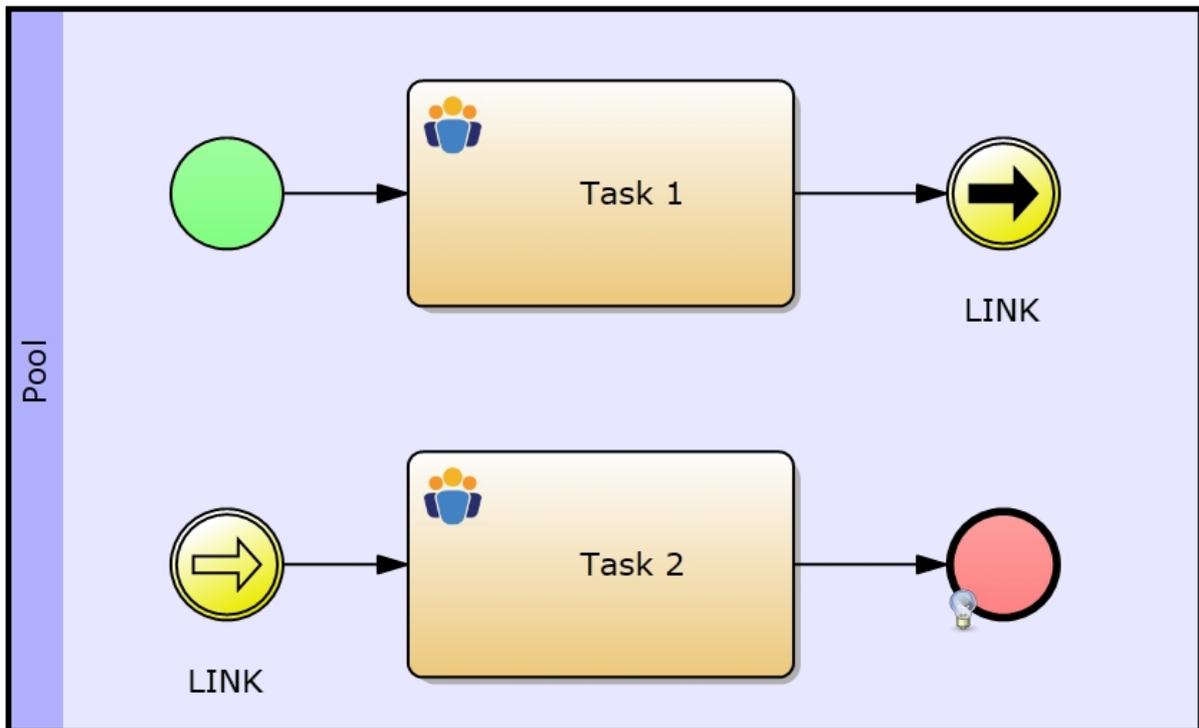


Figure 1: Model 8 - Link Events instead of Sequence Links / [YOPRO-Link](#)

### Approach

- Create a sending Link Event where you want the sequence flow to be stopped
- Create a receiving Link Event where you want the sequence flow to be continued
- You must give both events the same name, so that they are linked to each other
  - (see [Linking Events to each other](#) for more information)

### Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

### Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

### Report statistics

- This report shows that link events do not (and should not) have any impact on the process' lead time
- Please replace both link events with a simple edge between the to activities and simulate the model with the same run configuration
- The new report will be the same as this one
- The only difference will that there will not be any statistics on the link events

Histogram for Processing time of Pool: Pool#1 (started at event: #2)

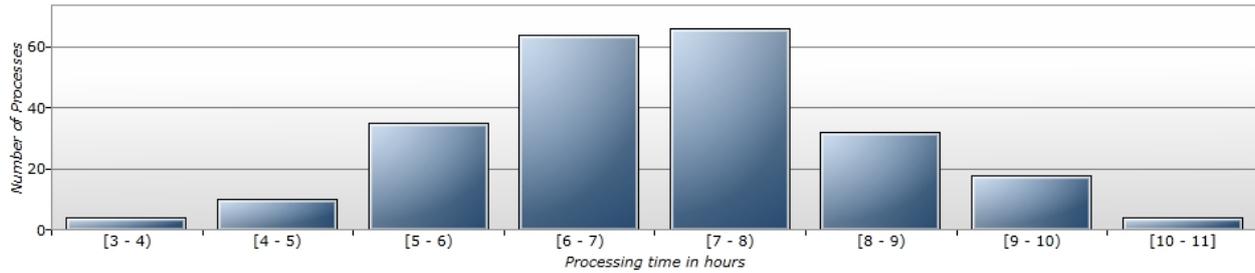


Figure 1: Histogram of the processing time

Process cycle times

Title	Mean	Std.Dv	Min	Max	Lower Quartile	Median	Upper Quartile
Processing time	0d, 7h, 2m and 30s	0d, 1h, 19m and 47s	0d, 3h, 9m and 53s	0d, 10h, 40m and 52s	0d, 6h, 11m and 48s	0d, 7h, 2m and 40s	0d, 7h, 52m and 19s

Figure 2: The processing time

Event counter

Title	Current Value
Triggered event "#1"	233
Triggered event "LINK#1"	236

Interarrival Times

Title	Mean	Std.Dv	Min	Max
Interarrivaltime for start event: ""	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s
Triggered event "#1" Interarrivaltime	0d, 1h, 0m and 5s	0d, 0h, 50m and 56s	0d, 0h, 0m and 31s	0d, 3h, 54m and 55s
Triggered event "LINK#1" Interarrivaltime	0d, 1h, 0m and 7s	0d, 0h, 45m and 39s	0d, 0h, 0m and 42s	0d, 4h, 4m and 13s

Figure 3: The report's event details

## 5. Sub-Processes

This section covers BPMN's Sub-Process Activities and Attached Events.

### Content

- [Model 9 - Activity Subprocesses](#): Depicts the usage of activity subprocesses in simulation.
- [Model 10 - Attached Error Events](#): Explains the functionality of attached events.
- [Model 11 - Events attached to Subprocesses](#): Describes how the subprocesses' end events can trigger events, which are attached to the subprocess activity

### Model 9 - Activity Sub-Processes

#### Goals of this lesson

- Usage of subprocess activities
- Considering the differences between configuring models and subprocess models for simulation purposes

#### How-to

Model 9 mainly consists of two activities, of which the first one is a Subprocess Activity. Subprocess activities are special because they contain another simple BPMN model, which will be executed in the activity's life-cycle.

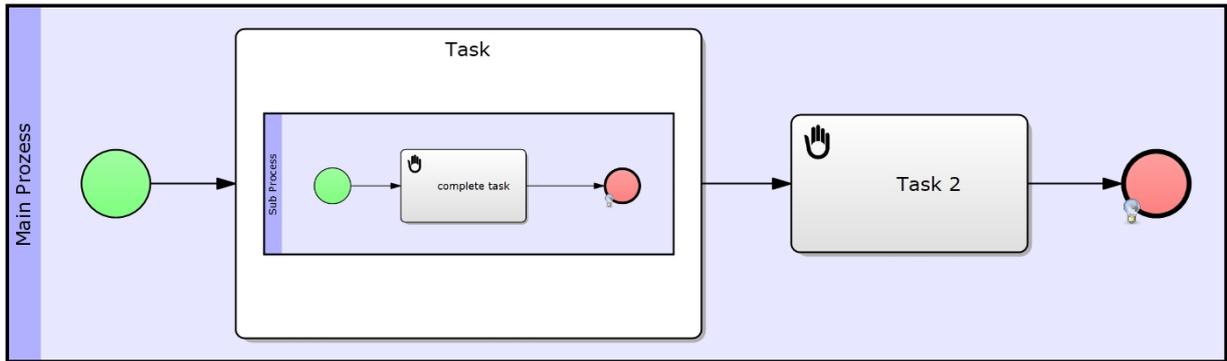


Figure 1: Model 9 - Activity Subprocess / [IYOPRO-Link](#)

### Approach

- Click the first activity (which should be a subprocess activity)
  - If the activity is not a subprocess activity, then select Subprocess from the Element Type drop-down-box under the General Properties
- Now click the Set button next to the Subprocess property under Subprocess Properties
  - Note: This option is only available in the Modeling Property Set
  - IYOPRO's File Browser shows up and you can select the model, which functions as the subprocess
- There is still one thing to understand
  - Subprocess activities do not contain any duration properties
- You have to make sure that your Subprocess model is simulation-ready configured
  - Important: The start events of Subprocess models must not have an inter-arrival time assigned
  - But all the other (allowed) elements like gateways and activities still need to be configured
  - You may also use variables, which are known to the subprocess activity in which the model is being executed
    - e. g. for decision making at gateways

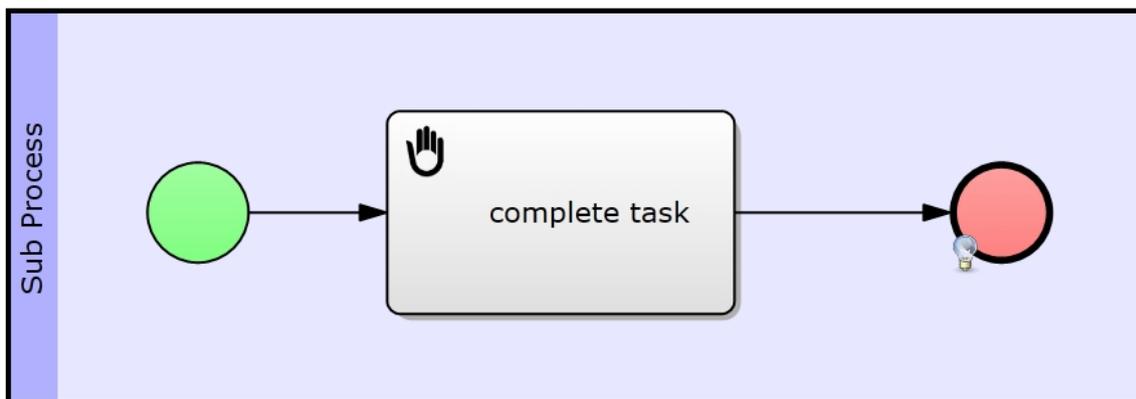


Figure 2: Model 9a -The subprocess for Model 9 / [IYOPRO-Link](#)

### Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

### Run configuration

- Stochastic Seed: 12345

- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

## Report statistics

- The report's process cycle times (figure 1) contain two processing times
  - The one for the main process
  - The one for the subprocess (which is being executed in the first task)
- Note that the statistics from "Task 1" (Figure 2 - activity run-times) and those from the subprocess (lead time) are equal
  - Since the subprocess is executed within the activity, this must be the case

Process cycle times

Title	Mean	Std.Dv	Min	Max	Lower Quartile	Median	Upper Quartile
Processing time of Pool: Main Prozess#1	0d, 4h, 9m and 24s	0d, 0h, 52m and 52s	0d, 2h, 11m and 36s	0d, 6h, 44m and 38s	0d, 3h, 33m and 59s	0d, 4h, 7m and 5s	0d, 4h, 46m and 32s
Processing time of Pool: Sub Prozess#1	0d, 2h, 0m and 0s	0d, 0h, 0m and 0s	0d, 2h, 0m and 0s	0d, 2h, 0m and 0s			

Figure 1: The processing times

Activity runtimes and waiting times

Title	Mean	Std.Dv	Min	Max
<input type="checkbox"/> Processing time of activity: "complete task#1"	0d, 2h, 0m and 0s	0d, 0h, 0m and 0s	0d, 2h, 0m and 0s	0d, 2h, 0m and 0s
<input type="checkbox"/> Processing time of activity: "Task 2#1"	0d, 2h, 9m and 24s	0d, 0h, 52m and 52s	0d, 0h, 11m and 36s	0d, 4h, 44m and 38s
<input checked="" type="checkbox"/> Processing time of activity: "Task#1"	0d, 2h, 0m and 0s	0d, 0h, 0m and 0s	0d, 2h, 0m and 0s	0d, 2h, 0m and 0s

Concurrently existing activities

Title	Mean	Std.Dv	Min	Max
<input type="checkbox"/> Coexistent instances of activity: "complete task#1"	1.99582	0.06455	1	2
<input type="checkbox"/> Coexistent instances of activity: "Task 2#1"	2.15604	0.77184	0	4
<input checked="" type="checkbox"/> Coexistent instances of activity: "Task#1"	1.99582	0.06455	1	2

Figure 2: The report's activity details

## Model 10 - Attached Error Event

### Goals of this lesson

- Understand the usage of Error Events

### How-to

In this model we introduce the error event. "Task 1" defines a variable, on which the decision making at the branching exclusive gateway is based on. If the variable's value leads to the error end event, it will fire the receiving error event attached to "Task 3". This event is interrupting and causes the process to proceed to the task "Task Handle Error". If the branching exclusive gateway's default sequence flow is being followed, "Task 3" will be executed normally.

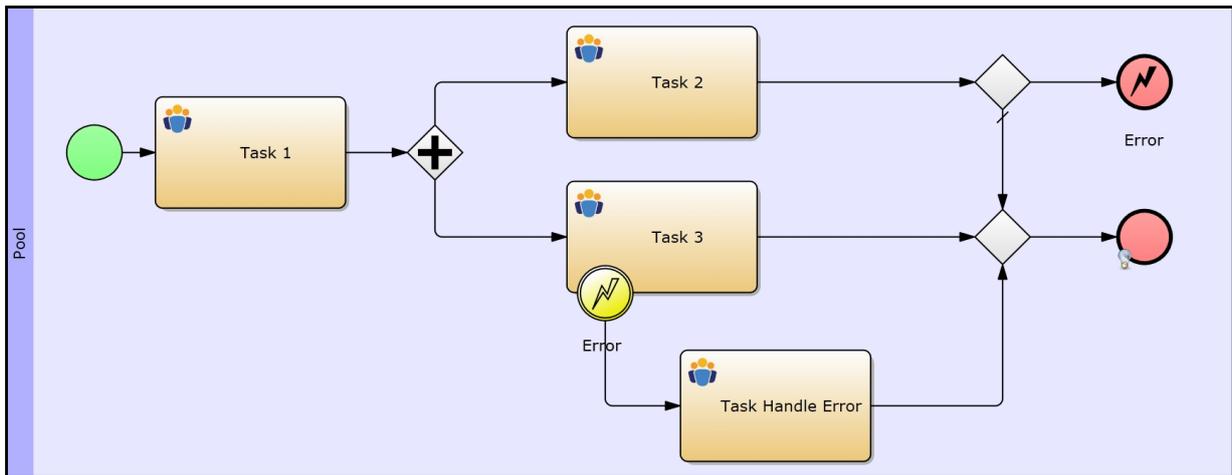


Figure 1: Model 10 - Attached Error Event / [YOPRO-Link](#)

### Approach

- Define a stochastic variable called  $x$  with a Discrete Uniform Distribution with
  - A minimum value of 2
  - A maximum value of 5
- At the branching exclusive gateway set these conditions for the edges:
  - Upper edge:  $x \leq 3$
  - Lower edge: default

### Hint

The other event types can be used in a similar way.

Caution: Some event types have interrupting and non-interrupting properties, others may be used only in an interrupting way (e.g. the Error Event). In case of an inaccurate usage the simulation will throw an error in the simulation report.

### Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

### Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

### Report statistics

Process cycle times

Title	Mean	Std.Dv	Min	Max	Lower Quartile	Median	Upper Quartile
Processing time	0d, 3h, 28m and 58s	0d, 0h, 30m and 2s	0d, 3h, 0m and 0s	0d, 4h, 0m and 0s	0d, 3h, 0m and 0s	0d, 3h, 0m and 0s	0d, 4h, 0m and 0s

Figure 1: The processing times

- First have a look at the gateway details from figure 4
  - Our gateway fired the top sequence flow leading to the error sending end event 115 times
  - Which means that "Task 3" has been interrupted 115 times (take a look at the event counter

- from figure 3)
- So, the task "Task Handle Error" has also been reached 115 times (it was only completed 114 times, figure 2)
- The fact, that the "Task Handle Error" is being executed, has an impact on the process cycle time (see figure 1)

**Activity runtimes and waiting times**

Title	Obs	Mean	Std.Dv	Min	Max
Processing time of activity: "Task 1#1"	238	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s
Processing time of activity: "Task 2#1"	237	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s
Processing time of activity: "Task 3#1"	237	0d, 1h, 30m and 53s	0d, 0h, 30m and 3s	0d, 1h, 0m and 0s	0d, 2h, 0m and 0s
Processing time of activity: "Task Handle Error#1"	114	0d, 2h, 0m and 0s	0d, 0h, 0m and 0s	0d, 2h, 0m and 0s	0d, 2h, 0m and 0s

**Concurrently existing activities**

Title	Obs	Mean	Std.Dv	Min	Max	Period
Coexistent instances of activity: "Task 1#1"	477	1	0	0	1	9d, 23h, 0m and 0s
Coexistent instances of activity: "Task 2#1"	475	1	0	0	1	9d, 22h, 0m and 0s
Coexistent instances of activity: "Task 3#1"	475	1.51261	0.49984	1	2	9d, 22h, 0m and 0s
Coexistent instances of activity: "Task Handle Error#1"	229	0.96624	0.68661	0	2	9d, 21h, 0m and 0s

Figure 2: The report's activity details

**Event counter**

Title	Current Value
Triggered event "#3"	358
Triggered event "Error#1"	115
Triggered event "Error#2"	115

**Interarrival Times**

Title	Obs	Mean	Std.Dv	Min	Max
Interarrivaltime for start event: ""	238	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s
Triggered event "#3" Interarrivaltime	357	0d, 0h, 39m and 29s	0d, 0h, 28m and 29s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s
Triggered event "Error#1" Interarrivaltime	114	0d, 2h, 4m and 12s	0d, 1h, 28m and 25s	0d, 1h, 0m and 0s	0d, 8h, 0m and 0s
Triggered event "Error#2" Interarrivaltime	114	0d, 2h, 4m and 12s	0d, 1h, 28m and 25s	0d, 1h, 0m and 0s	0d, 8h, 0m and 0s

Figure 3: The report's event details

### Chosen paths at conditional gateways

	Title	Obs	Least	Most
◇	Chosen paths at exclusive gateway: "#5"	237	#15	#16

Cell	Object	n	%	Cum. %
	#15	115	48.52321	48.52321
	#16	122	51.47679	100

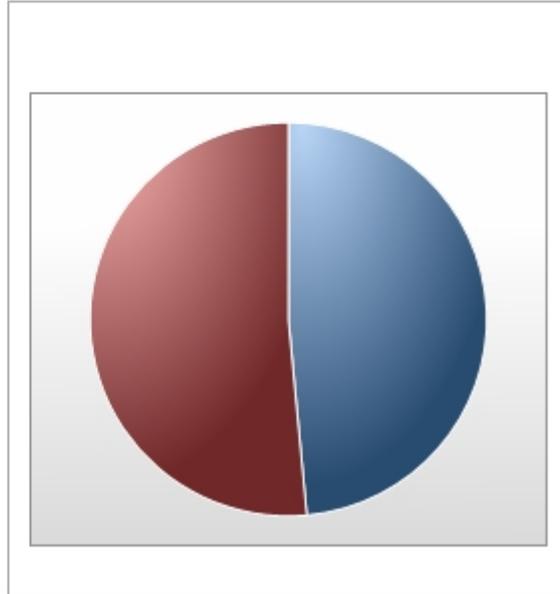


Figure 4: The report's gateway details

## Model 11 - Events attached to Sub-Processes

### Goals of this lesson

- Catching events from sub-processes

### How-to

In this model we will catch Cancel Events that are fired in the subprocess model of a subprocess activity. Have a look at Model 11 and Model 11a (the subprocess model).

Therefore we will define a stochastic variable which then will be used for the decision making at the branching exclusive gateway in Model 11. Depending on this variable the subprocess will either finish normally. Or it will throw one of the cancel events, which will fire the respective cancel event attached to the subprocess activity. Each of the two attached cancel events will lead to a respective activity that will be executed to treat the canceling of the subprocess.

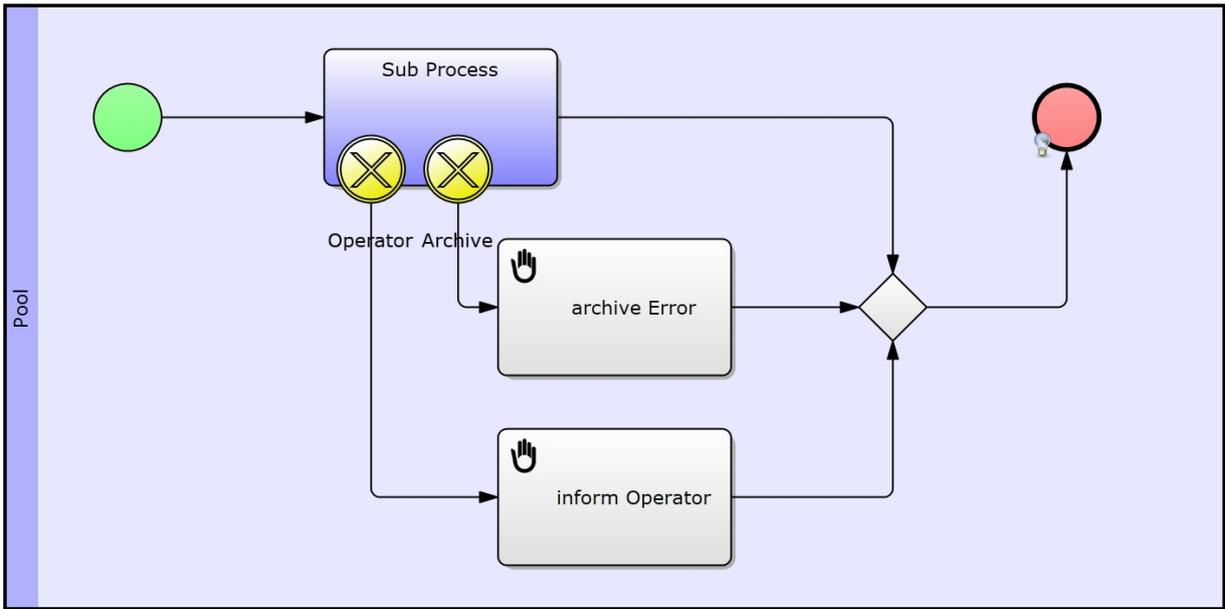


Figure 1: Model 11 - Events attached to Subprocesses / [YOPRO-Link](#)

### Approach

- Define a stochastic variable called  $x$  in Model 11a (the subprocess model, see Figure 2 below)
  - Pick the Discrete Uniform Distribution with
  - A minimum value of 2
  - A maximum value of 5
- Also create these conditions at the branching exclusive gateway in Model 11a
  - Upper edge:  $x > 3$
  - Lower edge:  $x < 3$
  - Middle edge: default

### Explanation

This means that the subprocess model will only finish with the general end event, if  $x$  equals 2. otherwise it will trigger one of the receiving cancel events that are attached to the subprocess activity.

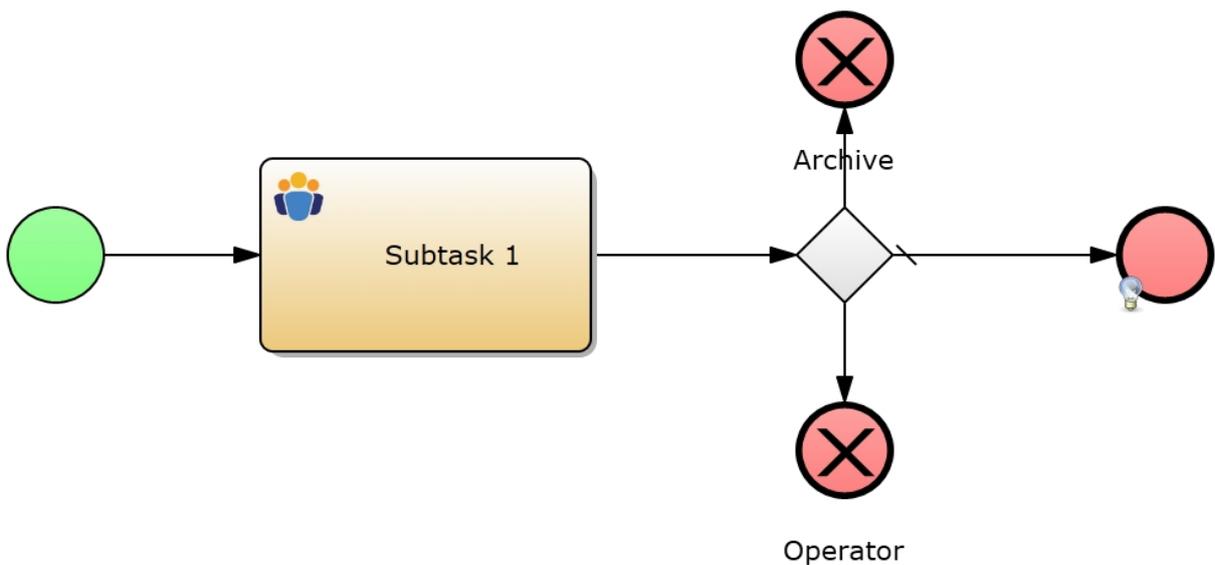


Figure 2: Model 11a - Subprocess model of Model 11 / [YOPRO-Link](#)

## Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

## Report statistics

- In figure 1 you find the process cycle times for the main process and the subprocess that is being executed in the subprocess activity "Sub Process"
- Note that the subprocess' cycle times are identical to the run-times of the subprocess activity (figure 2)

**Process cycle times**

Title	Mean	Std.Dv	Min	Max	Lower Quartile	Median	Upper Quartile
Processing time of DefaultPool	0d, 2h, 0m and 0s	0d, 0h, 0m and 0s	0d, 2h, 0m and 0s	0d, 2h, 0m and 0s	0d, 2h, 0m and 0s	0d, 2h, 0m and 0s	0d, 2h, 0m and 0s
Processing time of Pool: Pool#	0d, 4h, 4m and 5s	0d, 1h, 25m and 50s	0d, 2h, 0m and 0s	0d, 6h, 0m and 0s	0d, 4h, 0m and 0s	0d, 4h, 0m and 0s	0d, 6h, 0m and 0s

Figure 1: The processing times

**Activity runtimes and waiting times**

Title	Obs	Mean	Std.Dv	Min	Max
<input type="checkbox"/> Processing time of activity: "archive Error#1"	115	0d, 2h, 0m and 0s	0d, 0h, 0m and 0s	0d, 2h, 0m and 0s	0d, 2h, 0m and 0s
<input type="checkbox"/> Processing time of activity: "inform Operator#1"	64	0d, 4h, 0m and 0s	0d, 0h, 0m and 0s	0d, 4h, 0m and 0s	0d, 4h, 0m and 0s
<input checked="" type="checkbox"/> Processing time of activity: "Sub Process#1"	237	0d, 2h, 0m and 0s	0d, 0h, 0m and 0s	0d, 2h, 0m and 0s	0d, 2h, 0m and 0s
<input type="checkbox"/> Processing time of activity: "Subtask 1#1"	237	0d, 2h, 0m and 0s	0d, 0h, 0m and 0s	0d, 2h, 0m and 0s	0d, 2h, 0m and 0s

**Concurrently existing activities**

Title	Mean	Std.Dv	Min	Max
<input type="checkbox"/> Coexistent instances of activity: "archive Error#1"	0.99142	0.65507	0	2
<input type="checkbox"/> Coexistent instances of activity: "inform Operator#1"	1.08861	0.74386	0	3
<input checked="" type="checkbox"/> Coexistent instances of activity: "Sub Process#1"	1.99582	0.06455	1	3
<input type="checkbox"/> Coexistent instances of activity: "Subtask 1#1"	1.99582	0.06455	1	2

Figure 2: The report's activity details

- Figure 3 shows the event details, which can directly be linked to the gateway details from figure 4
- Due to our configuration of the subprocess' gateway depending on the stochastic variable x,
  - the sending cancel end event "Archive" has been executed most times
  - the default end event and the second cancel end event have been executed nearly similar ("#12" with 56 times, and "Operator#2" with 65 times)

### Interarrival Times

	Title	Obs	Mean	Std.Dv	Min	Max
●	Interarrivaltime for start event: ""	238	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s
●	Interarrivaltime for start event: ""	238	0d, 1h, 0m and 0s	0d, 0h, 0m and 0s	0d, 1h, 0m and 0s	0d, 1h, 0m and 0s
●	Triggered event "#12" Interarrivaltime	55	0d, 4h, 10m and 54s	0d, 2h, 55m and 54s	0d, 1h, 0m and 0s	0d, 14h, 0m and 0s
●	Triggered event "#2" Interarrivaltime	234	0d, 0h, 59m and 44s	0d, 0h, 48m and 18s	0d, 0h, 0m and 0s	0d, 4h, 0m and 0s
⊗	Triggered event "Archive#1" Interarrivaltime	115	0d, 2h, 1m and 2s	0d, 1h, 9m and 16s	0d, 1h, 0m and 0s	0d, 6h, 0m and 0s
⊗	Triggered event "Archive#2" Interarrivaltime	115	0d, 2h, 1m and 2s	0d, 1h, 9m and 16s	0d, 1h, 0m and 0s	0d, 6h, 0m and 0s
⊗	Triggered event "Operator#1" Interarrivaltime	64	0d, 3h, 40m and 18s	0d, 2h, 40m and 1s	0d, 1h, 0m and 0s	0d, 14h, 0m and 0s
⊗	Triggered event "Operator#2" Interarrivaltime	64	0d, 3h, 40m and 18s	0d, 2h, 40m and 1s	0d, 1h, 0m and 0s	0d, 14h, 0m and 0s

Figure 3: The report's event details

### Chosen paths at conditional gateways

	Title	Obs	Least	Most
◇	Chosen paths at exclusive gateway: "#13"	237	#17	#18

Cell	Object	n	%	Cum. %
■	#16	65	27.42616	27.42616
■	#17	56	23.62869	51.05485
■	#18	116	48.94515	100

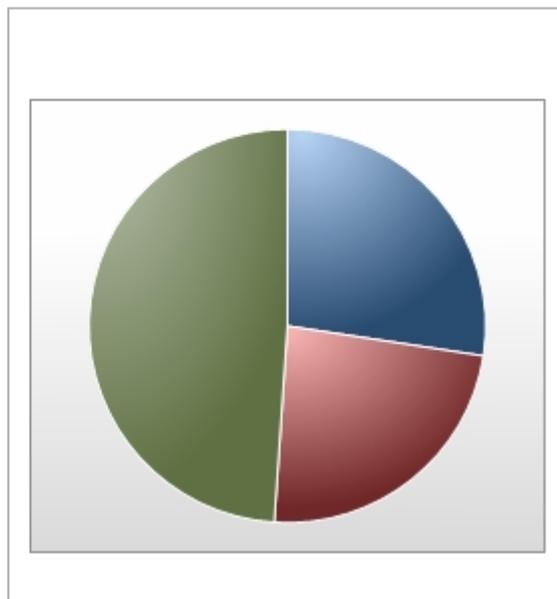


Figure 4: The report's gateway details

## 6. Scripting

This section gives a short introduction on how to handle inter-arrival times with a Python script.

### Content

- [Model 12 - Scripted Delay](#): Learn how to use a simple Python script to define gateway conditions and time instants based on the [Simulation's time](#).

### Model 12 - Scripted Delay

### Goals of this lesson

- Learn how to use a Scripted Delay for simulation purposes

Model 12 shows a work day process. The work starts at 9 a.m. and has several pauses. After each pause the branching Gateway checks, whether the simulation time is past 5 p.m. If this condition evaluates to true

the process terminates at the end event, otherwise the process will return to the task.

With the tools described until now, we are not able to define an exact time (or date), to start a process. That's what Scripted Delay was developed for. It enables the user to define a point in time, when something should happen depending on the simulation time. The denotation Scripted Delay originates from the programming language Python, that is used here.

## How-to

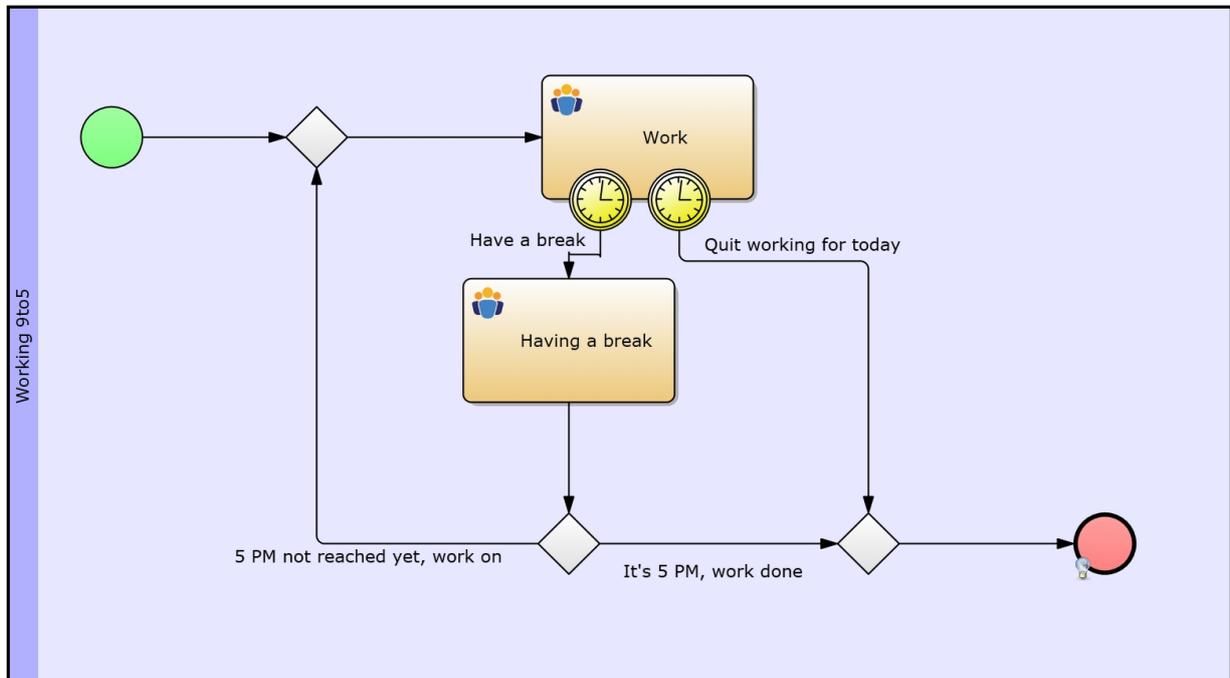


Figure 1: Model 12 - Scripted Delay / [IYOPRO-Link](#)

### Approach

1. We will define the scripted inter-arrival time at the start event first.
2. Parametrize the two timer events, that are attached to the activity "Work"
3. Create the conditions for the branching gateway

### Scripted inter-arrival time

- Click the start event (make sure you selected simulation from the property set)
- Open up the Time Editor to set up the inter-arrival time of the start event
- Now select Time Instant from the drop-down-box in the top-left corner of the time editor
- From the second drop-down-box (right to the previous one) select Scripted Time Instants
- The Time Editor now contains an input-box, in which you can write your Python code
- Put this line into the text area: `Result=SimulationContext.Time.GotoHour(9);` (see figure 2)
  - This tells the simulation to advance to 09:00 o'clock when the event is being initialized by the simulation
  - Note: If the simulation time is past 09:00 o'clock it will advance to the next day's 09:00 o'clock

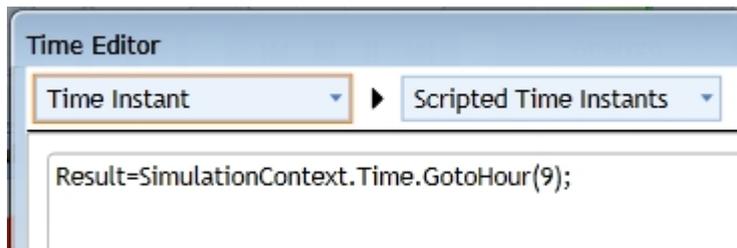


Figure 2 - Assigning the starting time for the start event

### Configuring the attached events

- "Have a break" contains a constant inter-arrival time of 1 hour.
  - The activity "Work" lasts 8 hours, which means that it will be interrupted every hour to have a break
  - "Having a break" lasts 0.1 hours or 6 minutes.
- "Quit working for today" has a scripted time assigned, see figure 3 for the code
  - First we check whether the current time has not yet reached 17 o'clock (SimulationContext.Time.Hour < 17)
  - If this is true: We assign the arrival time of the event to 17 o'clock
  - If this if false: We assign the arrival time to the current time, which means that it will be executed right away

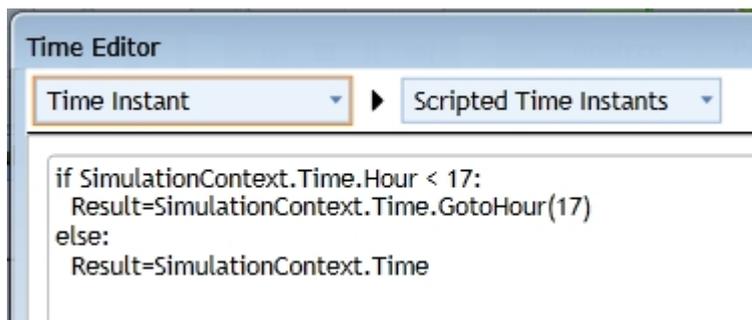


Figure 3 - Assigning the arrival time of event "Quit working for today"

### The conditions at the exclusive gateway

- We also need to check whether the employee has to quit working after the "Having a break" activity
- Therefore put the condition depicted in figure 4 on the left sequence flow at the branching exclusive gateway
  - We again check whether the current time is not past 17 o'clock by evaluating: SimulationContext.Time.Hour < 17
  - If true: the employee has to go on working
  - if false: the employee can quit
- The second condition checks the exact opposite: SimulationContext.Time.Hour >= 17
  - The current time must be equal or bigger than 17
- Of course we could make this more simple by putting up only one condition and then make the other edge a default sequence flow

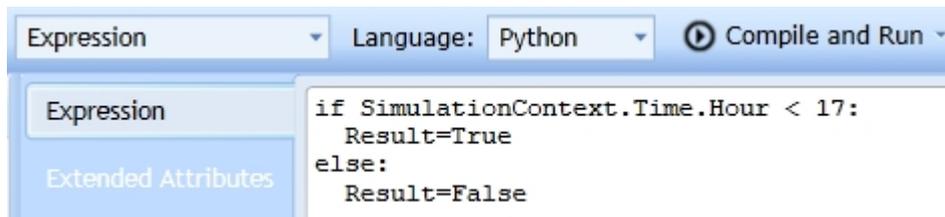


Figure 4 - The conditions at the branching exclusive gateway

### Explanation

As described in the introduction of this lesson, we modeled a simple work-from-9-5-process.

1. The process starts at 9 o'clock
2. The "Work" activity is being executed for 8 hours
3. There are two events attached to the activity
  - a) "Have a break" will be executed 1 hour after the activity has been started
    - i) The process then executes the "Have a break" activity
    - ii) Afterwards (at the gateway) it decides whether to work on
    - iii) This again depends on the current time. If it is past 17 o'clock the process will finish
    - iv) 'Elsewise it will return to the "Work" activity
  - b) At the "Quit working for today" event the simulation checks the Python script
    - i) If the time is past 17 o'clock, the event will be executed instantaneously
    - ii) Else it will be scheduled to 17 o'clock

### Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

### Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

### Report statistics

- From figure 1 we learn, that the processing time of this model is exactly 8 hours
- Figure 2 shows us that the "Having a break" activity lasts exactly 6 minutes
- These are the values with which we configured the model, so far so good
- But what about the mean value of the "Work" activity
  - We set up a constant duration of 1 hour. So why is it not exactly 1 hour?
  - Because our worker is having a break 7 times a day, which sums up to 42 minutes a day.
  - After he had 7 breaks, he still has 18 minutes to work before the second (attached) event fires and ends the process (see Min-Value from figure 2)
  - Just do the math: He works 7 hours and 18 minutes which is 7.3, then divide this by 8 (Observations per day) and multiply it with 60.
    - This results in 54.75 minutes, or 54 minutes and 45 seconds

Process cycle times

Title	Mean	Std.Dv	Min	Max	Lower Quartile	Median	Upper Quartile
Processing time	0d, 8h, 0m and 0s	0d, 0h, 0m and 0s	0d, 8h, 0m and 0s				

Figure 1: The processing times

Activity runtimes and waiting times						
Title	Obs	Mean	Std.Dv	Min	Max	
Processing time of activity: "Having a break#1"	70	0d, 0h, 6m and 0s	0d, 0h, 0m and 0s	0d, 0h, 6m and 0s	0d, 0h, 6m and 0s	
Processing time of activity: "Work#1"	80	0d, 0h, 54m and 45s	0d, 0h, 13m and 58s	0d, 0h, 18m and 0s	0d, 1h, 0m and 0s	

Figure 2: The activity details

Event counter		
Title	Current Value	
Triggered event "#4"	10	
Triggered event "Have a break#1"	70	
Triggered event "Quit working for today#1"	10	

Interarrival Times					
Title	Mean	Std.Dv	Min	Max	
Interarrivaltime for start event: ""	1d, 0h, 0m and 0s	0d, 0h, 0m and 0s	1d, 0h, 0m and 0s	1d, 0h, 0m and 0s	
Triggered event "#4" Interarrivaltime	1d, 0h, 0m and 0s	0d, 0h, 0m and 0s	1d, 0h, 0m and 0s	1d, 0h, 0m and 0s	
Triggered event "Have a break#1" Interarrivaltime	0d, 3h, 13m and 33s	0d, 5h, 31m and 47s	0d, 1h, 6m and 0s	0d, 17h, 23m and 59s	
Triggered event "Quit working for today#1" Interarrivaltime	1d, 0h, 0m and 0s	0d, 0h, 0m and 0s	1d, 0h, 0m and 0s	1d, 0h, 0m and 0s	

Figure 3: The event details

- Also note that the gateway always fires the same sequence flow (see figure 4)

Chosen paths at conditional gateways				
Title	Obs	Least	Most	
Chosen paths at exclusive gateway: "#2"	70	5 PM not reached yet, work on#1	5 PM not reached yet, work on#1	

Cell	Object	n	%	Cum. %
	5 PM not reached yet, work on#1	70	100	100

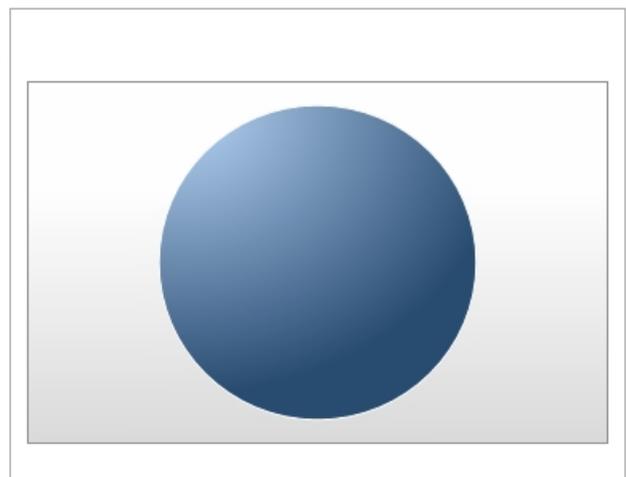


Figure 4: The gateway details

## 7. Resources

This chapter talks about Resources. A Participant, that is part of the execution of the activity, can be assigned to every activity. Although the Participant normally represents a human resource, you are free to expand the term to machines, storage capacities, loading vehicles, and so on.

## Prerequisites

If you want to assign resources to your models, you need IYOPRO organization diagrams to define them. More information can be found in the IYOPRO documentation.

The raw process to create a model with resources is as follows:

1. Create a BPI-Project
2. Create the organization diagram like the one from Figure 1
3. Add a collaboration diagram
4. Import the roles at any activity or pool / lane from the organization diagram
5. Now you can chose the roles at every activity or pool / lane

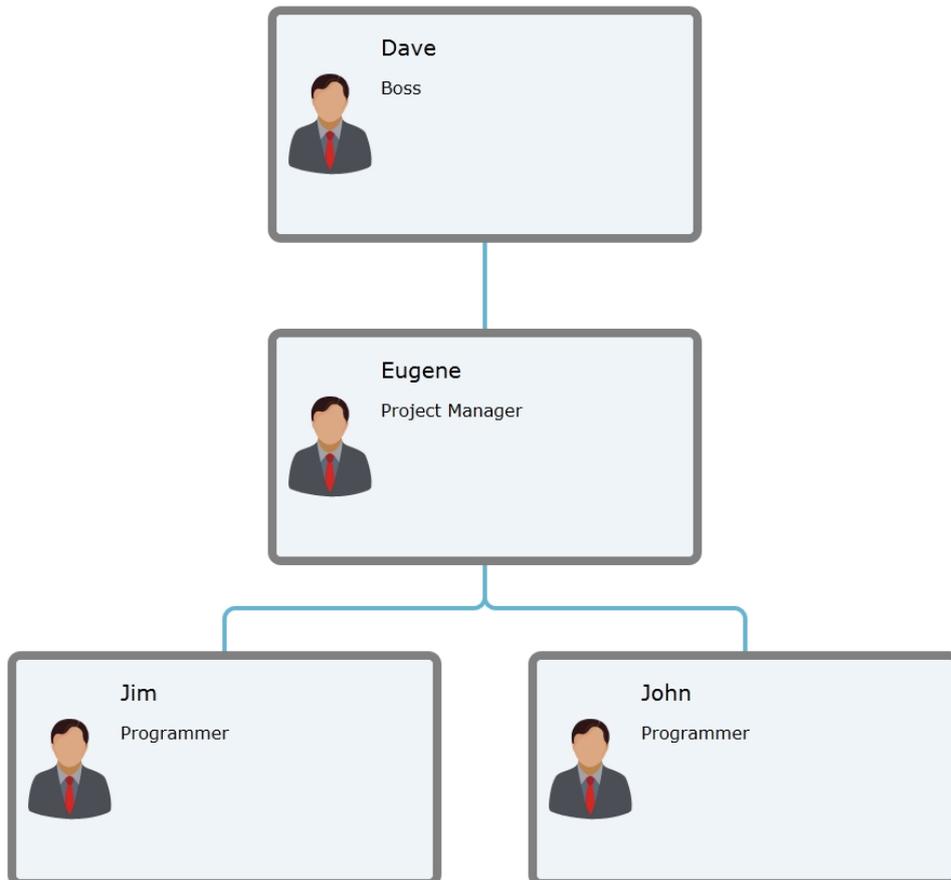


Figure 1: The Organization diagram that will be used for the following lessons

### Model 13 - Simple Example

#### Goals of this session

- Learn how to deploy resources to the simulation

In this lesson we will introduce resources to the simulation. Resources can be used to determine bottlenecks in the real-life resource allocation.

Model 13 simulates the management's work day. It starts on work days at 8 p.m., works for 8 hours and ends. As you can see, the activity Work uses a Participant, and the assigned role has to be a Boss.

#### How-to

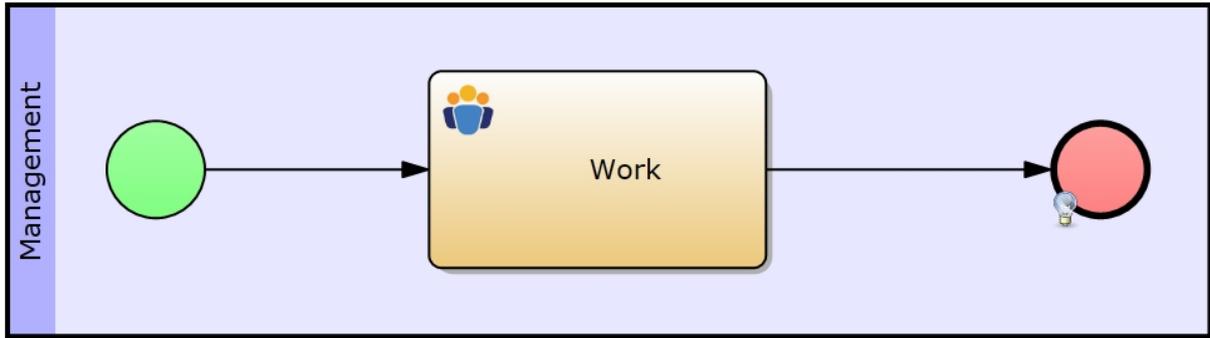


Figure 1: Model 13 - Simple Example for the Usage of Resources

### Approach

- Click on the Edit button next to the Participant property (see Figure 2 below)
- The Participant Editor shows up at the bottom of IYOPRO (see Figure 3 below)
- Click the Add button to add one participant to this activity
  - Note: Participants can only be assigned to User Tasks
- IYOPRO's Role Editor shows up, in which you can specify the participant's role (see Figure 4 below)
- Now select the role Boss from the drop-down-box and click Ok
- You may also specify an amount of the role to be used at this activity
  - Leave it to 1, because we have only got one Boss.

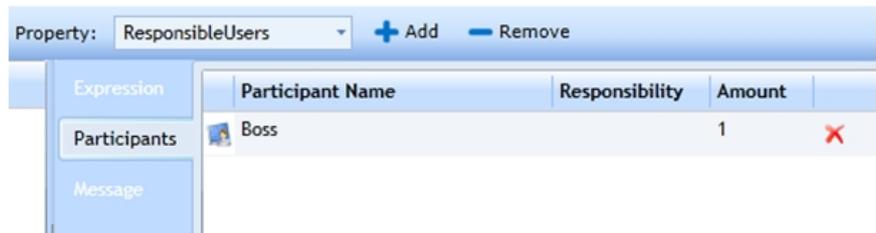
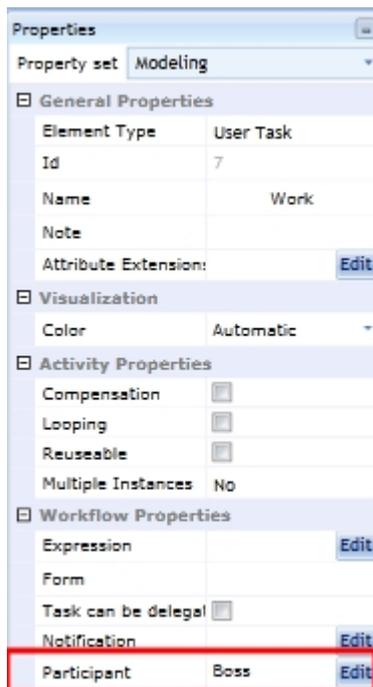


Figure 2: Simulations Properties Figure 3: IYOPRO's Participant editor of an activity - The Participant property highlighted

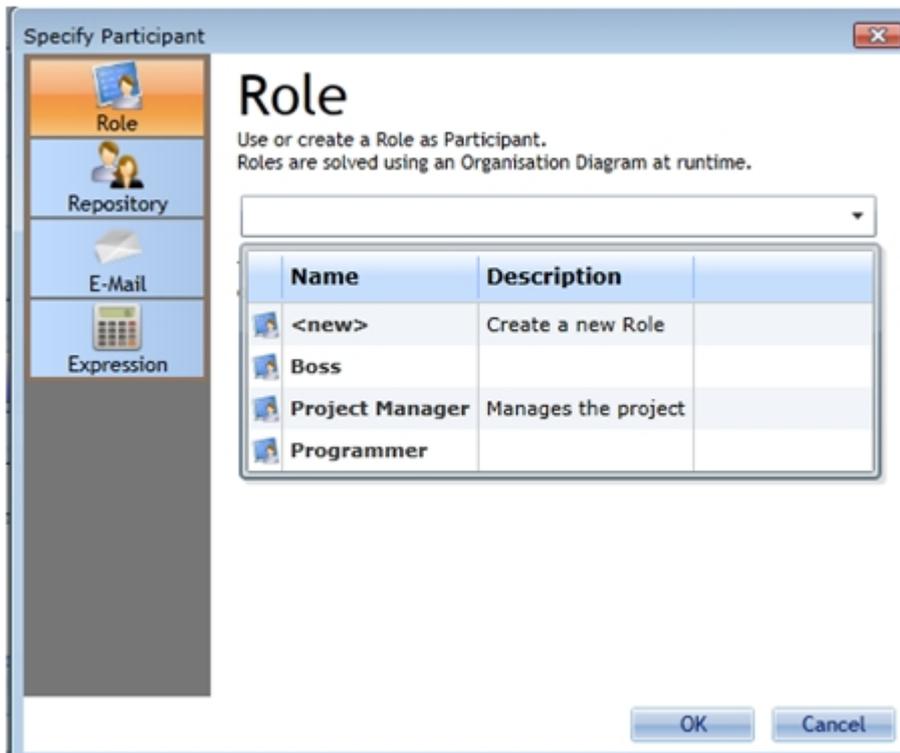


Figure 4: IYOPRO's Role Editor

## Explanation

Our process from Figure 1 works from 9 a.m. to 5 p.m. We also assigned a participant to the only activity, which must be capable of the role Boss. Have a look at the [report](#).

## Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

## Report

If you take a look at the report, you can see that there is a new category called Resource Details. See Section [Report > Gateways & Resources](#) for information on the resource specific report data.

## Diagrams

- Figure 1 depicts that the resource Boss was idle for 640 TU and in use for another 160 TU
  - This depends on our simulation parameters:
  - Simulation time of 26 days, working only on work days and work for 8 hours
  - Just do the math:
    - 26 days of which 20 days are work days
    - and 20 days multiplied with 8 hours work per day equals to 160 total working hours
- The other resources from our Organization Diagram were idle for the whole time because they have not been used in our model
- Further on you can find pie and bar charts in the Resource Details of the report for each resource
  - Although numbers are nice, we also offer pie charts which often allow for fast recognition of

obvious correlations

Title	Idle	Waiting	In use	Blocked
Resource Pool for Boss	640	0	160	0
Resource Pool for Project Manager	800	0	0	0
Resource Pool for Programmer	1600	0	0	0

Figure 1: The report's resource pool statistics of the resource details

### Queue statistics (advanced)

The last part of the Resource Details consists of detailed waiting queue statistics. Every time a process requires one or more resources, the process is queued for the required resource pool. In this example the process is always queued in the pool of the boss role and waits there until the resource is available.

Let us have a look at the statistics depicted Figure 2:

- It is recognizable that a process instance was pulled 20 times from the queue (Obs)
- The maximum queue size was 1 (Qmax)
  - which tells us, that there was one process instance waiting at the most and therefore there was no shortage in the provision of the resource
- Additionally there are 20 Zero-Values, which occur when a process instance was queued and instantly pulled from the queue because the resource was available.
  - These 20 Zeros conform with the Obs value, because no process instance had to wait for a resource.
- This also manifests in the waiting times (max. Wait and avg. Wait) with a value of 0. Not a single process instance had to wait for a single second until it got its resource needs.
- These four value shows us, that there was no bottleneck in the process (depending on the resources)

Title	Qorder	Obs	QLimit	Qmax	Qnow	Qavg.	Zeros	max.Wait	avg.Wait	refus.
Resource Pool for Boss	FIFO	20	unlimit.	1	0	0	20	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0
Resource Pool for Project Manager	FIFO	0	unlimit.	0	0	0	0	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0
Resource Pool for Programmer	FIFO	0	unlimit.	0	0	0	0	0d, 0h, 0m and 0s	0d, 0h, 0m and 0s	0

Figure 2: The report's queue statistics of the resource details

### Note

If a process needs two different resources, it is queued in both resource pools. Sometimes this results in waiting for one resource, while the other resource is already available. Then the process is dequeued from the available resource and stays in the queue of the unavailable one.

### Resource Shift Plan

In reality you often face the "problem", that resources do not work 24 hours a day, especially humans are prone to this fact. Therefore you need a feature that allows you to model a resources working time. We offer you a shift editor, in which you can set every resource's labor time for each day of the week.

### Approach

- Open an Organization Diagram (e. g. the one from Model 13)
- Click on any resource

- Make sure you have selected Simulation from IYOPRO's property set
- Now select the Simulation tab from IYOPRO's top menu bar
- You should see the Shift-button as depicted in figure 1
- Click on it

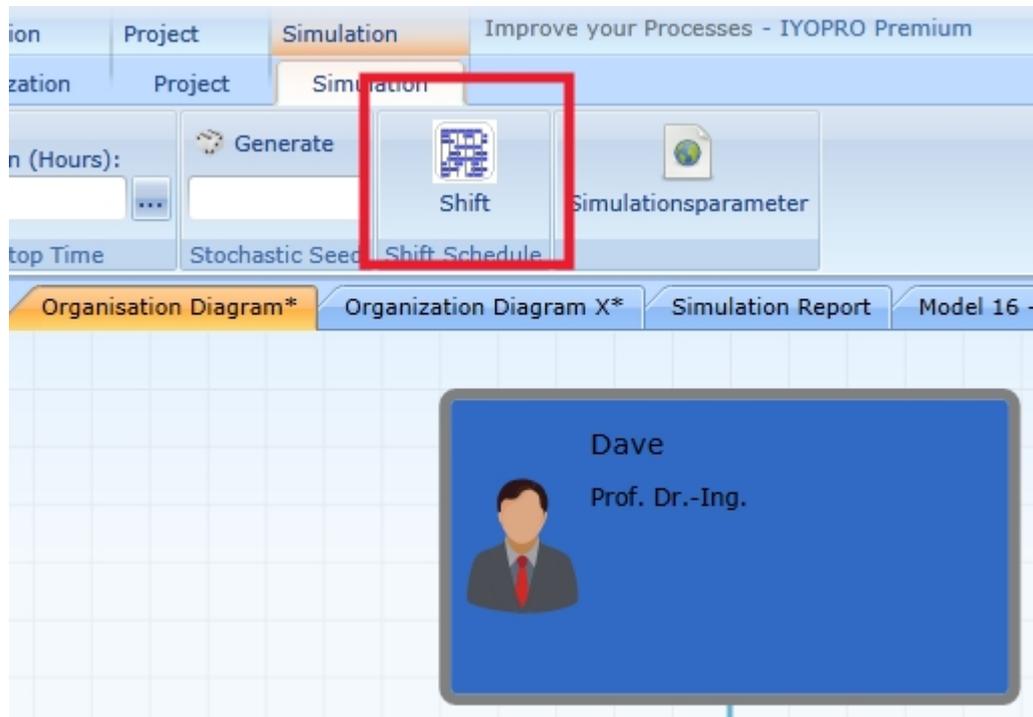


Figure 1: Location of shift editor button

- The Shift Editor appears
- Here you can set the labor time of each resource, that is contained in the organization diagram
- You can the start and end time of the resources working time for each day of the week
- In the drop-down-box below the string "Shifts for Dave" we offer you some preset shifts like
  - Full week: The resource works 24 h each day of the week (see figure 2)
  - 30 hours week with a lunch break: The resource works 30 hours a week and also has a lunch break of 1 hours included
    - You can interrupt a day's working time for a break like depicted in figure 3.
    - Or you can model night shifts, which start at 10 PM and stop at 6 AM next day.
    - Use the Plus (+) and Minus (-) buttons to achieve this

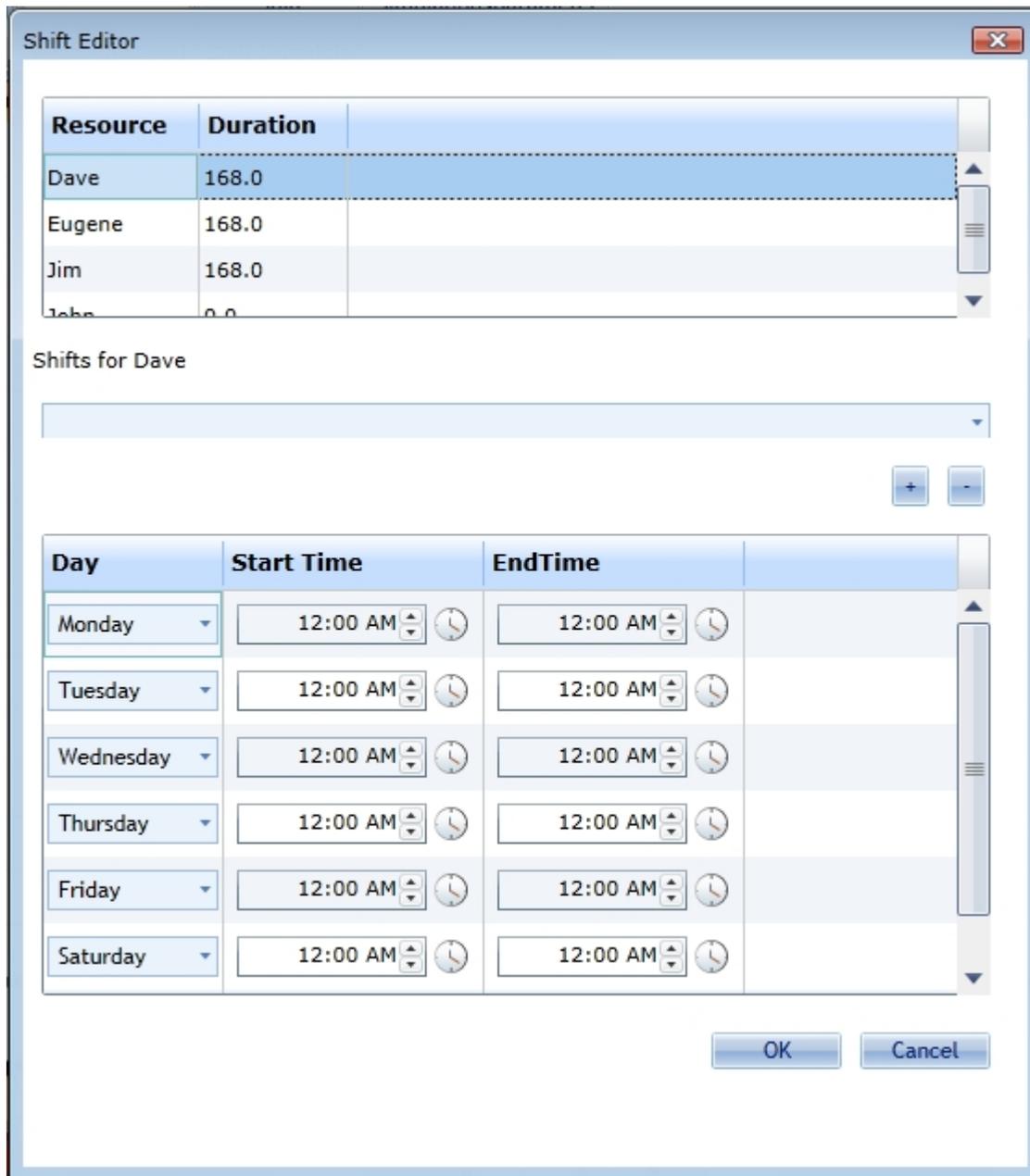


Figure 2: Shift editor for resources (Full Week preset)

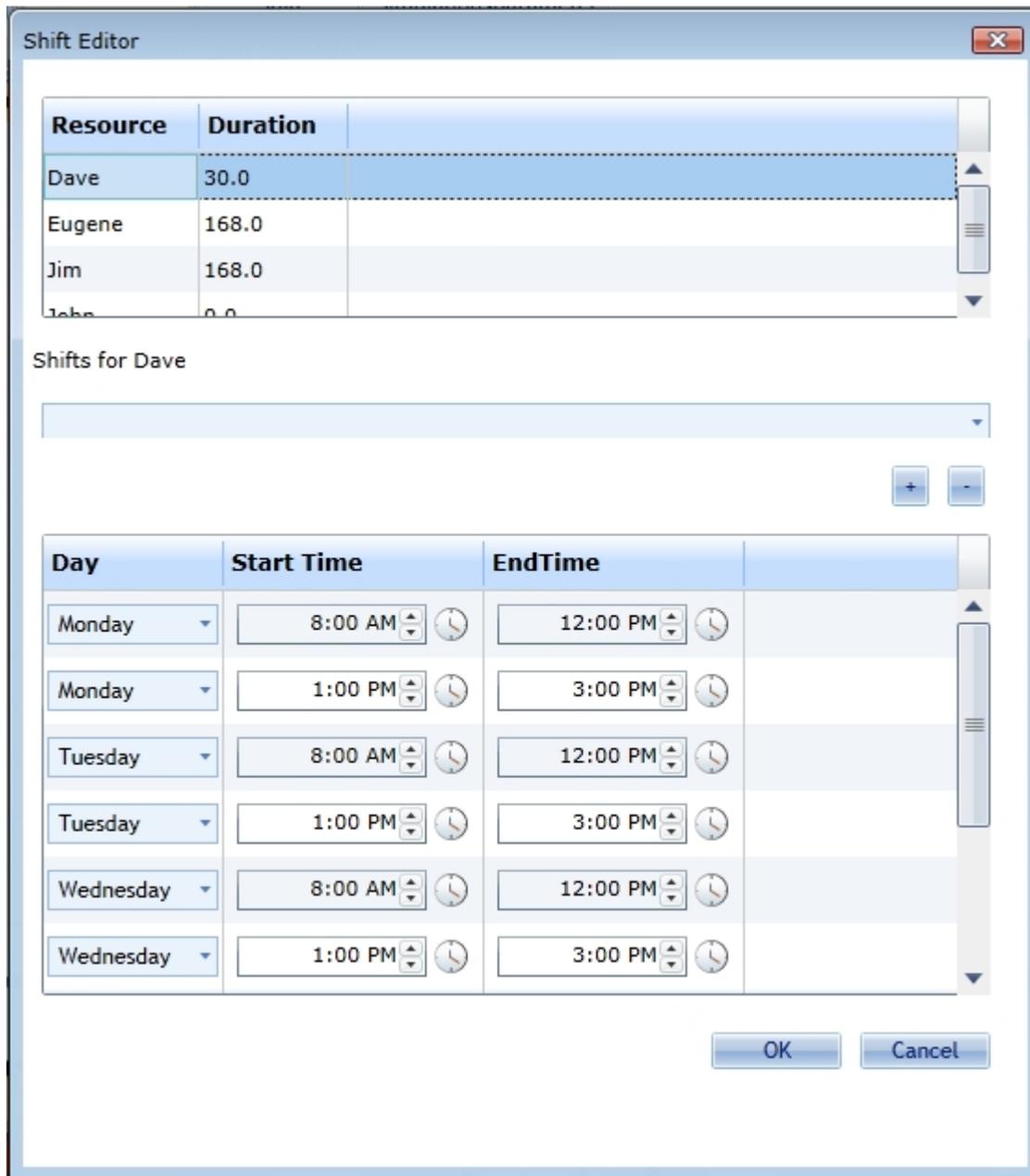


Figure 3: Shift editor for resources (30 hours with lunch break preset)

## Model 14 - Resources for Pools and Lanes

### Goals of this lesson

- Learn how to assign resources to pools and lanes
  - and by that assign them automatically to each activity, that lies in the pool or lane

Often we need to model divisions of companies. These divisions mostly use the same participant(s) throughout a pool or lane.

Therefore in IYOPRO resources can not only be assigned to activities, but also to pools and lanes. Model 14 consists of two pools, with the upper pool using the developer as participant. That means, every activity in this pool needs a developer resource. You will notice that the Simulation Properties of the Develop External activity has the option Inherit Resources, that is checked by default. If you do not want an activity in a pool to inherit the resources from the pool, you can deactivate this option.

### How-to

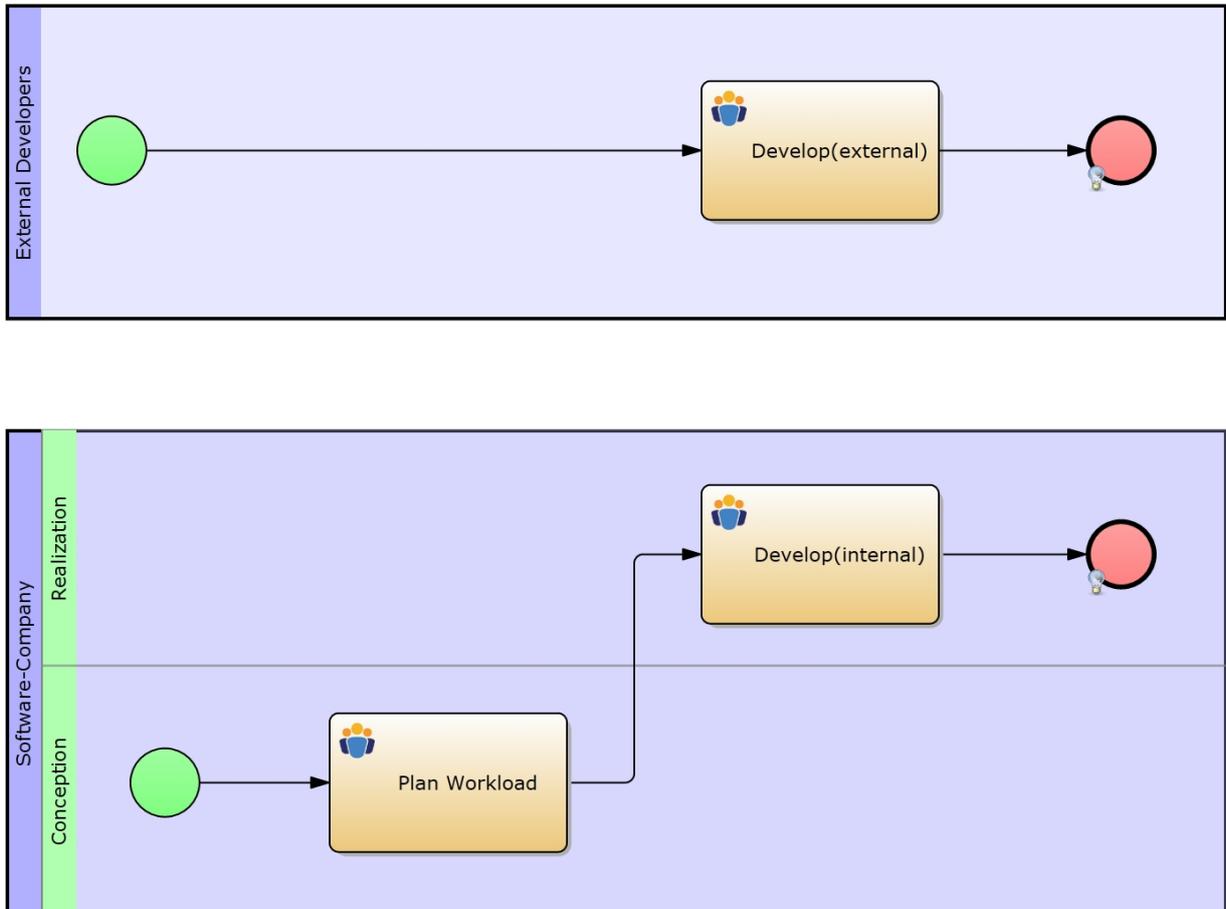


Figure 1: Model 14 - Resources for Pools and Lanes

### Approach

- Select the top pool
- Assign it the participant Programmer (just like you learned it in the previous lesson)
  - Under the Optional Simulation Properties of a User Task you will find the property Inherit Resources
  - This option tells the simulation, whether the specific activity in this pool shall inherit its resource needs
  - (It is activated by default. Just uncheck it, if you do not want the activity to inherit its pool's or lane's resource needs)
- Select the bottom pool
- Assign a participant called Project Manager to the pool
- Now select the top lane (Realization) of the pool
- And assign it a participant called Programmer

### Explanation

What we did was to assign a programmer to the top pool. Since the activity "Develop (external)" has its Inherit Resources option activated, it will always need one programmer to fulfill its duty.

In the bottom pool we assigned a Project Manager, which will be required by both activities. Then we also assigned a Programmer to the top lane. This means that the activity "Develop (internal)" will need one Project Manager and one Programmer to be completed.

### Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

## Report statistics

- Figure 1 shows us that both developers have been in use (almost) throughout the whole simulation. (The 3 hours of idle time result from the specifics of the start events' inter-arrival times).
- In contrary, the Project Manager has been in use for only 55 hours and was waiting for one of the Developers for 184 hours. Remember that the activity Develop (internal) requires one Project Manager and one Developer.

Resource Pools							
Title	Idle	Waiting	In use	Other role	Setup	Post-processing	Basic Load
Role: Boss	240	0	0	0	0	0	0
Role: Boss of X	240	0	0	0	0	0	0
Role: Developer	3	0	477	0	0	0	0
Role: Division Manager	480	0	0	0	0	0	0
Role: Project Manager	1	184	55	0	0	0	0

Figure 1: Resource statistics for Developer and Project Manager

- Let's take a look at Figure 2, which shows the utilization of the Developer role
- We learn that the pool External Developers has been acquiring both developers' working time for 94.30 % throughout the simulation. The Pool Software-Company got only 5.7 % of their working time.

## Utilization of Role: Developer in processes



Figure 2: Utilization of the Developer role

- Figure 3 depicts the queue statistics for the processes waiting for either one of the Developers or the Project Manager
- Due to the fact, that the Project Manager is being held up at activity Develop (internal), all the process instances arriving at activity Planning Workload must wait disproportionately high.

## Queues

Title	Qmax	Qnow	Qavg.	Zeros	max.Wait	avg.Wait
Waiting Queue at Role: Boss	0	0	0	0	0s	0s
Waiting Queue at Role: Boss of X	0	0	0	0	0s	0s
Waiting Queue at Role: Developer	15	15	9.29	0	14h	8h 31min 26s
Waiting Queue at Role: Division Manager	0	0	0	0	0s	0s
Waiting Queue at Role: Project Manager	212	212	101.62	0	7d 19h	2d 19h 17min 46s

Figure 3: Queue-Statistics for the Developer and Project Manager

- From figure 4 (Activity Runtimes) we learn that this is exactly the case
  - The Project Manager finished the activity Plan Workload (executed 28 times [Obs]).
  - But the activity Develop(internal) has been executed only 27 times.
  - So the Project Manager is currently held at Develop(internal)
  - See this chapter's subpage [Prioritization of Resources](#) for more information on this matter

Activity runtimes and waiting times						
Title	Obs	Mean	Std.Dv	Min	Max	
Processing time of activity: "Develop(external)#1"	224	2h	0s	2h	2h	
Processing time of activity: "Develop(internal)#1"	27	1h	0s	1h	1h	
Processing time of activity: "Plan Workload#1"	28	1h	0s	1h	1h	
Total time of activity: "Develop(external)#1"	224	10h 36min 25s	3h 28min 49s	2h	16h	
Total time of activity: "Develop(internal)#1"	27	7h 31min 6s	3h 59min 3s	1h	14h	
Total time of activity: "Plan Workload#1"	28	2d 20h 23min 34s	2d 14h 7min 16s	1h	9d 7h	
Waiting time for resources at activity: "Develop(external)#1"	226	8h 39min 1s	3h 29min 44s	0s	14h	
Waiting time for resources at activity: "Develop(internal)#1"	27	6h 31min 6s	3h 59min 3s	0s	13h	
Waiting time for resources at activity: "Plan Workload#1"	28	2d 19h 23min 34s	2d 14h 7min 16s	0s	9d 6h	

Figure 1: Activity Runtimes

### Explore

- Try to explore Model 14's dynamics by adding an additional Developer to the Organization Diagram.
  - On first thought, this measure should decrease the Project Manager's waiting time for a Developer and consequently the waiting processes in the queues and the processes' lead times.
  - Right? Try it! You will be surprised! Please try to understand on yourself, what happens there by adding another Developer.
  - You can, add another Project Manager to improve your understanding of the model's dynamics.

### Prioritization of Resources

In Model 14's report page we found out, that the Project Manager has to wait for a Developer at activity Develop(internal) disproportionately long. This bases on the fact, that all the activities have the same resource priority assigned. The resource manager (internal feature of the simulation) allocates the process instances to the respective resource's queue. Unfortunately the resource manager does not provide a sophisticated way to improve this allocation. It simply checks the activity's resource priority, and then sorts the queue using this priority.

Furthermore, the resource manager inserts the lastly arrived process instance in the first place of the respective sorted sequence in the queue.

Example:

- One process instance of pool External Developers (PIED) has been started.
  - It needs one Developer to work off the activity Develop (external)
  - Let's assume, that there is no Developer available currently, so the PIED is being inserted into the Developer's queue
- At the same time a process instance from the pool Software-Company (PISC) finishes the activity Plan Workload
- It now reaches the activity Develop (internal).
  - Since it already has the Project Manager available it needs one of the Developers.
  - Due to the fact, that all queues are using the first-in-first-out-principle and that all activities' resource priorities are the same, the PISC is being inserted behind the PIED into the Developer's queue
- While that PISC waits for Developer, other PIEDs are being started (inter-arrival time of 1 hour) and inserted into the Developer's queue
- Remember that all starting PISCs must wait at activity Plan Workload, because they wait for the single Project Manager, which is being held at activity Develop (internal)

This procedure is repeatedly executed. So, every following PISC has to wait longer for a Developer than the previous one. This is the result of equal prioritization of all activities in reference to the resources. All activities have the same importance for the queue and they are worked off using the FIFO (first in, first out) principle.

To avoid the behavior from the previous example, you can set the Resource Priority in the Simulation Properties of an activity (Optional Simulation Properties, see the figure below). This property defines the priority of an activity for the resource queue. There are five different levels (lowest, low, medium, high and highest). These levels are used to sort the activities in the queues of a resource pool, i.e. the higher the level, the more an activity is sorted to the fore.

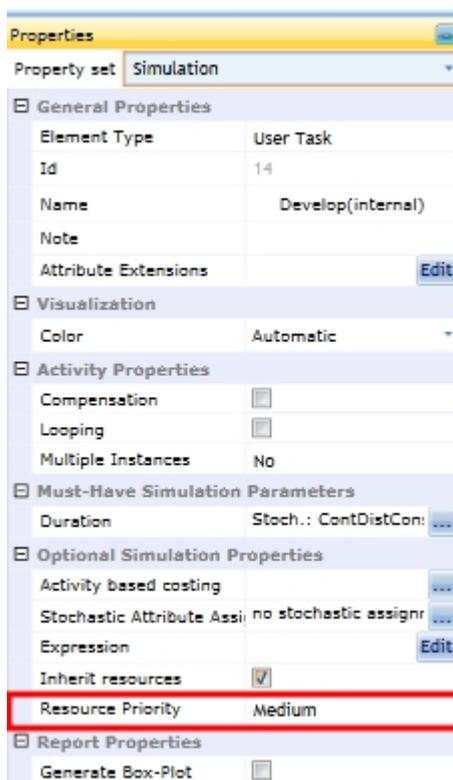


Figure 1: Simulation Properties of an activity - Resource Priority property highlighted

## Change Model 14's resource prioritization

Take Model 14 as an example and set the Resource Priority of the Develop(internal) activity to high. After simulating the model, you will discover a much better utilization of the Project Manager. See for yourself!

## Model 15 - Setup and Post-processing time

### Goals of this lesson

- Learn what setup and post-processing times are used for at resources and how to assign them

There are resources (typically machines) that need a specific amount of time to get set up before they can work properly. On the other hand, after finishing a task resources may need some time, before they can be used again (post-processing time).

Therefore you can assign setup or / and post-processing times for resources in IYOPRO.

We will use Model 15 to explain this feature. This model is quite simple. It contains one start event, one activity and one end event. Let's assume we describe the process of a meeting, in which the boss of company X talks to the Division Managers to assess the company's current situation. Also, let us assume, that the boss needs 30 minutes to get ready before the meeting, and that he needs one hour after the meeting to process all the information he has been given from his division managers.

### How-to

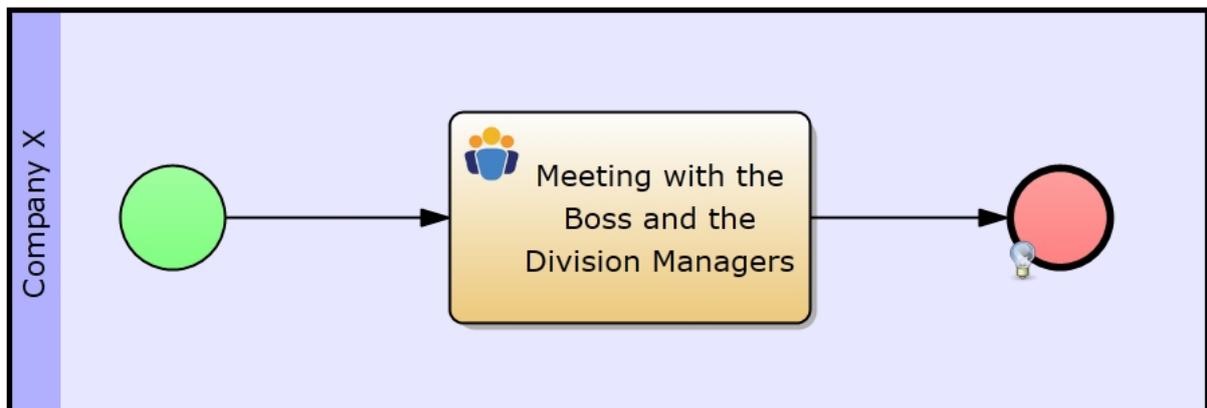


Figure 1: Model 15 - Setup and Post-processing time

### Approach

- Select the activity
  - Assign the participants Boss of X and Division Managers to it (just like you learned it in the previous lesson)
- Now go to the Organization Diagram X
- Select "Michael" from the organization diagram
  - Under the simulation properties you will now see the options Setup time and Post-processing time (see figure 3)
  - These are time spans, which can be set with the time editor, which you already are fond of
  - Set the following Distributions for:
    - Setup time: Constant Distribution with a constant value of 30 minutes or 0.5 hours
    - Post-processing time: Constant Distribution with a constant value of 1 hour
- Further, we assume that they meet every work day
- The duration of the meeting can be represented by a normal distribution with a mean value of 2 hours and a standard deviation of 30 minutes

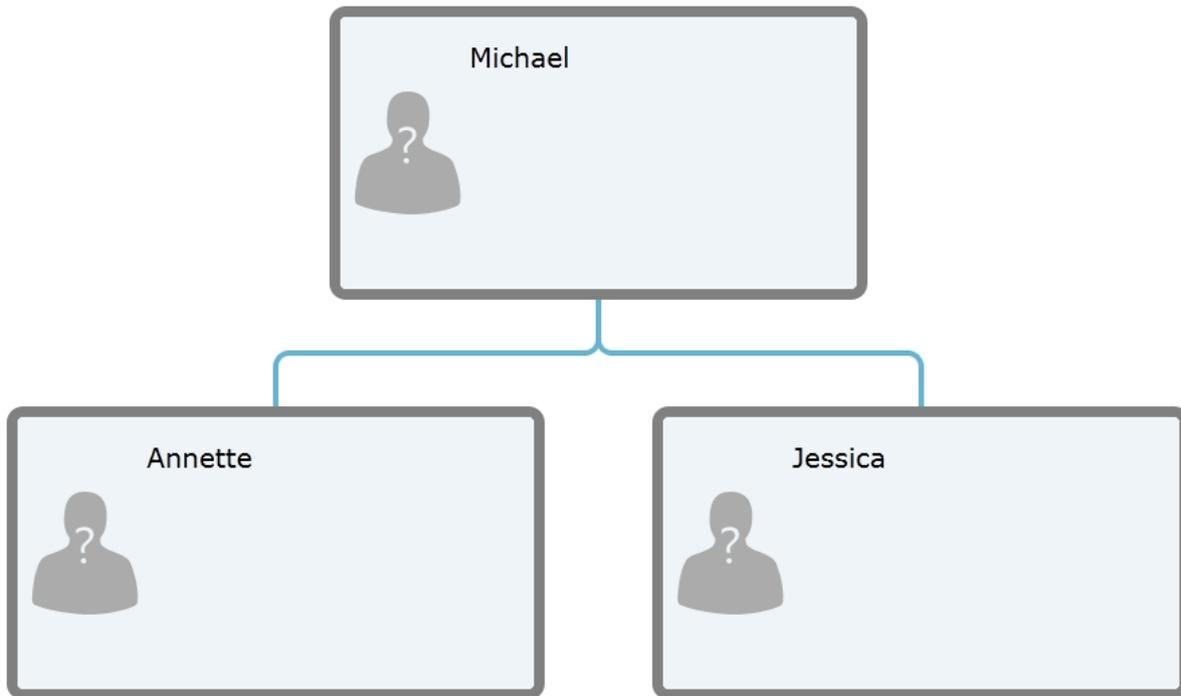


Figure 2: Model 15 - Organization Diagram

Property set: Simulation

General Properties	
Element Type	Human
Id	1
Name	 Michael
Note	
Attribute Extensions	<a href="#">Edit</a>
Visualization	
Color	Automatic
Optional Simulation Properties	
Costs	3
Has Basic Load	<input type="checkbox"/>
Payment	Wage rate
Post-processing Time	Constant(1h) <a href="#">...</a>
Setup Time	Constant(30min) <a href="#">...</a>
Time Unit	Day
Unit	Euro

Figure 3: Michael's simulation properties

### Explanation

We assigned a setup time of 30 minutes and a post-processing time of 1 hour to the role Boss of X. So, now every process instance has to acquire one Boss of X and two Division Managers when it reaches the activity. After all resource needs are met, the Boss of X's setup time starts, which means that the process instance has to wait 30 minutes before starting the activity. When the activity has been executed, the process instance moves on to the end event. The resource Boss of X is not available for other process instances yet, because it still has to post-process for 1 hour.

## Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

## Report statistics

- Figure 1 shows us the role Boss of X has been in use for 13.23 hours
- But the interesting part is that it dedicated 3.5 hours for to set up and 7 hours to post-process before and after the execution of an activity

Resource Pools							
Title	Idle	Waiting	In use	Other role	Setup	Post-processing	Basic Load
Role: Boss	240	0	0	0	0	0	0
Role: Boss of X	216.27	0	13.23	0	3.5	7	0
Role: Developer	480	0	0	0	0	0	0
Role: Division Manager	446.53	7	26.47	0	0	0	0
Role: Project Manager	240	0	0	0	0	0	0

Figure 1: Resource statistics for Boss of X

## Model 16 - Basic Load

### Goals of this lesson

- Learn what the basic load feature of a resource is used for and how to assign it

In business processes there are usually resources involved that work off incoming customer requests, but still need to handle other labor, that also needs to be done. Imagine an it-administrator who is part of the software-developer team of a company, but who also needs to take phone calls from customers. We do not want to discuss, whether this work-sharing is a sensible approach, but it is common in a lot of companies. On days with a high customer inquiry traffic the developer simply has cut off the service duties in order to keep up with his job's software-development part. We call this a resource's basic load.

Model 16 from figure 1 depicts the aforementioned circumstance: An IT-administrator belongs to a software-developer team, but he also has to work off incoming customer inquiries. Since he must fulfill his duties on the software-development team, he must make himself unavailable for the customer service, which can be done by assigning a basic load to its resource properties.

## How-to

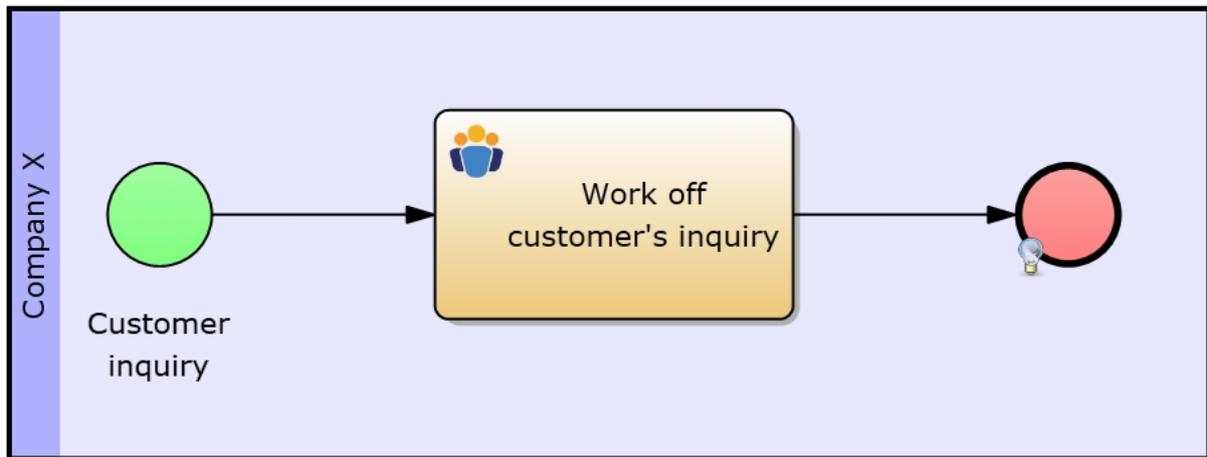


Figure 1: Model 16 - Basic Load

### Approach

- We will use Organization Diagram X from model 15 (see figure 2)
  - Chris and Joe belong to Annette's division and form Company X's developer team
  - They also have to attend to customer inquiries
  - On days with a high inquiry traffic they will need to cut off their customer service duties to fulfill their development tasks
- Select Chris or Joe
  - You will notice that there is a check box called Has Basic Load (see figure 3)
  - Check it and you will notice that two more options appear: Available Working Time and Basic Load Duration (see figure 3)
  - Available Working Time specifies the time span, after which the resource starts to work on its basic tasks. You could also call it the inter-arrival time for the basic load. But we called it Available Working Time because it represents the time, in which the resource can be assigned to other duties.
  - Basic Load Duration indicates the duration of the basic load.
  - So, when the time span given in Available Working Time passes, the basic load status is being initiated for the given duration, in which the resource will not be available to any process instance.
- In this case we assigned the following distributions for:
  - Chris
    - Available Working Time: Normal Distribution with a mean value of 1 hour and a standard deviation of 5 minutes
    - Basic Load Duration: Continuous Uniform Distribution with a lower border of 2 hours and an upper border of 3 hours
  - Joe
    - Available Working Time: Continuous Uniform Distribution with a lower border of 1 hour and an upper border of 2 hours
    - Basic Load Duration: Continuous Uniform Distribution with a lower border of 2 hours and an upper border of 3 hours
- Finally we assign the role Administrator to model 16's activity

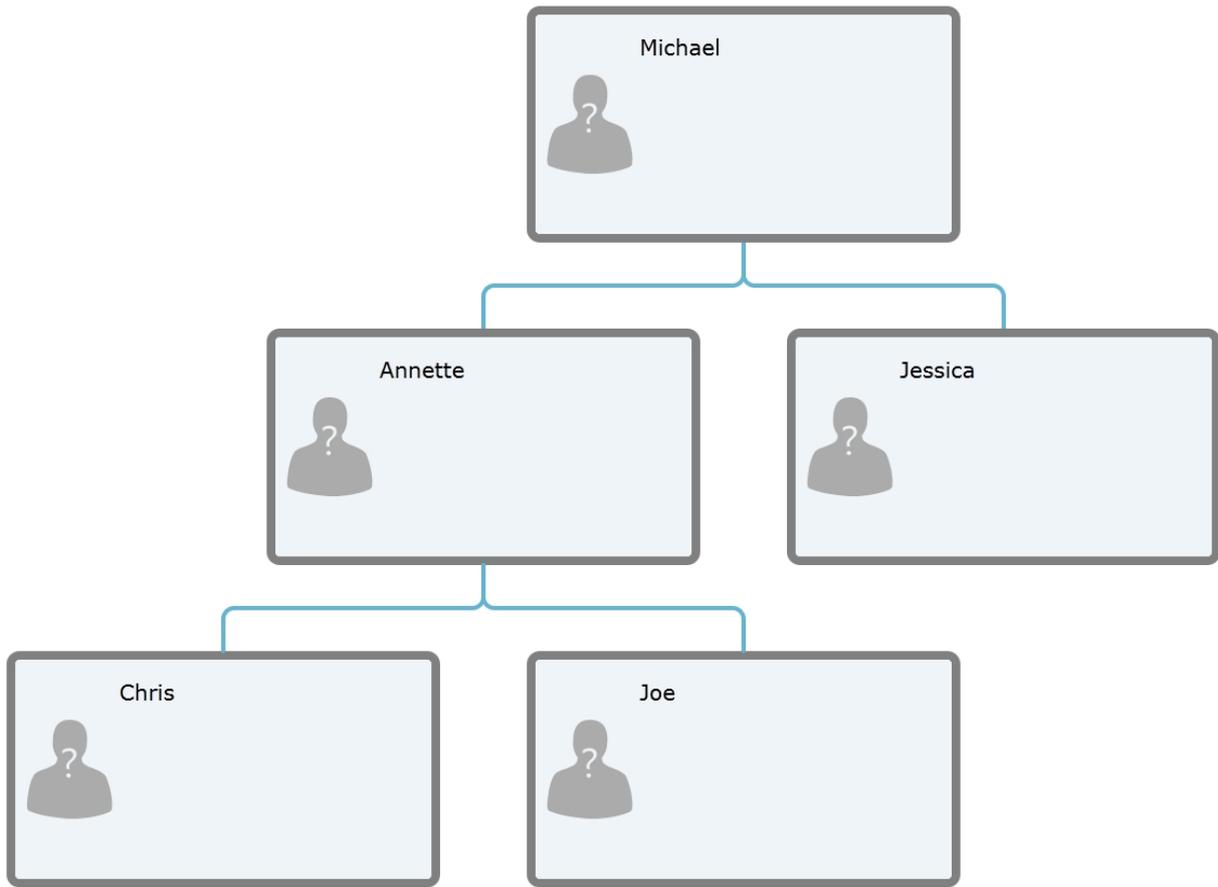


Figure 2: Model 16 - Organization Diagram

Property set: Simulation	
General Properties	
Element Type	Human
Id	6
Name	 Chris
Note	
Attribute Extensions	<a href="#">Edit</a>
Visualization	
Color	Automatic
Optional Simulation Properties	
Basic Load Duration	Normal(1h; 5min) <a href="#">...</a>
Available Working Time	Uniform(2h; 3h) <a href="#">...</a>
Costs	0
Has Basic Load	<input checked="" type="checkbox"/>
Payment	Wage rate
Post-processing Time	not selected <a href="#">...</a>
Setup Time	not selected <a href="#">...</a>
Time Unit	Day
Unit	Euro

Figure 3: Chris' simulation properties

## Explanation

We assigned a basic load to both Administrators. Let's have a look at the [report](#).

## Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2014 12:00:00 a.m.
- Simulation stop time: 240
- Simulation stop time unit: Hours

## Report statistics

- Figure 1 shows us the role Administrator has been in use for 359.68 hours
- But the interesting part is that it dedicated 116.93 hours to its basic load

Resource Pools							
Title	Idle	Waiting	In use	Other role	Setup	Post-processing	Basic Load
Role: Administrator	3.39	0	359.68	0	0	0	116.93
Role: Boss	240	0	0	0	0	0	0
Role: Boss of X	240	0	0	0	0	0	0
Role: Developer	480	0	0	0	0	0	0
Role: Division Manager	480	0	0	0	0	0	0
Role: Project Manager	240	0	0	0	0	0	0

Figure 1: Resource statistics for Administrator

## Explore

Try to increase or decrease one of the Basic Load elements of one of the Administrators to see the effect of the Basic Load feature on the process' cycle time.

## 8. Activity Based Costing

This sections provides a short example which introduces you to the usage of different cost types in activity based costing. The goal of this tutorial is to identify the different handling of cost centers and humans. This is especially important for a correct implementation of costs into the process model and a accurate interpretation of the costing details in the simulation report.

### Model 17

## Goals of this lesson

- Introduction for the use of organizational diagrams in Activity Based Costing
  - and assignment of cost factors to activities

The given example describes a simple customer support process which delivers product information for customers. Therefore a customer calls the customer support. The staff enlists the customers' wishes / questions in a ticket system and provides the required information afterwards. In this case we'll work with three cost factors – employee salary, costs for the ticket system license and energy costs caused by the use of personal computers.

## How-to

### Approach

### Modeling

- Create a collaboration diagram and model the process (figure 1)

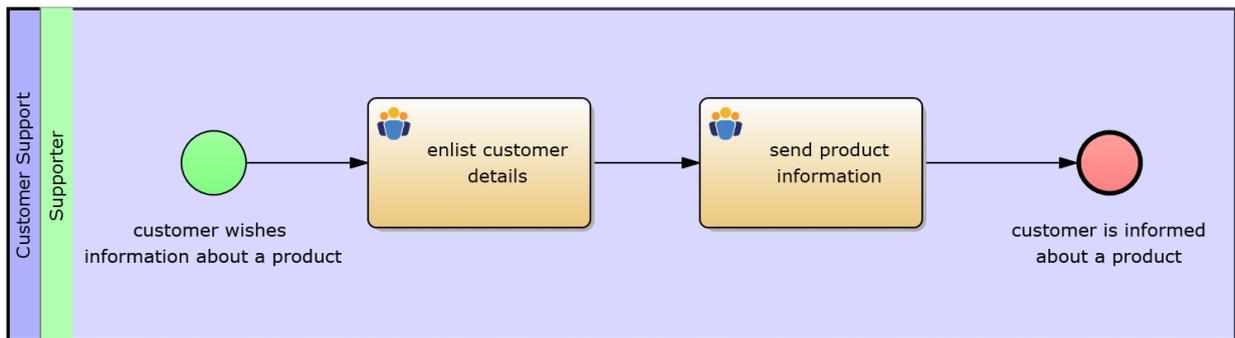


Figure 1: Collaboration Diagram of the example process / [YOPRO link](#)

- Create an organizational diagram and open it
  - Model the organizational set-up (figure 2)
  - Change the element types to 'Cost Center' and 'Human' in the simulation properties respectively (figure 3)

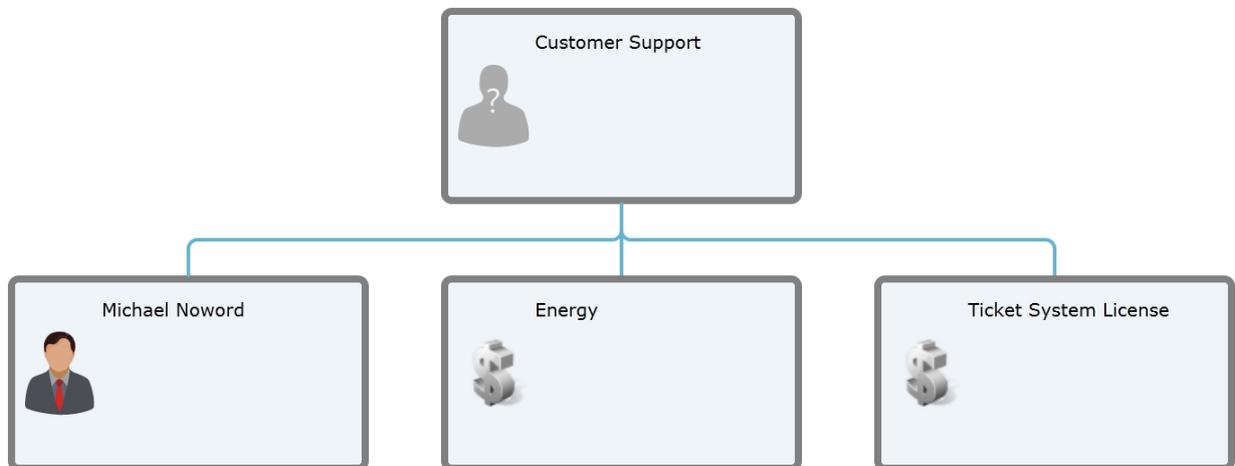


Figure 2: Organizational Diagram [YOPRO link](#)

Properties	
Property set	Simulation
General Properties	
Element Type	Human
Id	Human
Name	Cost Center
	Product Group
Note	
Attribute Extensions	<a href="#">Edit</a>
Visualization	
Color	Automatic
Optional Simulation Properties	
Costs	2400
Has Basic Load	<input type="checkbox"/>
Payment	Fixed salary
Post-processing Time	not selected <a href="#">...</a>
Setup Time	not selected <a href="#">...</a>
Time Unit	Month
Unit	Dollar

Figure 3: Simulation properties

### Defining payments and working times of employees

- Mark Michael Noword
- Open the simulation properties (figure 3)
  - Assign a fixed salary of 2400 Dollars per month to Michael Noword (figure 4)

Optional Simulation Properties	
Costs	2400
Has Basic Load	<input type="checkbox"/>
Payment	Fixed salary
Post-processing Time	not selected <a href="#">...</a>
Setup Time	not selected <a href="#">...</a>
Time Unit	Month
Unit	Dollar

Figure 4: Optional Simulation Properties

- Open the shift editor (figure 5)
  - Select the 40h week with lunch breaks from the drop down list
- Switch to the organization tab on the ribbon menu
- Create a new role called 'Customer Support' as already explained in the context of [organizational diagrams](#).
- Assign the role to Michael Noword

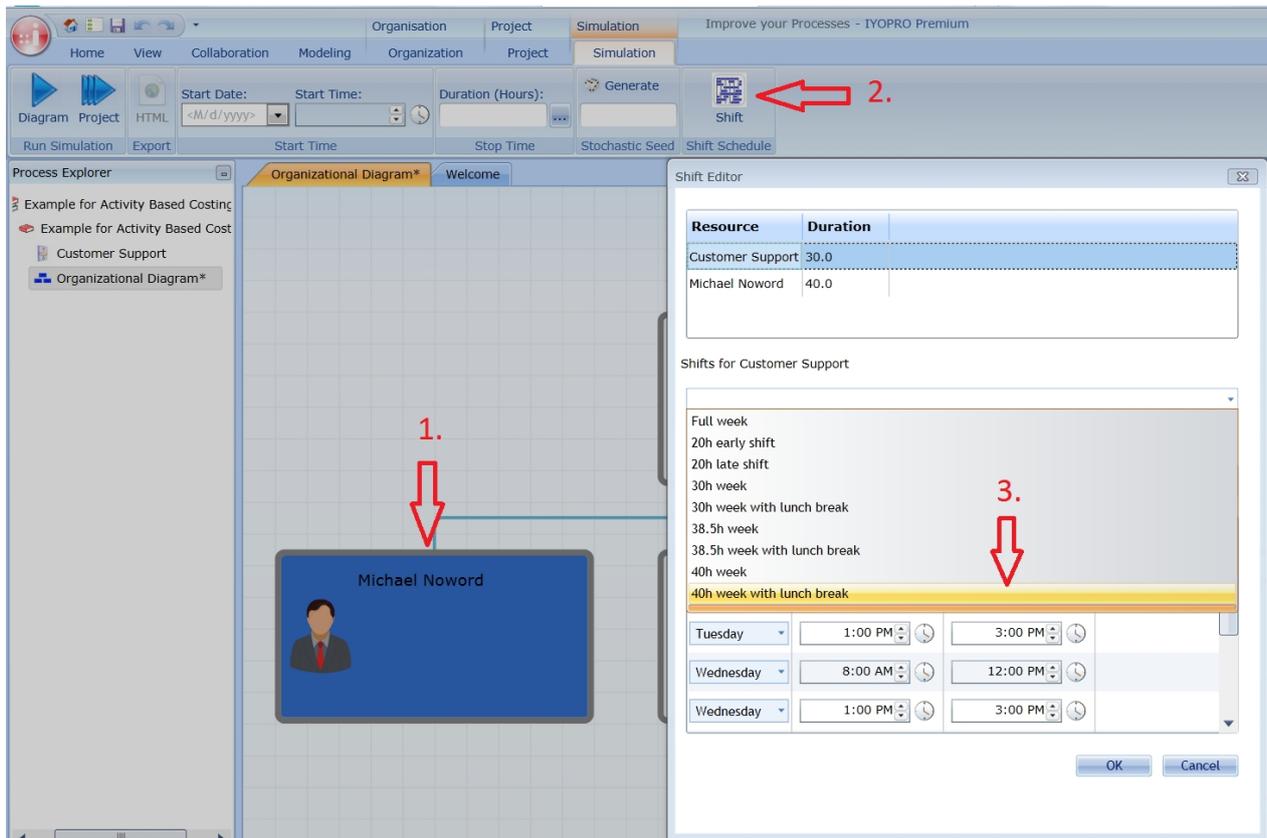


Figure 5: Shift Editor

### Distributing cost factors

- Switch back to the collaboration diagram (Customer Support)
  - Mark the task 'enlist customer details' and open the advanced properties editor
  - Add the role Customer Support as participant (figure 6)
  - Repeat the steps for activity 'send product information'

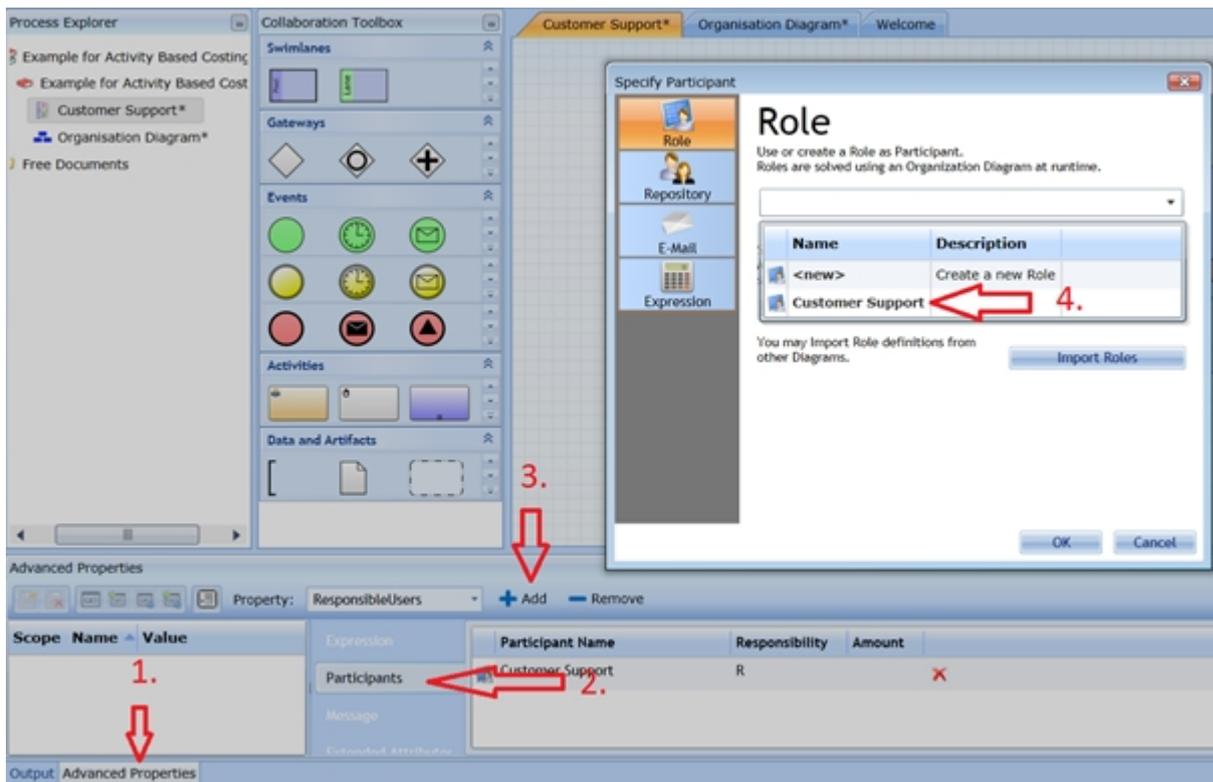


Figure 6: Adding participants

- Distribute the cost centers (figure 7)
  - Mark the activity 'enlist customer details' again and open the simulation properties editor
  - Open the area of Activity Based Costing and select energy
  - Set it to time dependent and type in the value 0.5 kWh
  - Repeat the steps for activity 'send product information'

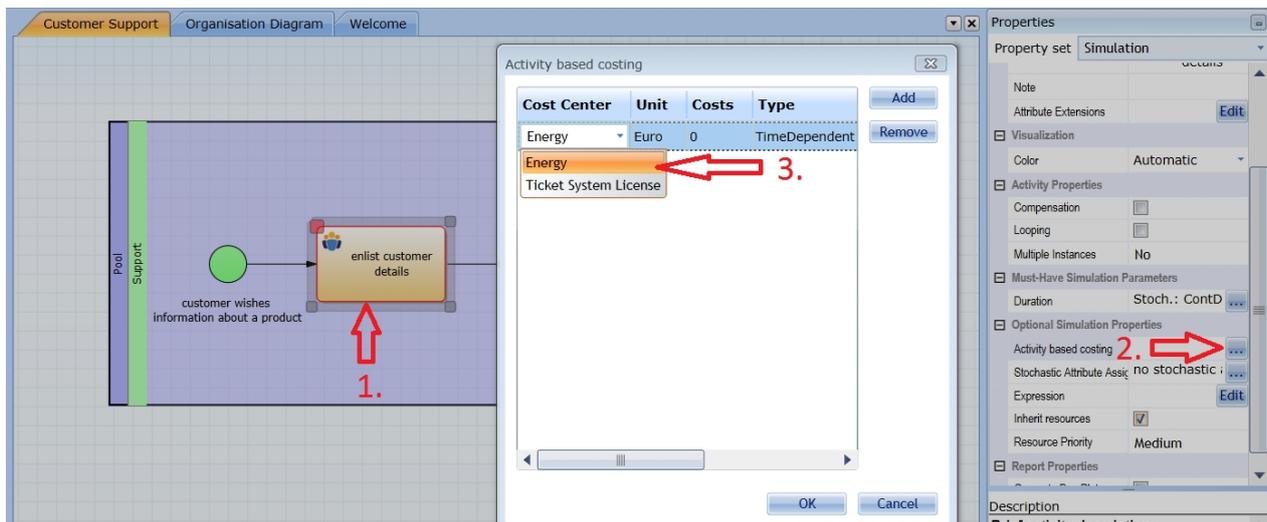


Figure 7: Distributing Cost Centers  
Adding duration and inter-arrival time

How to set [durations and inter-arrival time](#) has already been explained in a previous tutorial. Set the properties described as follows:

- Set the inter-arrival time to 5 minutes at the start event
- Set duration of 'enlisting customer details' as normally distributed with 4 minutes and 20 seconds on average and with a standard deviation of 1 minute and 20 seconds

- Set the duration of 'send product information' as a constant value of 1 minute and 20 seconds

## Explanation

First of all we modeled the process and the organizational set-up. Then we filled the organizational resources with the required information. Humans need salary assigned to cost factor as well as a working plan since they aren't available 24h per day. An employee can have different roles in different processes. Thus we had to assign Michael Noword to the role 'Customer Support' in context of our process. This role was attached to the tasks to point out who executes the tasks. Right after that we assigned the energy costs to our tasks. Tasks spend energy because the participant works with a personal computer. The more time is spent to finish his tasks the more energy is needed. So the consumption is time dependent. The cost center ticket system license doesn't have to be assigned to tasks because it isn't influenced by activity durations and the number of process cycles. These costs will be distributed over the number of process cycles in the simulation. In the end we set our activity durations and inter-arrival time to denote how much time our activities take and how often the process is started. These properties are also needed for our time dependent energy costs.

## Report

See below for the run configuration, which we set up to receive the report statistics from below.

Please refer to the section [Experiment Results](#) for more information on the report specific data.

## Run configuration

- Stochastic Seed: 12345
- Simulation start date: 1/1/2015 12:00:00 a.m.
- Simulation stop time: 24
- Simulation stop time unit: Hours

## Report

- Start the simulation by clicking the simulation button on the simulation tab of the ribbon menu (figure 1)
- Open simulation report and head for the costing details (figure 2)

The fixed salaries show the incurred costs by Michael Noword for this process. We also have the ticket system license displayed. 161,29 Dollar are the costs of the ticket system license for 1 day according to our simulation time. The costs of the ticket system license are 208.3% of our quantity-induced activity costs. If you simulate the process by using a longer simulation time the percentage of the ticket system costs becomes lower while the number of process cycles increases. This effect is called economies of scale. We can see the energy costs too. They employ a unit measurement so the results are displayed in a separate table. Further cost centers employing the same unit would be displayed in the same table.

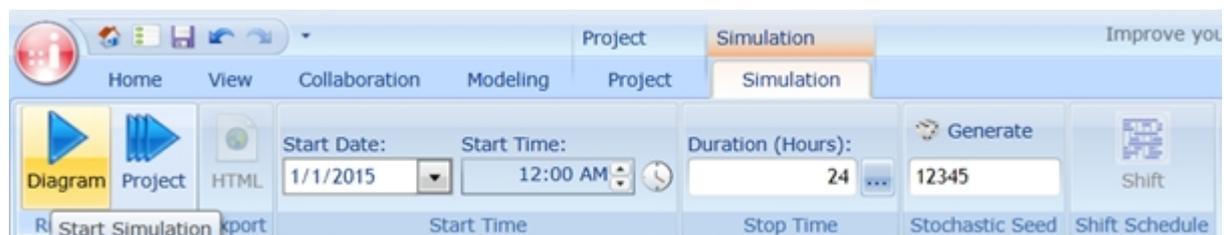


Figure 1: Simulating a diagram

**Process Costs Dollar**

	<b>Pool: Customer Support#1</b>	<b>Sum</b>	
Fixed salaries (activity based ratio)	77.42	77.42	
Total neutral costs (based on utilization)	77.42	77.42	
Total activity costs	77.42	77.42	
Ticket System License (208.3%)	161.29	161.29	
Total costs	238.71	238.71	

**Process Costs kWh**

	<b>Pool: Customer Support#1</b>	<b>Sum</b>	
Energy	8.04	8.04	
Total deduced costs	8.04	8.04	

Figure 2: Costing Details

# Business Process Automation

---

## Introduction

With IYOPRO it is very easy to automate business processes. Business Process Automation means, that you do not stop after you created a documentation of your process landscape.

The power of IYOPRO is to breathe life into your diagrams and let them do things for you. The result is called Workflow.

## Conceptual Overview

A workflow is usually build out of the following components:

- One or more [process- or collaboration](#) diagrams  
These diagrams usually define tasks which are executed in the drawn flow
- One or more [forms](#)  
Forms are used for human tasks, which have to be fulfilled by users, to provide input and output data.
- One or mode [schema diagrams](#)  
Schemas (ie. Databases) are required to permanently store process information for later use, even if the process instance has been finished.
- An [organization diagram](#)  
Organization diagrams bridge between human task definitions and real persons, which have to perform these tasks.

All these elements are defined within a [BPI-Project](#). IYOPRO generates a workflow out of these definitions with the "Build" Command

## Define Workflows

To define a workflow it is necessary to complete the following steps:

1. Open or create a [BPI-Project](#).  
It is useful to create the project in a [team folder](#) rather than in your private workspace to avoid later security issues.
2. Create or open all required [collaboration diagrams](#). Ensure that the diagram property "Executable" is set for all diagrams, which define the workflow. Disable the flag if the diagram is for documentation purposes only.
3. Ensure for all [pools](#) in your executable diagrams that the property "Executable" is set to "Full". Pools which do not have this setting are for documentation purposes only and describe collaboration partners which are not automated.
4. Keep in mind, that the [BPMN specification](#) creates a separate process instance for each executable pool. Communication is done by message flows. IYOPRO follows this specification. If you expect only one process instance, either draw your diagram in one pool or remove all pools.
5. Do not forget to assign a [Form](#) to all BPMN User Tasks. You'll receive an [error](#) on the [build](#) pass for each user task which misses an assigned form. If a specified task does not require a form, convert it to a manual task.
6. If you wish to start the workflow based on the submission of a form, [declare the form as workflow startup form](#). You'll find the option in the context menu of the form in the [process explorer](#).
7. Remember to declare the scope of [process instance variables](#) on each element as needed. This are of course script tasks, but also conditions on sequence flows and more.

8. Assign a process diagram of your project as [preview of your workflow](#) definition. The option can be found in the context menu of the diagram in the [process explorer](#).
9. Please fill in a workflow description to allow users to understand what the workflow is for. It's rich text so you may format the description and highlight important information. The description can be edited on the [workflow property page](#).
10. [Build the workflow](#). It will only be executable if the build pass was free of errors. Verify, that the build pass has generated the workflow executable in the "bin" directory.

You are now able to run and monitor your workflow. To do so please follow [this](#) guide.

## Process Instance Variables

One of the most important mechanisms in process automation is the use of process instance variables, also called attributes. This mechanism makes automation flexible and able to connect to other devices by web services or handle form data.

Process instance variables consist out of a unique name and a value.

### Variable Name

A variable name is a single word which starts with an alphanumeric Latin character (A-Z or a-z). It may contain the Latin letters A-Z or a-z and the digits 0-9. Other characters are not allowed.

Valid variable names are

- MyVar
- V1
- X

Invalid variable names are

- My Var
- My\_Var
- 0X

### Variable Value

The value of a variable may contain the following types of values:

- Integer
- Float / Double
- String
- DateTime
- Guid
- List<Object>
- Dictionary<String, Object>

Other types may work but must be serializable to store them in the global scope. The Objects in the List and Dictionary type may be out of these types too.

## Global Scope

The global scope specifies all process instance variables, which may be accessed by an operation. Each operation must specify, if and how variables out of the global scope are used or not. The following modes are

available:

- None  
The variable is not used
- Import  
The variable is made available in the operation but modifications of the variable value are not written back to the global scope.
- Export  
The global scope will create or update the variable value at the end of the operation.
- Import + Export  
The variable is made available in the operation and modifications are written back to the global scope at the end of the operation

The scope of each operation for the required variables may be edited in the [Advanced Properties Window](#)

There are also a number of predefined variables which are described [here](#)

### Best Practice

Depending on the purpose of variables in workflows their usage and availability during the life time of a process instance differ.

- Global Scope  
Variables with a global scope can be accessed in every state of a process instance. This results in a demand of performance. To avoid time delays and performance issues only those variables should have a global focus that are required during the whole life span of a process instance like form variables or atleast in many operations.
- Local Scope  
Those variables are generated in expressions of tasks they are used in. Especially operations with high mounds of data are supposed to be used with local scope. Common examples are content of data grids and displayed values of drop down lists with workflow attributes as source. Furthermore results of database queries are supposed to be handled this way via [schema diagram](#) and [query editor](#).

### Predefined Objects

When performing script tasks or actions, it's often necessary to access information about the workflow environment. IYOPRO provides the following predefined objects which provide objects to these information:

### Session

The Session object provides general information about the current Session. These are:

### Syntax

```
public class Session
```

### Members

 <code>string</code> GetAdapterPath( <code>string</code> adapterName)	This method returns the path of an installed adapter
 <code>User</code> CurrentUser	The current user. Only available during processing of a user form (action button and end action) and process instance creation, if invoked by a user. Is not available during backend workflow processing, as there is no specific user associated.
 <code>XMLMessageWrapper</code> XMLMessage( <code>string</code> XmlMsgGuid)	Retrieves the XML message

## Sample

This sample loads an Adapter to the PDM-System OpenEDM and performs a function call

```
import clr
clr.AddReferenceToFileAndPath(Session.GetAdapterPath
("IYOPRO.OPENEDM"))
from OpenEDMAdapter import *
a = Adapter()
OEResult = a.OECallFunction("", "", "DEMO.FUNCTION", "MyData")
```

## Workflow

The Workflow object provides general information about the current Workflow. These are:

## Syntax

```
public class Workflow
```

## Members

 <b>int</b> Id	Id of the workflow
 <b>int</b> FileId	Id of the workflow file
 <b>int</b> ProjectId	Id of the workflow project
 <b>Users</b> Users	A list of all users that have read privilege at the workflow and may become participant
 <b>Groups</b> Groups	A list of groups that are assigned to the team the workflow is associated to

## Sample

This sample reads the email address of the user with the name "Smith".

```
mailto = Workflow.Users.GetValue("Smith").Email
```

also possible:

```
mailto = Workflow.Users["Smith"].Email
```

## UserTask

The UserTask object provides general information about the current User task.

The UserTask object can be accessed using the RegisterCallback function of the ProcessInstance object.

## Syntax

```
public class UserTask
```

Access via the RegisterCallback function of the ProcessInstance object:

## Members

 <b>Guid</b> GuId	The unique identifier of the task
 <b>int</b> FormId	The Id of the form that is assigned to the task
 <b>ReadOnlyCollection&lt;User&gt;</b> Participants	The participants in the user task

 <code>UserTaskPriority</code> <code>InitialTaskPriority</code>	No usable result
--	------------------

Direct access via the UserTask object:

## Members

 <code>Guid</code> <code>GuId</code>	No usable result
 <code>int</code> <code>FormId</code>	No usable result
 <code>ReadOnlyCollection&lt;User&gt;</code> <code>Participants</code>	No usable result
 <code>UserTaskPriority</code> <code>InitialTaskPriority</code>	The priority the task has when generated. Default: None

## Syntax

```
public enum UserTaskPriority
```

## Members

 <code>None = -1</code>	The user task has no priority
 <code>VeryLow = -2</code>	The user task has a very low priority
 <code>Low = -3</code>	The user task has a low priority
 <code>Medium = -4</code>	The user task has a medium priority
 <code>High = -5</code>	The user task has a high priority
 <code>VeryHigh = -6</code>	The user task has a very high priority

## Sample

This sample sets the initial task priority to low.

```
UserTask.InitialTaskPriority = UserTask.Priority.Low
```

## Token

The Token object provides general information about a workflow token. These are:

## Syntax

```
public class Token
```

## Members

 <code>ProcessInstance</code> <code>Instance</code>	A reference to the current process instance
 <code>Diagram</code> <code>Diagram</code>	A Reference to the Diagram, the token belongs to
 <code>DiagramControl</code> <code>Control</code>	A Reference to the Diagram Control, the token is attached to
 <code>TokenAttribute</code> <code>Attributes</code>	A list of attributes assigned to the current token
 <code>string</code> <code>Details</code>	Gets or sets token details which are displayed in the overview of the tasks (if no details have been defined, the process instance details will be displayed instead)
 <code>object</code> <code>GetAttribute(string name)</code>	Retrieves an attribute from the token attributes
 <code>void</code> <code>SetAttribute(string name, object value)</code>	Stores an attribute in the token attributes
 <code>void</code> <code>Suspend(object until)</code>	Enables the use of cooperative multitasking by suspending the execution of the token until the given timestamp is reached (Python datetime or C# DateTime)
 <code>Users</code> <code>ResolveRole(string rolename)</code>	Resolves a role. It returns a list of users. The rolename may be a plain role name or a role

	keyword.
--	----------

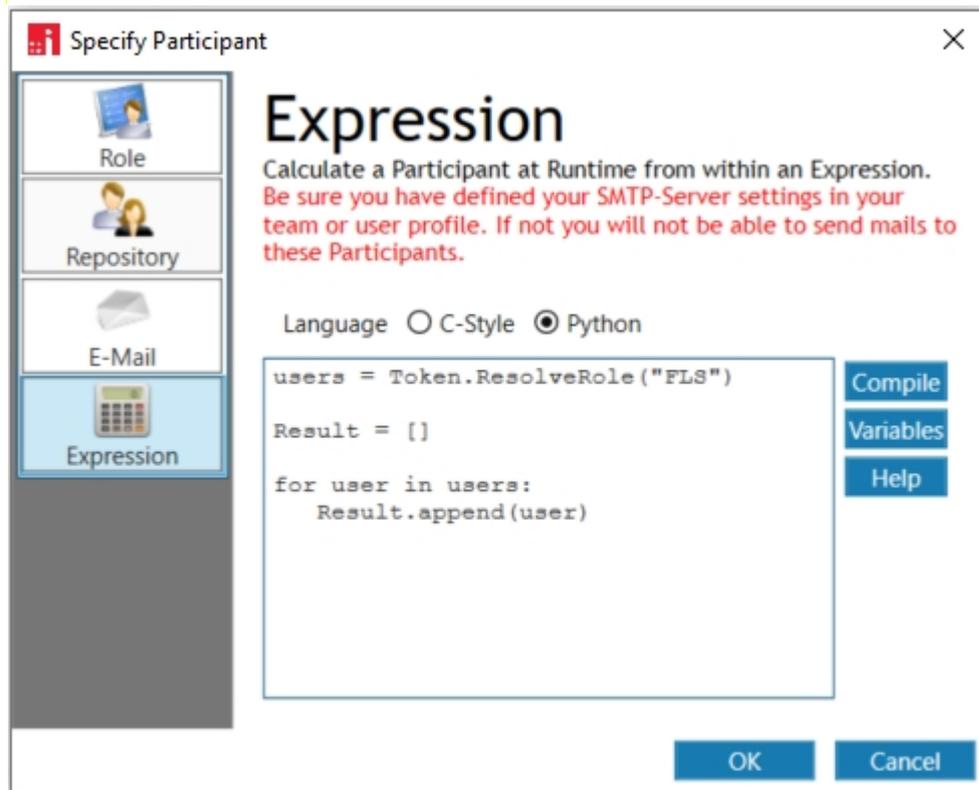
## Sample

This sample sets an information as token details.

```
Token.Details = "waiting for approval"
```

This sample uses ResolveRole to determine and set the participants of a task.

The role has been defined in an organization diagram and assigned to users there. By using ResolveRole, the organization diagram is searched for the users with the specified role, in this case the keyword.



## TokenAttribute

The TokenAttribute object provides general information about a workflow token attribute. These are:

## Syntax

```
public class TokenAttribute
```

## Members

 <b>int</b> Count	The number of records in the list
 <b>object</b> GetAttribute( <b>string</b> name)	Get the record of the given attribute name

## Sample

This sample retrieves a list of names.

```
nameList = Token.Attributes["names"]
```

## ProcessInstance

The ProcessInstance object provides general information about the current Process Instance. These are:

### Syntax

```
public class ProcessInstance
```

### Members

 <code>int</code> Id	The Id of the process instance
 <code>Guid</code> GuId	The Guld of the process instance
 <code>string</code> Name	The Name of the process instance
 <code>DateTime</code> CreatedAt	The time stamp, the process instance has been created
 <code>User</code> Created	This Attribute contains all attributes of the initiator of the process instance. If the process instance has been created anonymously, this attribute will be null.
 <code>Roles</code> Roles	A list of roles that have been defined in the process context
 <code>Events</code> Events	A list of events in the process instance context
 <code>string</code> Details	Gets or sets process instance details that are displayed in the overview of running instances
 <code>byte[]</code> ReadFileBytes( <code>string</code> filename)	Read the bytes of a process instance file into a byte array
 <code>Stream</code> ReadFileToStream( <code>string</code> filename)	Read a process instance file into a stream
 <code>void</code> WriteFileBytes( <code>string</code> filename, <code>byte[]</code> bytes)	Write the bytes of a byte array to a process instance file
 <code>void</code> WriteFileFromMemoryStream( <code>string</code> filename, <code>System.IO.MemoryStream</code> mem)	Write the content of a memory stream to a process instance file
 <code>void</code> GetFileUrl( <code>string</code> filename)	Get the URL required to download a process instance file
 <code>void</code> Warn( <code>string</code> message)	Append a warning into the process instance log
 <code>void</code> Info( <code>string</code> message)	Append an information to the process instance log
 <code>void</code> Error( <code>string</code> message)	Append an error to the process instance log CAUTION: This will force the process instance to halt in error state
 <code>void</code> RegisterCallback ( <code>ProcessInstanceCallbackType</code> callbackType, <code>string</code> script, <code>string</code> method)	Register a callback on certain event types.
 <code>object</code> GetWorkflowAttribute( <code>string</code> name)	Retrieves a workflow attribute from the instance even if it has not been imported
 <code>void</code> SetWorkflowAttribute( <code>string</code> name, <code>object</code> value)	Stores a workflow attribute in the instance even if it has not been imported
 <code>string</code> GetFileUrl( <code>string</code> filename)	Retrieve the URL to a file
 <code>string</code> GetRepositoryFileUrl( <code>string</code> relativefilename)	Retrieves the repository file url of the given relative file name. The path must be relative based on the workflow file
 <code>void</code> TriggerRolesChanged ( <code>IEnumerable&lt;string&gt;</code> rolenames)	Method triggers the roles changed event
 <code>DataSet</code> GetRootDataSet()	Get the record of the root dataset
 <code>DataSet</code> GetCurrentDataSet()	Get the current dataset

## Syntax

```
public enum ProcessInstanceCallbackType
```

## Members

 None = 0	Unspecified
 UserTaskStarted = 1	Indicates that a user task has been started
 UserTaskFinished = 2	Indicates that a user task has been finished
 UserTaskAccepted = 3	Indicates that a user task has been accepted
 ProcessInstanceFinished = 4	Indicates that the process instance has been finished
 RolesChanged = 5	Indicates that the roles of the process instance have been modified
 TaskFinished = 6	Indicates that a task has been finished
 SubProcessStarted = 7	Indicates that the sub process has been started
 ProcessInstancePreDelete = 8	Indicates that the process instance is about to be deleted

## Method parameters

The method for the callback must have a parameter.

This parameter represents an array that enables access to objects depending on the callback type.

Callback type	Parameter array
UserTaskStarted	[Token, UserTask]
UserTaskFinished	[Token, UserTask, User]
UserTaskAccepted	[UserTask, User]
ProcessInstanceFinished	[ProcessInstance]
RolesChanged	[ProcessInstance, string modifiedRoleName]
TaskFinished	[Token]
SubProcessStarted	[Token, ProcessInstance]
ProcessInstancePreDelete	[ProcessInstance, Workflow, User]

## Sample

This sample retrieves the email address of the process instance initiator

```
mailto = ProcessInstance.Created.Email
```

This sample modifies the content of a word document. The variable myfile points to a file which may either be uploaded from a web form or which could be retrieved out of a data store.

```
import clr
clr.AddReferenceToFileAndPath(Session.GetAdapterPath("IYOPRO.WORD"))
from IYOPRO.WORD import *
data = ProcessInstance.ReadFileBytes(myfile)
with Adapter() as word:
```

```
word.SetDocumentBytes(data)
word.ReplaceBookmark('MyBookMark','Hello World')
stream = word.GetDocumentStream()
ProcessInstance.WriteFileFromMemoryStream(myfile,stream)
```

This sample registers a callback every time, a user has accepted a user task.

```
ProcessInstance.RegisterCallback
(ProcessInstanceCallbackType.UserTaskAccepted,"MyBPIScript.py","UserAccepted")
```

The method that you implement for the callback must have a parameter. This parameter represents an array that enables access to [UserTask](#) and [User](#) (for `UserAccepted(args)` it would be `args[0]` for `UserTask` and `args[1]` for `User`).

You could use the method, for example, to record which task was accepted by whom. You could also find out for which tasks the person working on them is constantly changing.

Likewise, you could protocol with `UserTaskFinished` which tasks were finished by whom.

Or `ProcessInstancePreDelete` could be used so that if someone tries to delete a process instance, it is checked in advance whether certain requirements are met before this process instance is actually deleted.

## User

The User object provides general information about a registered IYOPRO User. These are:

### Syntax

```
public class User
```

### Members

 <code>int</code> Id	The unique Id of the user
 <code>string</code> Name	The name of the user
 <code>string</code> Email	The email address of the user
 <code>string</code> Account	The account of the user
 <code>string</code> Language	The language set when the user was created
 <code>string</code> SecurityId	The directory services security id of an user, if the user is synchronized with the directory service. If not an empty string.
 <code>void</code> SetAbsence( <code>bool</code> gone, <code>object</code> fromDate, <code>object</code> untilDate, <code>string</code> reason = "", <code>object</code> creator = null)	Specify if the given user is absent (gone = true) or present (gone = false). fromDate and untilDate can be either C# DateTime objects or Python datetime's. However currently only the date part is used to define absence or presence of the user. The parameter creator is either another User object (e.g. user who started the process instance) or a string.

### Sample

This sample retrieves the email address of the process instance initiator

```
user = ProcessInstance.Created
mailto = user.Email
```

## Users

The Users object groups objects of type RepositoryUserRecord together

### Syntax

```
public class Users
```

### Members

 <code>int</code> Count	The number of user records in the list
 <code>User</code> GetValue( <code>string</code> username)	Get the record of the given User

### Sample

This sample reads the email address of the user with the (account)name "Smith".

```
mailto = Workflow.Users.GetValue("Smith").Email
```

also possible:

```
mailto = Workflow.Users["Smith"].Email
```

## Role

The Role object provides general information about a role in a process instance. These are:

### Syntax

```
public class Role
```

### Members

 <code>string</code> Name	The name of the role
 <code>ReadOnlyCollection&lt;User&gt;</code> Users	The users in the role
 <code>void</code> Add( <code>User</code> user)	Add a user to the role
 <code>void</code> Remove( <code>User</code> user)	Remove a user from the role
 <code>void</code> ClearUsers()	Remove all users from the Role

### Sample

This sample prints the user names of all roles in the process instance

```
for role in ProcessInstance.Roles:  
    for user in role.Users:  
        print user.Name
```

## Roles

The Roles object groups objects of type Role together

### Syntax

```
public class Roles
```

## Members

 <code>int</code> Count	The number of records in the list
 <code>Role</code> GetRole( <code>string</code> name)	Get the record of the given role name
 <code>Role</code> Create( <code>string</code> rolename)	Create a new role. If the role exists, the existing role will be returned
 <code>void</code> Delete( <code>string</code> rolename)	Delete a role.

## Sample

This sample reads the roles of a process instance

```
for role in ProcessInstance.Roles:  
    print role.Name
```

This sample creates a role

```
ProcessInstance.Roles.Create("Manager")
```

## Group

The Group object provides general information about an IYOPRO Usergroup. These are:

## Syntax

```
public class Group
```

## Members

 <code>string</code> Name	The name of the group
 <code>ReadOnlyCollection&lt;User&gt;</code> Users	The users in the group

## Sample

This sample shows all usergroups in the workflow team

```
for group in Workflow.Groups:  
    print group.Name
```

This sample shows all user in the group 'Administrators'

```
group = Workflow.Groups.GetGroup("Administrators")  
for user in group.Users:  
    print user.Name
```

## Groups

The Groups object groups objects of type Group together

## Syntax

```
public class Groups
```

## Members

 <code>int</code> Count	The number of records in the list
 <code>Group</code> GetGroup( <code>string</code> name)	Get the record of the given group name

## Sample

This sample retrieves the administrators group.

```
admingroup = Workflow.Groups.GetGroup("Administrators")
```

also possible:

```
admingroup = Workflow.Groups["Administrators"]
```

```
print admingroup.Name
```

## DocumentationPackage

The DocumentationPackage object contains all information about a documentation package which has to be released

## Syntax

```
public class DocumentationPackage
```

## Members

 <code>int</code> Id	The id of the package
 <code>string</code> Name	The name of the package
 <code>string</code> LinkUrl	An URL to open the package in IYOPRO
 <code>int</code> ReviewerGroupId	The id of the reviewer group
 <code>string</code> ReviewerGroup	The name of the reviewer group
 <code>DateTime?</code> ValidFrom	The valid from time stamp of the package
 <code>DateTime?</code> ValidUntil	The valid until time stamp of the package
 <code>bool</code> Release()	Method to release the documentation package
 <code>bool</code> Delete()	Method to delete the documentation package

## Sample

[This sample](#) creates a sample release workflow.

## LibraryElement

The LibraryElement object contains all information about a release package for library elements

## Syntax

```
public class LibraryElement
```

## Members

 <code>int</code> Id	The id of the package
 <code>string</code> Name	The name of the package
 <code>Guid</code> GuId	The library element's unique id
 <code>int</code> ReviewerGroupId	The id of the reviewer group
 <code>string</code> ReviewerGroup	The name of the reviewer group
 <code>DateTime?</code> ValidFrom	The valid from time stamp of the package
 <code>DateTime?</code> ValidUntil	The valid until time stamp of the package
 <code>bool</code> Release()	Method to release the documentation package
 <code>bool</code> Delete()	Method to delete the documentation package

## Diagram

The Diagram object contains all public information about a diagram

## Syntax

```
public class Diagram
```

## Members

 <code>int</code> Id	The id of the diagram
---	-----------------------

## DiagramControl

The DiagramControl object contains all public information about a diagramcontrol

## Syntax

```
public class DiagramControl
```

## Members

 <code>int</code> Id	The id of the control
 <code>string</code> Name	The name of the control
 <code>string</code> TypeName	The type name of the control

## Database

The Database object provides general information about Databases in the process instance. These are:

## Syntax

```
public class Database
```

## Members

 Schema GetSchema( <code>string</code> name)	Retrieves a schema by name of the associated <a href="#">schema diagram</a> .
---	---

## Syntax

```
public class Schema
```

## Members

 <code>object</code> Execute( <code>string</code> cmd)	Execute an SQL-statement on the schema
 <code>DBReader</code> Select( <code>string</code> cmd)	Execute an SQL-statement on the schema which returns data
 <code>void</code> Close()	Close the schema

## Syntax

```
public class DBReader
```

## Members

 <code>bool</code> IsClosed()	Indicates if the reader has been closed
 <code>bool</code> Read()	Read the next row from the database
 <code>void</code> Close()	Close the reader
 <code>void</code> Dispose()	Disposes the reader. Its important in non-sandboxed environments to dispose the reader to avoid memory leaks
 <code>object</code> GetValue( <code>int</code> index)	Get the value of the column of the current row by index

## Sample

This sample sets and updates information about a person

```
s = Database.GetSchema("Datastorage")
hasData = False
with s.Select("select Id, Name, Age from Person") as reader:
    while reader.Read():
        hasData = True
        ProcessInstance.Warn("DBData:"+str(reader.GetValue(1)))
        mId = reader.GetValue(0)
        mAge = reader.GetValue(2)
if hasData:
    s.Execute("UPDATE Person SET Age=%d WHERE ID=%d" % (mAge + 1, mId))
else:
    s.Execute("INSERT INTO Person (Name, Age) VALUES ('Frank', 45)")
s.Close()
```

## ScriptContext

ScriptContext is used to define case distinctions for workflow expressions depending on the given workflow setting, point of execution and user actions

## Members

	Undefined	Value that is always active when no other value is chosen
	InitInstance	Value that is active when the diagram / pool expression is executed. If there is a start form, it will be active after the start form has been submitted
	PrepareForm	Value that is active when the diagram / pool expression is executed and there is a startup form. It is active before the start form is displayed
	SaveForm	Value that is active when the end expression of a workflow form is executed after pressing the save button
	SubmitForm	Value that is active when the end expression of a workflow form is executed after pressing the submit button
	ActivityExpression	Value that is active when the activity expression is executed
	MultiInstanceExpression	Value that is active when the activity expression is executed for each instance of a multi-instance activity

## Samples

This sample shows differing workflow behaviour depending on the question if a start form is used or not.

If you have a start form, you could use this to initialize form data:

```
if ScriptContext.ActiveContext == ScriptContext.PrepareForm:
    # Prepare Form Data - no Process Instance available
    SFORM = { "User" : "John Doe", "Department" : "public
relations" }
```

If there is no start form or after the start form has been submitted, you could use this to define data at the beginning of the instance:

```
if ScriptContext.ActiveContext == ScriptContext.InitInstance:
    import System
    date = System.DateTime.UtcNow
```

This sample shows a possibility to save the user who submitted the form into a variable for further use in the process. The sample is placed in the end action of the form.

```
if ScriptContext.ActiveContext == ScriptContext.SubmitForm:
    finishedBy = Session.CurrentUser.Account
```

A workflow expression of a user task can be executed multiple times, e.g. with a parallel user task. ScriptContext can be used to find out whether the execution is the initial token (ActivityExpression) or one of the multiple (MultiInstanceExpression). That way you can differentiate, e.g. which data should only be prepared once initially and which data should only be used in the parallel executions.

## ExecutionContext

ExecutionContext provides information about the context in which expressions of a workflow are executed

## Members

	ScriptContext ActiveContext	The currently active expression context during workflow execution
---	-----------------------------	---

## Application

The Application object allows retrieval of custom attributes.

In the backend configuration of IYOPRO, system-specific attributes can be stored in the form of key value pairs. These values can be retrieved through the configuration value.

## Syntax

```
public class Application
```

## Members

 <code>string</code> GetConfigValue( <code>string</code> key, <code>string</code> defaultvalue)	Get the configuration value
--	-----------------------------

## Sample

This sample retrieves a configuration value

```
sapsystem = Application.GetConfigValue("MySAPSystem", None)
```

## DataSet

The DataSet object provides general information about a dataset in a process instance. These are:

## Syntax

```
public class DataSet
```

## Members

 <code>Guid</code> Id	The unique identifier of the dataset
 <code>Guid</code> ParentId	The unique identifier of the parent of the dataset
 <code>string</code> Name	The name of the dataset
 <code>string</code> Keyword	The keyword of the dataset
 <code>int</code> Index	The index of the dataset
 <code>DataSets</code> GetSubDataSets()	Retrieves the datasets of the subprocesses
 <code>object</code> GetAttribute( <code>string</code> name)	Retrieves an attribute from the dataset attributes
 <code>void</code> SetAttribute( <code>string</code> name, <code>object</code> value)	Stores an attribute in the dataset attributes

## Sample

This sample provides the information about the contents of the dataset of each started parallel subprocess.

```
Root = ProcessInstance.GetRootDataSet()  
SubDataSets = Root.GetSubDataSets()  
  
for ds in SubDataSets:  
    ProcessInstance.Info(ds.Name + " " + ds.Keyword + " " + unicode  
(ds.Index))  
    value = ds.GetAttribute("MyTest")  
    ProcessInstance.Info(value)
```

## DataSets

The DataSets object groups objects of type DataSet together

## Syntax

```
public class DataSets
```

## Members

 <code>int</code> Count	The number of records in the list
--	-----------------------------------

 DataSet GetValue(Guid guid)	Get the record of the given Guid
---	----------------------------------

## Define Participants

Participants must be defined at least for all [user tasks](#). Participants can be defined with the [participants editor](#).

## Declare Start Form

The start form is a form which must be filled out by a user. With the submission a new process instance will be created.

## Declare Startup Form

To declare the start form open the context menu of the form in the process explorer. Then hit the "Set as startup form" option and rebuild your workflow.

## Remove Startup Form

The Startup Form setting is stored in the workflow properties of the project. Activate the project in the process explorer and clear the property "Startup Form". Then rebuild your workflow.

Workflow Properties	
Workflow Description	This process is used to monitor the temperature of a heating enging. If the temperature exceeds 35 degree a warning mail will be send.
Workflow Name	Temperature Monit
Preview From	Collaboration Diagr
Specify Name	<input checked="" type="checkbox"/>
Startup Form	

## Providing Form Data

### Providing data for forms

To display workflow attributes in a form, the data has be available to the form. There are two modes to link the form elements to the workflow attributes.

If form itself has an empty property "Variable" the simple mode is used. If the property contains a variable name the standard mode is used.

### Simple mode

For each form element which allows user input, the property "Variable" is inspected. If there is a workflow attribute with the same name, it is used to provide the data for the form element. Likewise changes performed in the form are stored in the workflow attribute.

### Standard mode

If the property "Variable" is specified for the form, it is interpreted as a python dictionary. The dictionary keys are the variable names of the individual form elements.

The data for the elements is provided by the value of the corresponding dictionary entry. Changes to the data are stored in the dictionary as well.

In addition any other entry added to the dictionary is available to the form as well. This feature can be used to control addition form element properties like the visibility or if the element is disabled or not.

The standard mode is the preferred way to exchange data between the workflow and the forms.

## Declare Workflow Preview

The workflow preview is a small image, that acts as graphical representation of the workflow in certain circumstances.

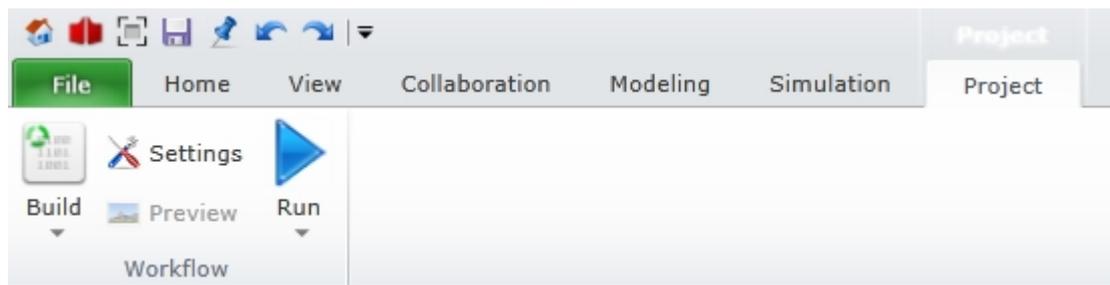
In IYOPRO the workflow preview is build out of one diagram of the [BPI-Project](#).

There are two ways available to define the preview:

- Use the Preview Button in the Project Ribbon Tab
- Use the "Set as Workflow Preview" option in the diagrams context menu of the process explorer

## Project Ribbon Tab

To declare the content of a diagram as workflow preview ensure, that the desired diagram is the active document. Navigate to the Project Ribbon Tab. If the active diagram is suitable the "Preview" button of the Workflow Group is enabled and may be pressed. Then rebuild your workflow.



## Process Explorer

To declare the content of a diagram as workflow preview within the [process explorer](#) hit on the "Set as Workflow Preview" option of the diagrams context menu. If the option is missing, the diagram is not suitable. Then rebuild your workflow.

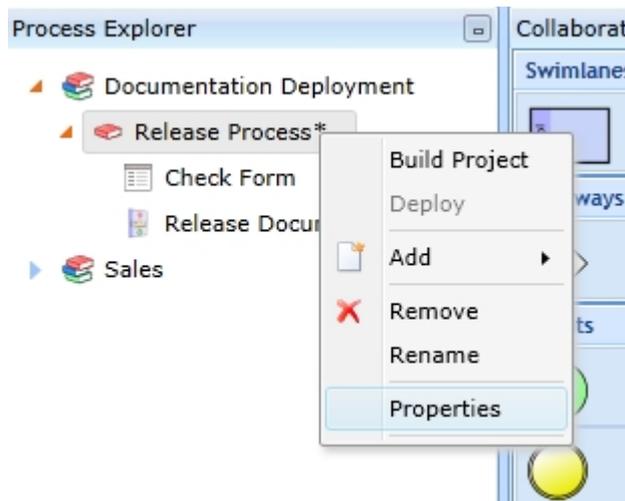
## Remove Preview

The preview setting is stored in the workflow properties of the project. Activate the project in the process explorer and clear the property "Preview From". Then rebuild your workflow.

Workflow Properties	
Workflow Description	This process is used to monitor the temperature of a heating enging. If the temperature exceeds 35 degree a warning mail will be send.
Workflow Name	Temperature Monit
Preview From	Collaboration Diagr
Specify Name	<input checked="" type="checkbox"/>
Startup Form	

## Workflow Property Page

The workflow property page is used to control settings which affect the workflow build pass. To open the workflow property page right click on the BPI-Project which describes the workflow and choose Properties.



Workflow Build Options

Specify Name

Workflow Name

Description

Techn. Admin

Test User

API User (not set)

Tags

## Name

To specify a name for the workflow that differs from the project name, toggle the "Specify Name" checkbox. Then enter the workflow name.

## Description

The Description should describe the usage of the workflow.

## Technical Administrator

The access and rights of team administrators are granted to the specified user or user group. This functional access is limited to the associated workflow and corresponding instances.

## Test User

Specify a test user / test users who are allowed to see the development version of the workflow in the runnable list in the web portal. Team administrators and technical administrators can always display them.

## API User

An API user has to be specified if API calls are to be used in a workflow. The specified API user has to be a technical administrator.

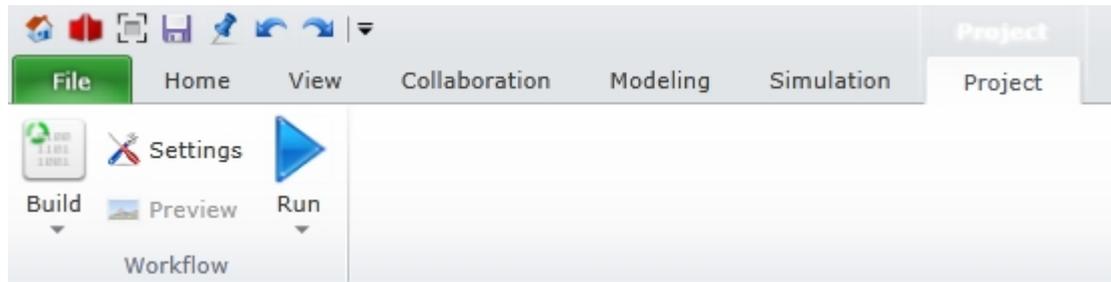
## Tags

Tag the workflow to indicate the usage. The following tags are predefined:

- Documentation Releaser  
The workflow handles an IYOPRO [documentation deployment package](#).
- Library Releaser  
The workflow handles an IYOPRO [library deployment package](#).

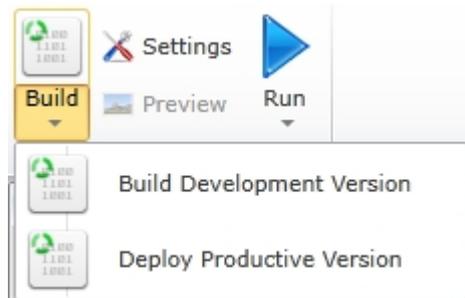
## Build a Workflow

To build a Workflow select the Project Tab out of the [Ribbon](#) menu. If the Project tab is not available make sure, that at least one document of your BPI-Project has the current input focus.



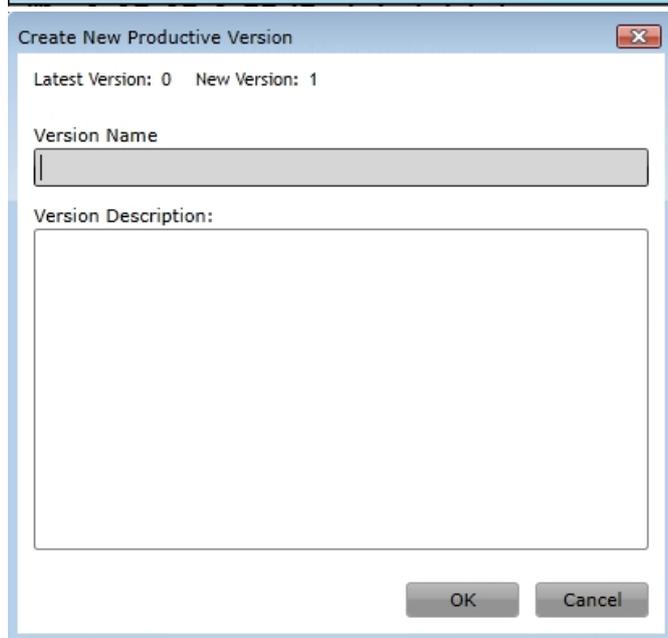
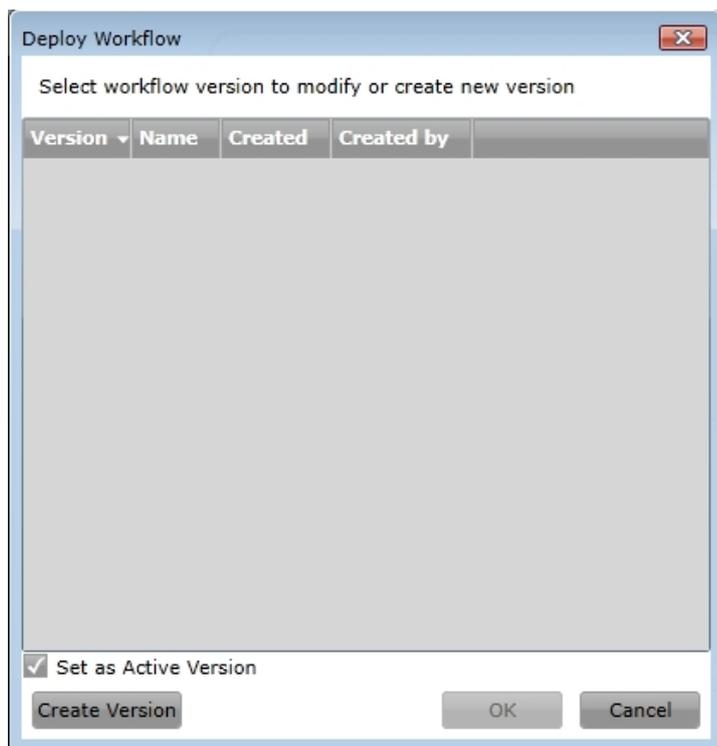
Press the Build button. YOPRO will now compile the workflow definition into a single executable workflow image.

There are two possibilities to build a workflow. It can either be build as a Development Version or as a Productive Version. The default of the Build button is the Development Version.



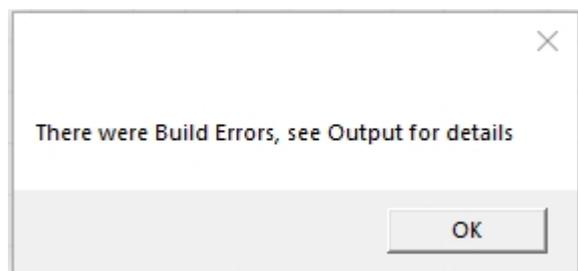
Both versions are separate. A workflow can be modified and tested in the Development Version without any consequences to the (same) workflow built as a Productive Version.

When the workflow is completely modified, one can then select the Productive Version and create or modify a (new) workflow version:

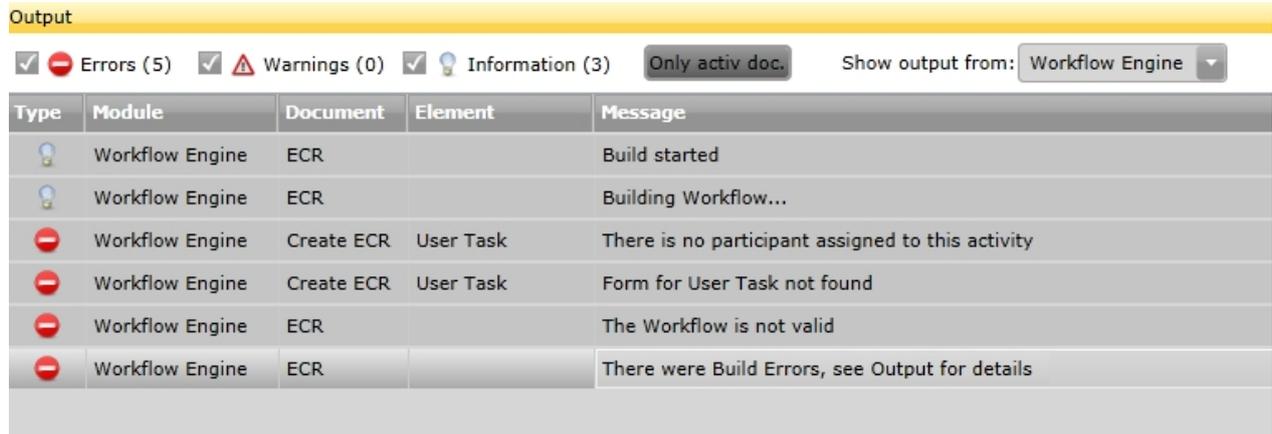


Please note: Only the active version of a workflow is displayed in the [Runnable](#) list. For more information about the uses of the Active Version feature, see also "[Start Workflow outside IYOPRO](#)".

In case of any errors, IYOPRO will show the following message:

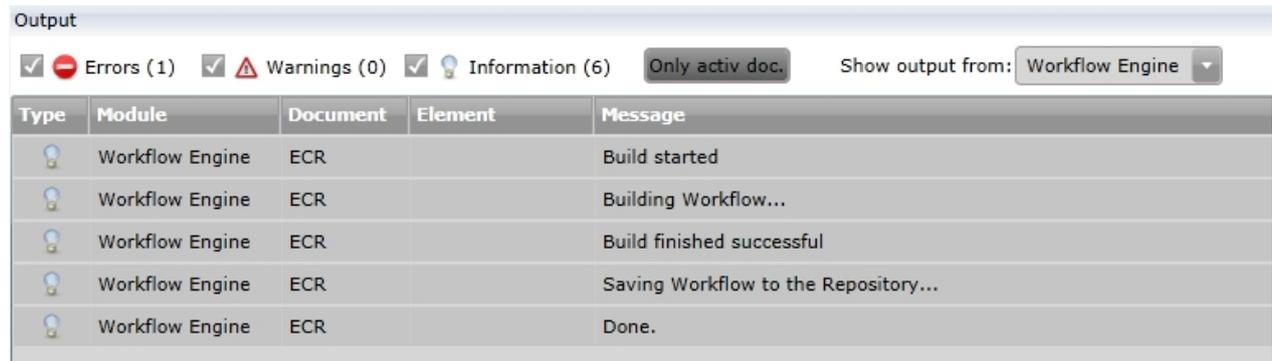


Open the output window to view the details of the build process. It may be a good idea to set the "Show output from" option to "Workflow Engine" and toggle the "only active doc." to filter all messages for the relevant operation. It may then also be necessary to activate the Project in the process explorer to link the active document filter to our output.



In this case the user task misses a participant and a form assignment which both may be corrected in the user task's properties.

After a repeated build, the output is the following:



After a successful build pass, IYOPRO will store the workflow in the "Bin" folder of the project. The workflow will be executable as long as you do not delete this file.



## Language Extensions

This section provides information about IYOPRO specific language extensions.

### Working with Scripts

Aside from the Script task, many shapes allow you to assign Python code snippets to adjust process instance variables as desired.

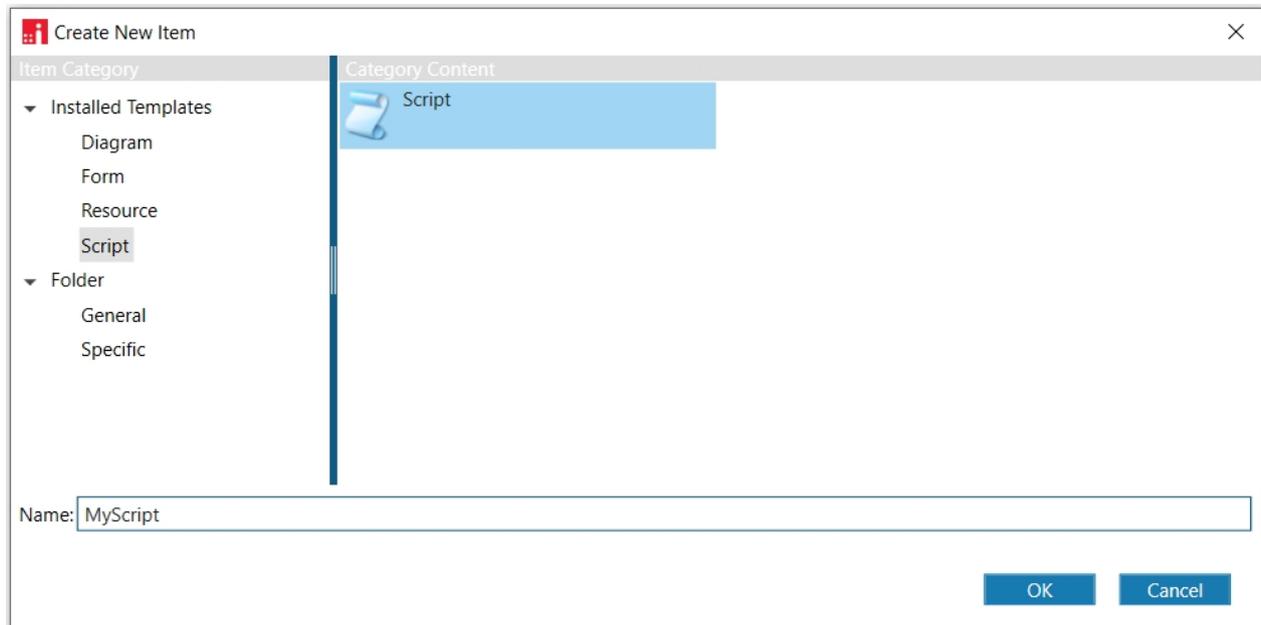
If you have a lot of code and that in many shapes, this can quickly become confusing / difficult to maintain,

the larger the process models become.

Then it will be more useful to centralize the code in one script and provide the code snippets as functions within the script.

In order to be able to do this, there is the Script as a separate file and the language extension "ImportModule" to bind the Script in the shape.

First add a Script to your BPI-Project:

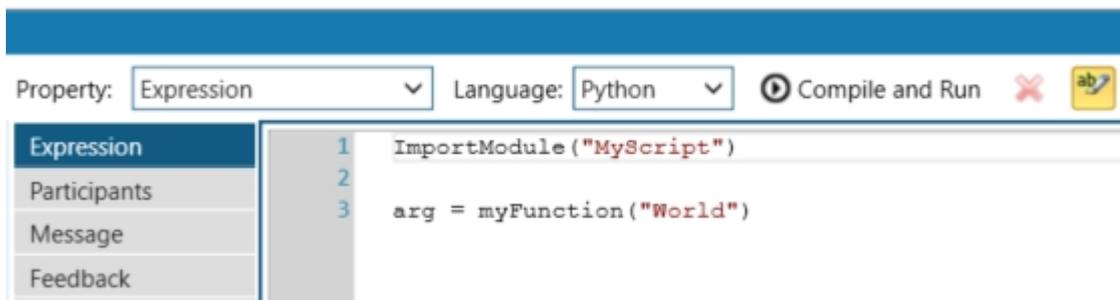


Type in a name for the script. This sample uses "MyScript".

Edit the script and create your functions as desired:

```
1 def myFunction(arg1):
2     arg2 = "Hello " + arg1
3     return arg2
4
```

To bind the script to the shape open the expression editor of the shape and type:



### Using toast/pop-up notifications

If you want to get specific information to the user while they are completing a task, you can use toast notifications.

To generate toast notifications, you can use the following methods:

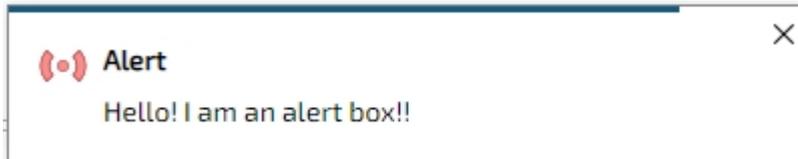
- alert()

- `notify()`

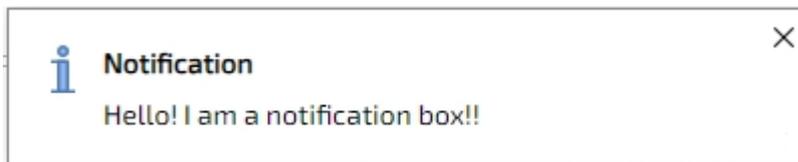
Both methods can be used to display a message. The difference is that the `alert()` method throws an exception, thereby stopping the execution of the script. The `notify()` method only causes an attempt to display a message pop-up in the user's web client asynchronously.

In addition to a message, both methods allow you to set a title, an icon and a timeout. With the exception of the message, however, the other parameters are optional.

```
alert(string message, string title, string iconname, int timeout)
```



```
notify(string message, string title, string iconname, int timeout)
```



If you set a timeout, the toast notification will disappear after the defined time. Otherwise, it remains until the user clicks it away.

For the parameter `iconname`, you can use the icons available in the web portal. You can find an overview here: <https://www.iyopro.com/iyopro/icons.htm>. The icon is displayed in size 24.

Typical icons may be:

error	
warning	
info	
help	
hint	
bell	
alert	

If you have an on-premises installation, you can also look up the existing icons in the icons directory of the frontend directory on the server on which your IYOPRO is installed. As long as you follow the naming scheme there, you may also add your own icons.

## Proceed

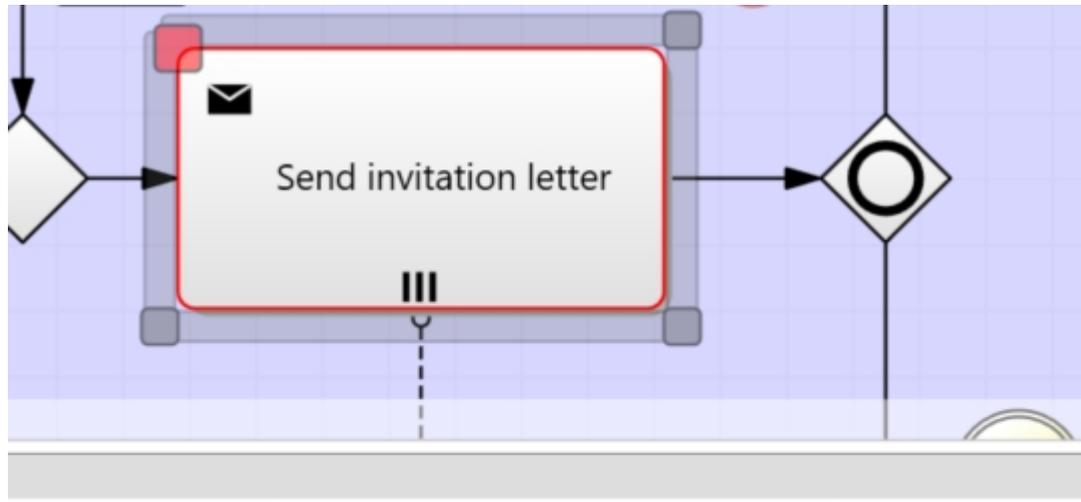
"proceed" can be used as a special replacement for "return" as well as in place of "Result".

The Python return statement is a special statement that can be used within a function or method to terminate the function call and return the result of the function back to the caller.

However, there is no such call in the script expressions of the controls. A return statement cannot be used here. However, the use of return is advantageous, e.g. for case distinctions, since otherwise you would have to work with nested statements.

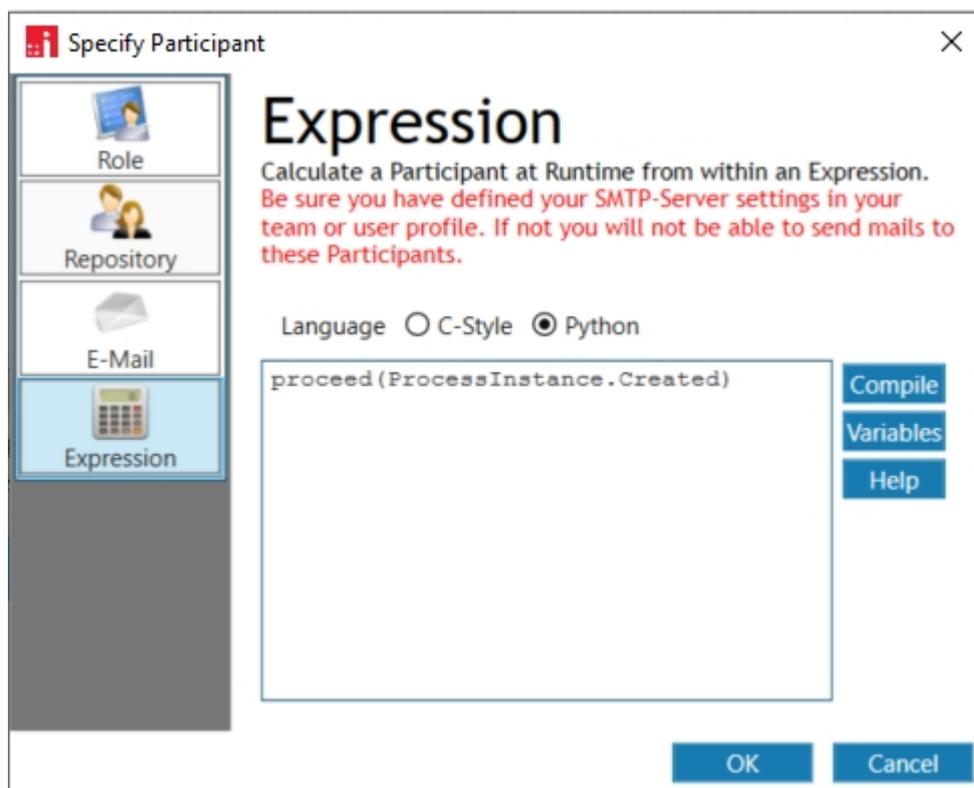
The proceed statement allows you to use it like a return, except that proceed cannot only be used within a function.

### Example - proceed instead of return



```
Language: Python [Compile and Run] [ab] [Insert Code Snippet]
1 #(participant = {"FIRSTNAME": "...", "LASTNAME": "...", "EMAIL": "..."})
2 p = Token.Attributes["participant"]
3 if not p: proceed()
4 lastname = p["LASTNAME"]
5 firstname = p["FIRSTNAME"]
6 Token.SetAttribute("participantName", firstname + ' ' + lastname)
7
```

### Example - proceed instead of Result



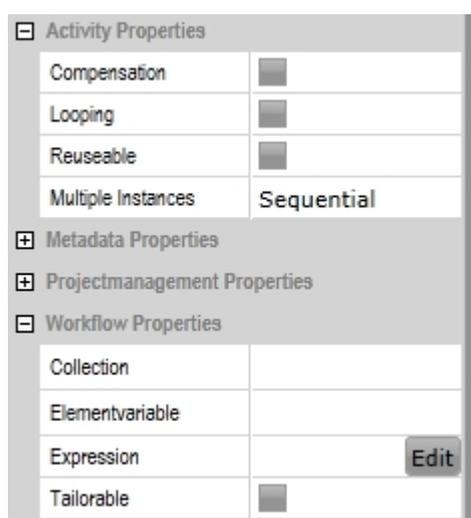
## Working with multiple task instances

There are two types of [multiple instances](#): sequential and parallel. The difference between the two lies in the order of execution.

If a task is sequential, then only one activity can be executed at a time. For a sequential user task, this means, for example, that the first person who has received a task must complete it before the next person gets their task.

If a task is parallel, then zero to all the activities can be performed at the same time.

To work with sequential or parallel task instances, select one of the two options in the "Activity Properties". Define a "Collection" and name an "Elementvariable" in the "Workflow Properties".



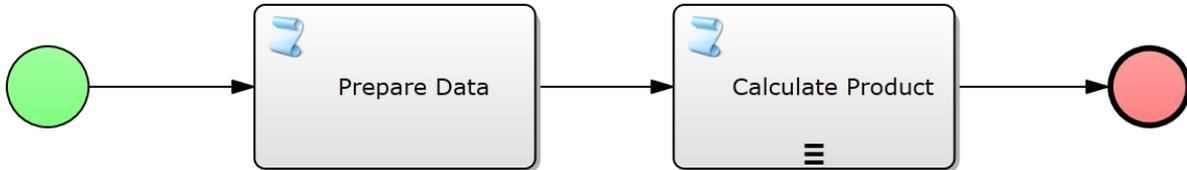
The following table provides an overview of possible scenarios or combinations of this feature with regard to their form data usage:

		type of multiple instances	
		parallel	sequential
type of task	script task	<ul style="list-style-type: none"> <li>• <a href="#">simple and standard mode</a></li> <li>• use <code>Token.Attributes["&lt;Elementvariable&gt;"]</code> to get the elements of the collection</li> </ul>	
	user task	<a href="#">advanced mode:</a> <ul style="list-style-type: none"> <li>• the parallel user task uses an advanced mode to provide the form data</li> <li>• to use the data in other/ following forms, you have to convert it from advanced mode to standard mode</li> </ul>	simple and standard mode
	subprocess task	advanced mode; more details can be found <a href="#">here</a>	

Examples and further information can be found at: [Multiple script task instances](#), [Multiple user task instances](#) and [Multiple subprocess task instances](#)

## Multiple script task instances

### Example



The following workflow properties have been set for the sequential script task:

Workflow Properties	
Collection	Data
Elementvariable	Figure
Expression	<input type="text"/> <input type="button" value="Edit"/>
Tailorable	<input type="checkbox"/>

The "Prepare Data" script task provides the collection and also initializes the result variable for the "Calculate Product" script task.

The workflow expression of the preparation task consists of this Python code:

```
Data = [1, 2, 3, 4]
Product = 1
```

So we have an array "Data" with the numbers we want to calculate the product with.

The workflow expression of the "Calculate Product" script task performs the actual calculation. This is done with the following Python code:

```
Product = Product * Token.Attributes["Figure"]
```

Result = Product

We obtain a value of the collection by addressing its element variable with Token.Attributes, calculate the product of the collection and store it in the result variable.

## Multiple user task instances

The difference in the execution of sequential and parallel task instances becomes apparent in the user tasks.

A useful example of a sequential user task, with its attached form and a number of participants, is that all participants are required

- to validate some of the information contained in the form and / or
- to build on the results of the other participants

This does not apply to parallel user tasks because they are created at the same time and therefore are independent of each other in terms of content.

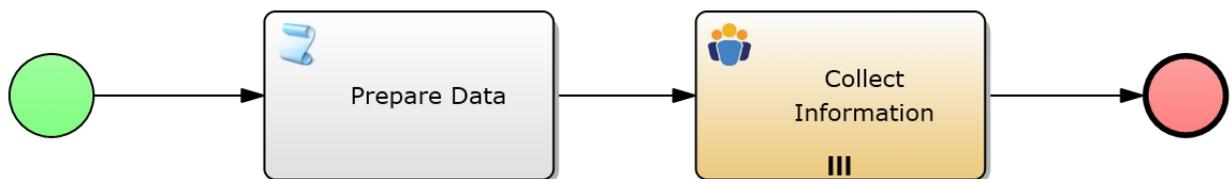
## Providing form data

For this reason, the provision of form data differs between sequential and parallel user tasks instances.

Please refer to the topic "[Providing Form Data](#)" to learn about the simple and standard modes used in the sequential user task instances.

When using parallel user tasks, both modes are not sufficient because all users would see the same form with the same data. In addition, changes made by one user to the data are visible to other users and users can override the other data. Therefore, the parallel user task uses an advanced mode to provide the form data. Instead of having all form data combined in a dictionary, as in standard mode, a dictionary with one entry is used for each collection entry. The keys to the dictionary are the values provided by the collection. The values are again a dictionary of form values.

## Example



The following workflow properties have been set for the parallel user task:

Workflow Properties	
Collection	Recipients
Deadline	None <span>Edit</span>
Elementvariable	User
Expression	<span>Edit</span>

The "Prepare Data" script task provides the collection of the users who will receive a task as well as any needed data for the form data.

The workflow expression of the preparation task consists of this Python code:

```
Recipients = ["paul.paulo@example.com", "justin.justice@example.com"]  
FormData = {}
```

So we have an array "Recipients" with the email addresses of the users to whom we want to assign a task, and a dictionary "FormData" where we want to store the data from the form.

In the participants expression of the "Collect Information" user task, the recipients of the user task are set. This is done with the following Python code:

```
Result = Token.Attributes["User"]
```

We obtain a value of the collection by addressing its element variable with Token.Attributes, which gives us the participant of the user task in this case.

When the parallel user tasks are completed, the data of "FormData" would look like this:

```
FormData = {"paul.paulo@example.com": {"Text": "Hello World"},  
"justin.justice@example.com": {"Text": "Good Morning"}}
```

The workflow can then use the collection to access the data that users specified in the form.

## Extended example

As explained in "Providing form data", the parallel user task uses an advanced mode to provide the form data. In order for the advanced mode form data to be used in standard mode, the data structures must match. Therefore, the form data of a multiple user task must be prepared prior to use in a single user task.



The following workflow properties have been set for the parallel user task:

Workflow Properties	
Collection	Recipients
Deadline	None <span>Edit</span>
Elementvariable	User
Expression	<span>Edit</span>

The "Prepare Data" script task provides the collection of the users who will receive a task as well as any needed data for the form data.

The workflow expression of the preparation task consists of this Python code:

```
Recipients = ["paul.paulo@example.com", "justin.justice@example.com"]  
FormData = {}  
MultipleFormData = {}
```

So we have an array "Recipients" with the email addresses of the users to whom we want to assign a task, a dictionary "FormData" and a separate dictionary "MultipleFormData" where we want to store the data from the form of the parallel user task.

In the participants expression of the "Collect Information" user task, the recipients of the user task are set. This is done with the following Python code:

```
Result = Token.Attributes["User"]
```

We obtain a value of the collection by addressing its element variable with `Token.Attributes`, which gives us the participant of the user task in this case.

When the parallel user tasks are completed, the data of "MultipleFormData" would look like this:

```
MultipleFormData = {"paul.paulo@example.com": {"Text": "Hello World"},  
"justin.justice@example.com": {"Text": "Good Morning"}}
```

If, for example, the information gathered in the parallel user task is to be displayed in "Display Information" in a data grid, the values of the two keys in the dictionary "MultipleFormData" must be transferred into the variable of such a data grid in "FormData". This could be done with the following Python code:

```
FormData["Grid"] = MultipleFormData.values()
```

The data of "FormData" would look like this:

```
FormData = {"Grid": [{"Text": "Hello World"}, {"Text": "Good Morning"}]}
```

### Note

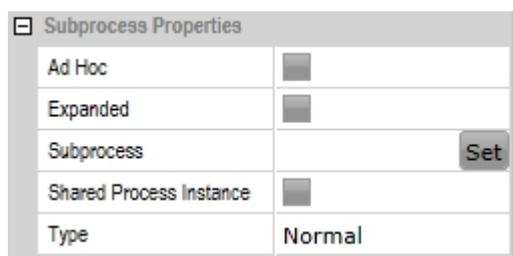
Users do not always have to be addressed via `Token.Attributes` or the collection does not always have to refer to the users in a parallel / sequential user task. The collection may also refer to form data or the like. It only has to be clear to the workflow which user task to assign to which user.

### Multiple subprocess task instances

Please refer to [Multiple user task instances](#) for an explanation of the differences between form data in sequential and parallel user task instances.

When using a subprocess task, it can be decided whether the workflow should create a shared process instance for the main process and the subprocess(es), or whether it should create a separate process instance for each subprocess of the task.

To work with only one process instance, select "Shared Process Instance" in the "Subprocess Properties".



A difference between the two variants is the use of data. If subprocesses are generated as separate process instances, (for example) additional steps are required to transfer data from one process instance to another. These aren't required for a shared process instance.

A token that arrives at a multiple (subprocess) task is split according to the entries in the collection. For this reason, there is a difference between a shared and a separate process instance. In the context of a separate process instance, a separate process instance is created for each token generated in a multiple subprocess task. The workflow can recognize which token is assigned to which subprocess. For a shared process instance, exactly this token-level differentiation still needs to take place.

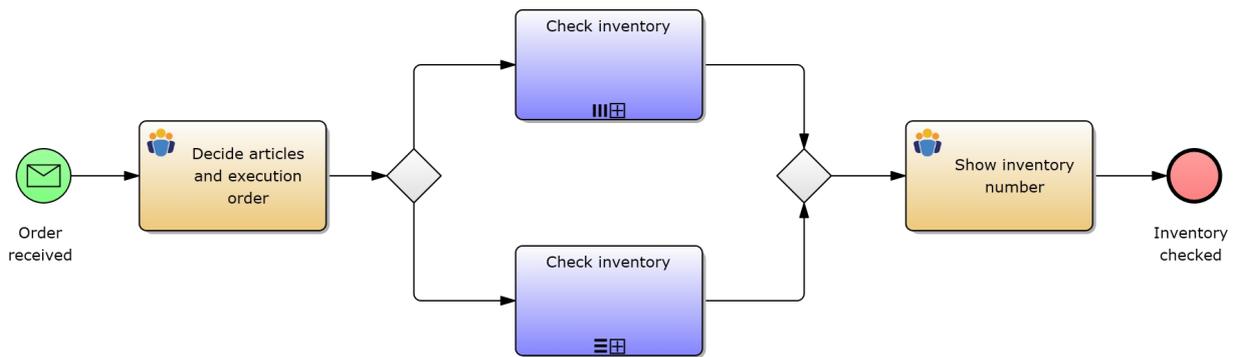
The following table provides an overview of possible combinations for a multiple subprocess task with regard to their form data usage:

		parallel		sequential (the use of private instance attributes in sequential subprocesses is not available)
		private instance attributes		
		no	yes	
shared process instance	yes	<p>If you use a shared process instance and no private instance attributes,</p> <ul style="list-style-type: none"> <li>• you will get a single process instance for the main process and the subprocesses.</li> <li>• there is no separation between the data of the main process and the data of the subprocesses. You don't need to import the variables of the main process into the subprocess or export them out of the subprocess.</li> </ul> <p><a href="#">how to</a></p>	<p>If you use a shared process instance and private instance attributes,</p> <ul style="list-style-type: none"> <li>• you will get a single process instance for the main process and the subprocesses.</li> <li>• the data for each subprocess is separated so that it can be viewed and used separately in the shared process instance</li> </ul> <p><a href="#">how to</a></p>	<p>If you use a shared process instance, there is no separation between the data of the main process and the data of the subprocesses. You don't need to import the variables of the main process into the subprocess or export them out of the subprocess. The sequential subprocesses are executed one after the other and use the data of the previous subprocess.</p> <p><a href="#">how to</a></p>
	no	<p>If you don't use a shared process instance or private instance attributes,</p> <ul style="list-style-type: none"> <li>• a separate process instance is created for each token generated in a multiple subprocess task in addition to the main process. You need to import the variables of the main process into the subprocess and export them out of the subprocess.</li> <li>• you can only get the data of the last completed subprocess back to the main process. In order to get the data of all parallel subprocesses back to the main process, you have to work with data stores</li> </ul> <p><a href="#">how to</a></p>	<p>(the use of private instance attributes is not available)</p>	<p>If you don't use a shared process instance, a separate process instance is created for each token generated in a multiple subprocess task in addition to the main process. To use the data of the main process in the subprocess, you need to import the variables into the subprocess. Export them out of the subprocess to get them back into the main process. The sequential subprocesses are executed one after the other and use the data of the previous subprocess.</p> <p><a href="#">how to</a></p>

### Example: inventory check for an order

In this example, we are an online store and receive an order for multiple items. It is now necessary to check

whether the stock of this article is sufficient.



In the form of the first user task you can see (for example) the article number of the article to be checked as well as decide whether the inventory check should be parallel or sequential.

In the subprocesses happen the actual inventory checks.

The form of the last user task will show you (at least) the current number of articles in stock, which has been determined in the subprocesses. In this example the data will be displayed in a data grid.

The following workflow properties have been set for the multiple subprocess task:

Workflow Properties	
Collection	Order
Elementvariable	Article

## No Private Instance Attributes Separate Process Instances

The following variables have been set for the process:

```
MainData  
MultipleData  
Order  
Article
```

In preparation for the first user task the variables must be defined as follows:

```
MainData = {}  
MultipleData = {}  
Order = []
```

So we have an array "Order" that will consist of the article numbers of the articles for which we need to check the inventory, and a dictionary "MainData" in which we want to collect the data of the whole process as well as a "MultipleData" for the data in the multiple subprocesses.

Since each subprocess has its own process instance, the process instances do not know of the other's form data. You need to import and export the associated variables. The Collection is directly linked to the subprocesses through the setting of the workflow properties, so you don't have to import or export that variable. In separate process instances the workflow can recognize which article number belongs to which subprocess. Because of that you don't address the token to get the value of the Elementvariable (the article number) of the Collection, but address the Elementvariable previously set as a global variable.

So getting the data of MainData into MultipleData could be done with the following Python code:

```
MultipleData[Article] = MainData
```

In order to transfer the data collected in the subprocess and stored in MultipleData back into the main process for further use, the data structure must be compatible. Since the multiple subprocesses use the [advanced mode](#), the data must be prepared.

In this example the information gathered in the subprocess is to be displayed in a data grid, so the values of the keys in the dictionary "MultipleData" must be transferred into the variable of a data grid in "MainData". This could be done with the following Python code:

```
MainData["Grid"] = MultipleData.values()
```

### Differences depending on execution order

```
MultipleData = {"<Article 1>": {"article number": "1234", "stock": "3"},  
"<Article 2>": {"article number": "5678", "stock": "5"}}
```

### Parallel Subprocesses

When the parallel subprocess tasks are completed, "MainData" contains only the data of the last completed subprocess. It would look like this:

```
MainData = {"grid": [{"article number": "5678", "stock": "5"}]}
```

In order to get the data of each of the parallel subprocesses back into the main process, you have to either use the Private Instance Attributes as described [below](#) or work with data stores.

### Sequential Subprocesses

When the sequential subprocess tasks are completed, the data of "MainData" would look like this:

```
MainData = {"grid": [{"article number": "1234", "stock": "3"}, {"article  
number": "5678", "stock": "5"}]}
```

### Shared Process Instances

The following variables have been set for the process:

```
MainData  
MultipleData  
Order
```

In preparation for the first user task the variables must be defined as follows:

```
MainData = {}  
MultipleData = {}  
Order = []
```

So we have an array "Order" that will consist of the article numbers of the articles for which we need to check the inventory, and a dictionary "MainData" in which we want to collect the data of the whole process as well as a "MultipleData" for the data in the multiple subprocesses.

Since there is only one process instance, the subprocesses know the form data of the main process. Therefore, you don't need to import or export any variables. In separate process instances the workflow can recognize which article number belongs to which subprocess. For a shared process instance, exactly this token-level differentiation still needs to take place. Because of that you address the token to get the value of the Elementvariable (the article number) of the Collection.

So getting the data of MainData into MultipleData could be done with the following Python code:

```
MultipleData[Token.Attributes["Article"]] = MainData
```

In order to transfer the data collected in the subprocess and stored in MultipleData back into the main process for further use, the data structure must be compatible. Since the multiple subprocesses use the [advanced mode](#), the data must be prepared.

In this example the information gathered in the subprocess is to be displayed in a data grid, so the values of the keys in the dictionary "MultipleData" must be transferred into the variable of a data grid in "MainData". This could be done with the following Python code:

```
MainData["Grid"] = MultipleData.values()
```

The data in MultipleData that looks like this:

```
MultipleData = {"<Article 1>": {"article number": "1234", "stock": "3"},  
"<Article 2>": {"article number": "5678", "stock": "5"}}
```

will then be transferred into the grid in MainData and have the standard grid structure:

```
MainData = {"grid": [{"article number": "1234", "stock": "3"}, {"article
```

```
number": "5678", "stock": "5"]}]}
```

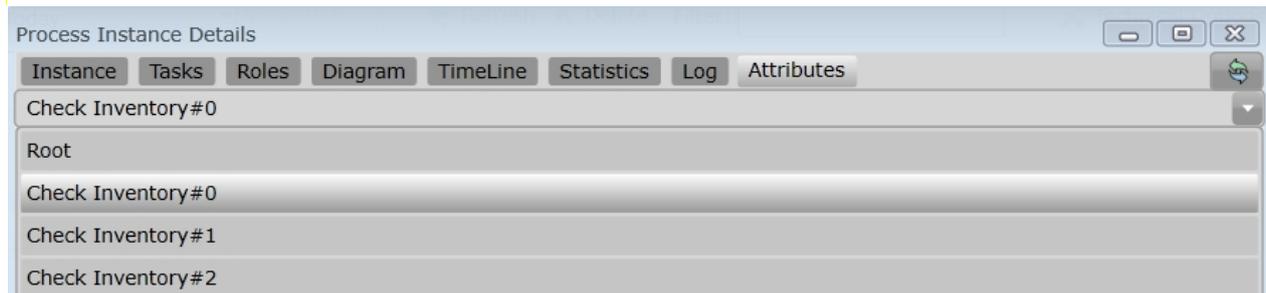
## Using the Private Instance Attributes Shared Process Instances

The Private Instance Attributes are only available for parallel subprocesses with shared process instances. For shared process instances, this feature separates the data for each subprocess. So by using the Private Instance Attributes, the attributes are stored in [datasets](#). It is then possible to access the attributes via these datasets.

To work with the Private Instance Attributes, select the check box in the workflow properties. There you can also choose a name for the Private Instance Attribute Identifier.

Workflow Properties	
Collection	Order
Elementvariable	Article
Expression	<input type="text"/> <input type="button" value="Edit"/>
Private Instance Attribute Identifier	Inventory check
Private Instance Attributes	<input checked="" type="checkbox"/>
Tailorable	<input type="checkbox"/>

In the Process Instance Details, the attributes of the subprocesses are displayed separately:



The following variables have been set for the process:

```
MainData  
MultipleData  
Order
```

In preparation for the first user task the variables must be defined as follows:

```
MainData = {}  
MultipleData = {}  
Order = []
```

So we have an array "Order" that will consist of the article numbers of the articles for which we need to check the inventory, and a dictionary "MainData" in which we want to collect the data of the whole process as well as a "MultipleData" for the data in the multiple subprocesses.

Although you have a shared process instance, because you use the private instance attributes, you need to import the associated variables into the subprocesses. The Collection is directly linked to the subprocesses through the setting of the workflow properties, so you don't have to import or export that variable. In separate process instances the workflow can recognize which article number belongs to which subprocess. For a shared process instance, exactly this token-level differentiation still needs to take place. Because of that you address the token to get the value of the Elementvariable (the article number) of the Collection. So getting the data of MainData into MultipleData could be done with the following Python code:

```
MultipleData[Token.Attributes["Article"]] = MainData
```

Using the private instance attributes separates the data for each subprocess and stores the attributes in [datasets](#). In order to transfer the data collected in the subprocess back into the main process for further use, the data structure must be compatible. Since the multiple subprocesses use the [advanced mode](#), the data must be prepared.

In this example the information gathered in the subprocess is to be displayed in a data grid, so the values of the keys in the dictionary "MultipleData" must be transferred from the datasets into the variable of a data grid in "MainData". This could be done with the following Python code:

```
MainData["grid"] = []
Root = ProcessInstance.GetRootDataSet()
SubDataSets = Root.GetSubDataSets()

for ds in SubDataSets:
    subdata = ds.GetAttribute("MultipleData")
    MainData["grid"].extend(subdata.values())
```

The data in MultipleData that looks like this:

```
Check Inventory #0: MultipleData = {"<Article 0>": {"article number": "1234",
"stock": "3"}}
```

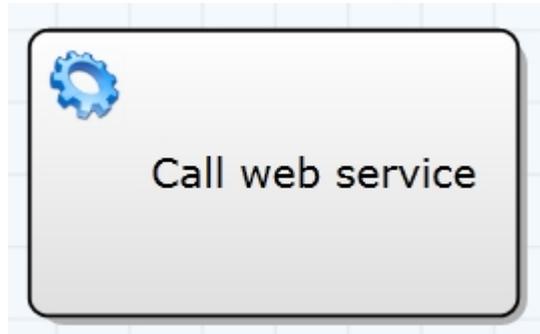
```
Check Inventory #1: MultipleData = {"<Article 1>": {"article number": "5678",
"stock": "5"}}
```

will then be transferred into the grid in MainData and have the standard grid structure:

```
MainData = {"grid": [{"article number": "1234", "stock": "3"}, {"article
number": "5678", "stock": "5"}]}
```

## Calling web services

The service task collaboration diagram control allows to invoke web services from the workflow.



From the elements properties, invoke the service definitions edit mode to maintain the tasks parameters to call the web service.

First specify the URL of the web services wsdl file and press query. IYOPRO will access the services definition via the provided URL

and provide the available actions in respective the combo box.

Select the desired action to have the input parameters populated. You can provide constant data for simple types (string, integer) directly

as the respective value. If you need to provide the parameter value from a process instance variable, you need to provide a conversion logic.

In the simplest case this is a direct assignment like:

**Result = MyProcessInstanceVariable**

However in more complex cases like compound data types the converter allows you to populate the input parameter values as needed.

IYOPRO accepts only one output parameter from the web service. Again a converter is needed to provide the mapping of the web service result to the process instance variable. If the web service result is to be stored in a process instance variable "as-is", the convert logic again is a simple assignment:

### Result = Value

Here "Value" is a predefined variable name containing the web service reply.

Once your web service invocation definition is complete, click on the button "Define Service Call" in the toolbar, to convert the definition to executable code available to the workflow engine.

### Special web service functions

There are some functions specific to the web service definition



1. **Show In Browser**  
Opens the provided URL in a new browser window. This allows a detailed inspection of the web services WSDL.
2. **Show Generated Code**  
Based on the provided details to invoke the web service, IYOPRO generates python code to perform the actual invocation an parameter mapping. The generated code can be displayed with this button.
3. **Test Service Call**  
Invokes the web service with the provided mappings and converters. Each step of the processing provides output. This allows to test the service call and to find potential errors in the conversion of the parameter values
4. **Define Service Call**  
This button allows to "save" the generated code as workflow expression. It is required to use this function to make the web service invocation definitions available as executable code to the workflow engine.

Note: Changes to the source code will be overwritten, if the button is clicked again.

### Processing XML replies

Often web services reply with a XML document specific to the web service. IYOPRO supports the parsing of these XML replies. The C# XML processing can be used to easily process a XML reply.

Example:

Assume the reply from the web service is a XML document like this:

```
<?xml version="1.0" encoding="utf-16"?>
<CurrentWeather>
  <Location>Hamburg-Finkenwerder, Germany (EDHI) 53-32N 009-50E 13M</Location>
  <Time>Jun 27, 2016 - 08:50 AM EDT / 2016.06.27 1250 UTC</Time>
  <Wind> from the SSW (210 degrees) at 14 MPH (12 KT) (direction variable):0</Wind>
  <Visibility> greater than 7 mile(s):0</Visibility>
  <SkyConditions> mostly cloudy</SkyConditions>
```

```

    <Temperature> 69 F (21 C)</Temperature>
    <DewPoint> 51 F (11 C)</DewPoint>
    <RelativeHumidity> 52%</RelativeHumidity>
    <Pressure> 29.97 in. Hg (1015 hPa)</Pressure>
    <Status>Success</Status>
</CurrentWeather>

```

However you are only interested in the relative humidity. Using the C# XML processing one can obtain the value e.g. like this:

```

import clr
clr.AddReference('System.Xml')
from System.Xml import XmlDocument

xmlDoc = XmlDocument()
xmlDoc.LoadXml(Value)

nodes = xmlDoc.SelectNodes("/CurrentWeather/RelativeHumidity")
for node in nodes:
    Result = node.InnerText
    break

```

It is also possible and often easier to use the XML 2 Linq approach to obtain the value:

```

import clr
clr.AddReference('System.Xml.Linq')
from System.Xml.Linq import XElement
xmlElement = XElement.Parse(Value)
Result = xmlElement.Element("RelativeHumidity").Value

```

## Using adapters

You can either use the predefined adapters that are already available or you can create additional adapters.

Either way, the adapters have to be activated in order to use them.

Please note that the adapters are third party products, so we cannot guarantee that they will work.

### Activate adapters

So that you can use the adapters they must be activated in the Backend.Config.xml.

There are two steps to be done:

First, there has to be a key value pair for the adapter path. Take a look at the already activated adapters for reference.

```
<add key="ADAPTER_PATH_<name for the adapter>" value="<path to dll>"/>
```

Second, you have to add the adapter path to the ADAPTER\_PATH\_LIST.

### Predefined adapters

#### CSV

With the CSV adapter you can export a csv file with a specific content.

### Syntax

```
public class CSVDocument
```

## Members

 CSVDocument( <b>string</b> Delimiter = ";", <b>string</b> stringEscape="\\", <b>string</b> Escape="\\", <b>string</b> LineBreak = "\r\n")	The constructor
 <b>void</b> SetValue( <b>int</b> row, <b>int</b> col, <b>object</b> value)	Sets a value into a specified row and column
 <b>void</b> SaveAs( <b>Stream</b> s)	Saves the file

## Sample

This sample creates a .csv file with given values

```
import clr
clr.AddReferenceToFileAndPath(Session.GetAdapterPath("CSV"))
from IYOPROCSvAdapter import CSVDocument
import System
from System.IO import FileStream, FileMode

fs = FileStream("CSVSample.csv", FileMode.Create)
csv = CSVDocument()

#Header
csv.SetValue(1, 1, "Column A")
csv.SetValue(1, 2, "Column B")
csv.SetValue(1, 3, "Column C")

#Content
csv.SetValue(2, 1, "Hello")
csv.SetValue(2, 2, 3)
csv.SetValue(2, 3, 4.2)

csv.SaveAs(fs)
fs.Close()
```

This sample creates and saves a .csv file with values of a data grid

```
import clr
clr.AddReferenceToFileAndPath(Session.GetAdapterPath("CSV"))
from IYOPROCSvAdapter import CSVDocument
import System
from System.IO import FileStream, FileMode

#Specify storage location and file name
localFile = "D:\\CSV Example Folder\\CSVSample.csv"

fs = FileStream(localFile, FileMode.Create)
csv = CSVDocument()

#Header (first row)
csv.SetValue(1, 1, "Column A")
csv.SetValue(1, 2, "Column B")
csv.SetValue(1, 3, "Column C")

#Content (from second row onwards)
i = 2
```

```

for entry in grid:
    csv.SetValue(i, 1, entry["<key>"])
    csv.SetValue(i, 2, entry["<key>"])
    csv.SetValue(i, 3, entry["<key>"])
    i = i + 1

```

```

csv.SaveAs(fs)
fs.Close()

```

## Excel / SpreadsheetLight

The SpreadsheetLight adapter is a complete replacement for the Excel adapter and enables the use of SpreadsheetLight, an open source open XML spreadsheet library.

Further information on the library can be found on the [SpreadsheetLight website](#).

## Sample

This sample is the Python version of an example from the SpreadsheetLight website that saves an .xlsx file with different kinds of content

```

import clr
clr.AddReferenceToFileAndPath(Session.GetAdapterPath("SPREADSHEETLIGHT"))
clr.AddReference("SpreadsheetLight")
clr.AddReference("DocumentFormat.OpenXml")
import System
from SpreadsheetLight import SLDocument, SLConvert

sl = SLDocument()

# set a boolean at "A1"
sl.SetCellValue("A1", True)

# set at row 2, columns 1 through 20, a value that's equal to the column index
for i in range(20):
    sl.SetCellValue(2, i, i)

# set the value of PI
sl.SetCellValue("B3", 3.14159)

''' set the value of PI at row 4, column 2 (or "B4") in string form.
    use this when you already have numeric data in string form and don't
    want to parse it to a double or float variable type
    and then set it as a value.
    Note that "3,14159" is invalid. Excel (or Open XML) stores numerals in
    invariant culture mode. Frankly, even "1,234,567.89" is invalid because
    of the comma. If you can assign it in code, then it's fine, like so:
    fTemp = 1234567.89'''
sl.SetCellValueNumeric(4, 2, "3.14159")

# normal string data
sl.SetCellValue("C6", "This is at C6!")

# typical XML-invalid characters are taken care of,
# in particular the & and < and >
sl.SetCellValue("I6", "Dinner & Dance costs < $10")

''' this sets a cell formula
    Note that if you want to set a string that starts with the equal sign,
    but is not a formula, prepend a single quote.
    For example, "'==' will display 2 equal signs'''

```

```

sl.SetCellValue(7, 3, "=SUM(A2:T2)")

# if you need cell references and cell ranges *really* badly, consider the
SLConvert class.
sl.SetCellValue(SLConvert.ToCellReference(7, 4), "=SUM({0})".format
(SLConvert.ToCellRange(2, 1, 2, 20)))

# dates need the format code to be displayed as the typical date.
# Otherwise it just looks like a floating point number.
sl.SetCellValue("C8", System.DateTime(3141, 5, 9))
style = sl.CreateStyle();
style.FormatCode = "d-mmm-yyyy"
sl.SetCellStyle("C8", style)

sl.SetCellValue(8, 6, "I predict this to be a significant date. Why, I do not
know...")

sl.SetCellValue(9, 4, 456.123789)
# we don't have to create a new SLStyle because
# we only used the FormatCode property
style.FormatCode = "0.000%"
sl.SetCellStyle(9, 4, style)

sl.SetCellValue(9, 6, "Perhaps a phenomenal growth in something?")

sl.SaveAs("SpreadsheetLight.xlsx")

```

## Google Calendar

With the Google Calendar adapter you can create or edit events in the Google Calendar from within IYOPRO.

To be able to interact with Google Calendar, you need authorization. The following link shows possibilities on how to get this authorization: <https://developers.google.com/identity/protocols/oauth2/service-account>

Check and adjust the service account and the json file entry in the GoogleCalendar.config. You should have received a json file after creating the service account. We recommend storing the json file in the folder with the Google Calendar dlls. You can look up the path to your adapter directory in the Backend.config.

## Syntax

```
public class Adapter : IDisposable
```

## Members

 <b>string</b> DefaultTimezone	The default time zone for all Google Calendar events attached to your account (time zone set for your Google account)
 Adapter()	The constructor
 <b>Event[]</b> GetEvents( <b>string</b> calendarId, <b>DateTime?</b> from = <b>null</b> , <b>DateTime?</b> to = <b>null</b> )	Gets an array with the events of the calendar that belongs to the calendar ID. The period can be limited with "from" and "to"
 <b>Color[]</b> GetEventColors()	Gets the colors defined in the Google Calendar as an array
 <b>Event.SourceData</b> CreateEventSourceData( <b>string</b> title= <b>null</b> , <b>string</b> url= <b>null</b> )	Create source data for the event

 <code>Event GetEvent(string calendarId, string eventId)</code>	Get the event that has the specified event ID and is in the calendar that the specified calendar ID belongs to
 <code>Event InsertEvent(string calendarId, string summary, string description, DateTime start, DateTime end, bool fullDay = false, string timezone = null, string colorId = null)</code>	Add an event to a specific calendar using the calendar id. You can add a summary (title of the event) and a description. Enter the time period of the event and whether it is an all-day event or not. You can change the time zone of the event and its color using the color ID
 <code>Event CreateEvent(string summary, string description, DateTime start, DateTime end, bool fullDay = false, string timezone = null, string colorId = null)</code>	Create an event with specific properties
 <code>Event InsertEvent(string calendarId, Event ev)</code>	Add an event to a specific calendar using the calendar id.
 <code>void DeleteEvent(string calendarId, string eventId)</code>	Delete the event with the specified event ID from the calendar. If the event is recurring, use <code>RecurringEventId</code> to delete the following events as well
 <code>Event UpdateEvent(string calendarId, Event ev)</code>	Update an event of the calendar that the specified calendar ID belongs to
 <code>EventDateTime ToEventDateTime(DateTime dateTime, bool fullDay = false, string timezone = null)</code>	Change a C# <code>DateTime</code> value to an <code>EventDateTime</code> value with the additional information about the time zone and whether it is an all-day event
 <code>DateTime? ToDateTime(EventDateTime eventDateTime)</code>	Change an <code>EventDateTime</code> value to a C# <code>DateTime</code> value
 <code>void Dispose()</code>	Releases the resources used

You can also use the [event resources](#) of the Calendar API.

## Syntax

```
public class Color
```

## Members

 <code>string Id</code>	The id of the color
 <code>string Background</code>	The (background) color of the event
 <code>string Foreground</code>	The font color suitable for the background color (readability)

## Sample

This sample inserts an event into a Google Calendar and prints the id of that event

```
import clr
clr.AddReferenceToFileAndPath(Session.GetAdapterPath("GOOGLECALENDAR"))
from IYOPRO.GoogleCalendar import Adapter
from System import DateTime

calendarId = "<Google Calendar id>"
with Adapter() as gc:
    event = gc.InsertEvent(calendarId, "IYOPRO Google Calendar adapter test",
"Generated in script task", DateTime.Now.AddHours(24), DateTime.Now.AddHours
(25))
```

```
print event.Id
```

## OpenEDM (IYOPRO PLM)

With the OpenEDM adapter you can connect IYOPRO BPM to your IYOPRO PLM. The functions available depend on your IYOPRO PLM implementation.

## Syntax

```
public class Adapter : IDisposable
```

## Members

Adapter()	The constructor
string OECallFunction(string username, string password, string function, string data)	Calls a function of the OpenEDM
XmlDocument OECallFunctionReturnXML (string username, string password, string function, string data)	Returns the xml to the function call

## Sample

This sample loads an Adapter to the PDM-System IYOPRO PLM (formerly OpenEDM) and performs a function call

```
import clr
clr.AddReferenceToFileAndPath(Session.GetAdapterPath
("IYOPRO.OPENEDM"))
from OpenEDMAdapter import *

a = Adapter()
OEResult = a.OECallFunction("", "", "DEMO.FUNCTION", "")
data = eval(OEResult)
```

## PDF

This adapter only enables the use of the two .NET libraries PDFSharp and Migradoc. Further information on the libraries can be found here: <http://www.pdfsharp.net/>

## Sample 1

This sample generates a PDF with a link as text and a few other properties

```
#IYOPRO.Backend
# Include wrappers for PDF libs and import namespaces
# -----
import clr
clr.AddReferenceToFileAndPath(Session.GetAdapterPath("IYOPRO.PDF"))
clr.AddReference("MigraDoc.DocumentObjectModel-WPF")
clr.AddReference("MigraDoc.Rendering-WPF")
from MigraDoc.DocumentObjectModel import Document, TabAlignment, StyleNames,
Orientation, HyperlinkType, TextFormat, Color, Shapes
from MigraDoc.Rendering import PdfDocumentRenderer
clr.AddReference("PdfSharp-WPF")
from PdfSharp.Pdf import PdfFontEmbedding
```

```

# To save / read the PDF in a stream
# -----
import System
from System.IO import MemoryStream

# Define the file name for the generated PDF
# -----
PdfName='IYOPRO-Sample.pdf'

# Create PDF document container and set properties
# -----
pdf = Document()

pdf.Info.Title = "Dynamic PDF"
pdf.Info.Subject = "Sample script for PDF creation"
pdf.Info.Author = "IYOPRO Support"
pdf.Info.Keywords = "IronPython, Migradoc, PDFSharp, PDF"

# Create text styles
# -----
style = pdf.Styles["Normal"];
style.Font.Name = "Verdana";

# Set content including page layout
# -----
section = pdf.AddSection()
pageSetup = pdf.DefaultPageSetup.Clone();
pageSetup.Orientation = Orientation.Landscape;
section.PageSetup = pageSetup

paragraph = section.AddParagraph()
paragraph.Format.Font.Color = Color.FromCmyk(66, 39, 0, 7);
paragraph = paragraph.AddHyperlink("http://www.pdfsharp.net/
PDFsharpFeatures.ashx",HyperlinkType.Web);
paragraph.AddFormattedText("More at http://www.pdfsharp.net/
PDFsharpFeatures.ashx", TextFormat.Underline);

# Render PDF and set other properties
# -----
pdf.UseCmykColor = True;
pdfRenderer = PdfDocumentRenderer(True, PdfFontEmbedding.Always)
pdfRenderer.Document = pdf
pdfRenderer.RenderDocument()
pdfRenderer.PdfDocument.Info.Author = "Support"
pdfRenderer.PdfDocument.Info.CreationDate = System.DateTime.Now
pdfRenderer.PdfDocument.Info.Creator = "Iron Python"
pdfRenderer.PdfDocument.Info.Keywords = "Test, IYOPRO"
pdfRenderer.PdfDocument.Info.ModificationDate = System.DateTime.Now
pdfRenderer.PdfDocument.Info.Subject = "An example"
pdfRenderer.PdfDocument.Info.Title = "My generated PDF"

pdfRenderer.PdfDocument.SecuritySettings.PermitAnnotations = False
pdfRenderer.PdfDocument.SecuritySettings.PermitExtractContent = False
pdfRenderer.PdfDocument.SecuritySettings.PermitFormsFill = False
pdfRenderer.PdfDocument.SecuritySettings.PermitModifyDocument = False

# Save PDF
# -----
with MemoryStream() as mem:
    pdfRenderer.Save(mem, False)

```

```
ProcessInstance.WriteFileFromMemoryStream(PdfName,mem)
```

```
# Make file name(s) available for attachments  
attachments =[ PdfName ]
```

## Sample 2

This sample concatenates PDFs that were previously uploaded to a form

```
#IYOPRO.Backend  
# Include wrappers for PDF libs and import namespaces  
# -----  
import clr  
clr.AddReferenceToFileAndPath(Session.GetAdapterPath("IYOPRO.PDF"))  
clr.AddReference("PdfSharp-WPF")  
from PdfSharp.Pdf import PdfDocument  
from PdfSharp.Pdf.IO import PdfReader, PdfDocumentOpenMode  
# For marking  
from PdfSharp.Drawing import XFont, XFontStyle, XGraphics, XBrushes, XRect,  
XStringFormat  
  
# To save / read the PDF in a stream  
# -----  
from System.IO import MemoryStream  
  
# Define the file name for the generated PDF  
# -----  
PdfName='IYOPRO-Concat-Sample.pdf'  
  
# Generate output PDF  
# -----  
outputDocument = PdfDocument()  
  
# For marking  
font = XFont("Verdana", 10, XFontStyle.Bold)  
  
# Process uploaded files  
# -----  
files = Form["Uploads"]  
numfiles = len(files)  
for fidx, file in enumerate(files):  
    ProcessInstance.Info("Processing "+file)  
    # MemoryStream des Files erzeugen  
    with ProcessInstance.ReadFileToStream(file) as instream:  
        inputDocument = PdfReader.Open(instream, PdfDocumentOpenMode.Import)  
        numpages = inputDocument.Pages.Count  
        for pidx in xrange(numpages):  
            inpage = inputDocument.Pages[pidx]  
            outpage = outputDocument.AddPage(inpage)  
            # Markierung  
            gfx = XGraphics.FromPdfPage(outpage)  
            info = "Seite "+unicode(pidx+1)+" / "+unicode(numpages)+" von  
Dokument "+unicode(fidx+1)+" / "+unicode(numfiles)  
            ProcessInstance.Info("Adding "+info)  
            gfx.DrawString(info, font, XBrushes.Black, XRect(0.0,  
outpage.Height.Value-30, outpage.Width.Value, outpage.Height.Value),  
XStringFormat.TopLeft)  
  
# Save PDF
```

```
# -----
with MemoryStream() as mem:
    outputDocument.Save(mem, False)
    ProcessInstance.WriteFileFromMemoryStream(PdfName, mem)

# Make file names available for attachments
# -----
attachments =[ PdfName ]
```

## PowerShell

You can use the PowerShell adapter to run your PowerShell scripts and get the corresponding output or errors.

## Syntax

```
public class PShell
```

## Members

 <code>void AddScript(string code)</code>	Add a script code
 <code>void SetParameter(string name, object value)</code>	Set a parameter with the specified name and its value
 <code>bool Execute()</code>	Execute a PowerShell instance
 <code>bool HasOutput()</code>	Indicates whether there was an output
 <code>List&lt;object&gt; GetOutput()</code>	Get a list with the output
 <code>bool HasError()</code>	Indicates whether there was an error
 <code>List&lt;object&gt; GetErrors()</code>	Get a list of errors that have occurred

## Sample

This sample gives you information about your computer.

```
import clr
clr.AddReferenceToFileAndPath(Session.GetAdapterPath("IYOPRO.POWERSHELL"))
from IYOPROPsAdapter import PShell

code = r'''
param
(
    [string]$Computer = XXX
)
Get-Wmiobject -computername $Computer win32_computersystem
'''

code = code.Replace("XXX", "\".\\"")

shell = PShell()
shell.AddScript(code)

shell.SetParameter("Computer", "<device name>")

ok = shell.Execute()
results = shell.GetOutput()
errors = shell.GetErrors()
if results.Count > 0:
    o = results[0]
    print(o.SystemType)
```

```

print(o.BootupState)
print(o.Manufacturer)
print(o.Model)

if errors.Count>0:
    o = errors[0]
    print(o.Exception.Message)

```

## RemoteExec

You can use the RemoteExec adapter to execute a given command on a specified target machine.

## Syntax

```
public class RemoteExec
```

## Members

RemoteExec(string computer, string user, string password, string command, string commandargs = null, string extraRemoteExecArgs = null)	The constructor
int RunProcess()	Run the process. If successful, the exit code of the process is returned
bool HasStartError()	Indicates whether there was a start error
string GetStartError()	Returns the start error that occurred
string GetCommandLine()	Get a command line with the remoteExec file name and arguments

## Sample

This sample shows how you execute Notepad on the specified computer as the session user

```

import clr
clr.AddReferenceToFileAndPath(Session.GetAdapterPath("IYOPRO.REMOTEEXEC"))
from IYOPRORemoteExecAdapter import RemoteExec

r = RemoteExec("<device name>", "", "", "notepad.exe", None, "-su -cs")
x = r.RunProcess()
print(r)
print(r.GetCommandLine())
print(r.HasStartError())

```

## SAP

With the SAP adapter you can connect IYOPRO to your SAP.

Check and adjust the entries in the IYOPRO.SAP.config so that you have the correct destinations and their required parameters.

Example for the entries of the config:

```

Comma separated list of known destination names
<add key="destinations" value="T11,T12"/>

```

```

Parameters of the T11 destination as required
<add key="T11_USER" value="<USERNAME>"/>
<add key="T11_PASSWD" value="<PASSWORD>"/>
<add key="T11_CLIENT" value="500"/>

```

```
<add key="T11_LANG" value="DE"/>
<add key="T11_ASHOST" value="<IP address>"/>
<add key="T11_SYS_NR" value="01"/>
```

## Syntax

```
public class ConnectionManager
```

## Members

 <b>RfcDestination</b> GetDestination( <b>string</b> name)	Get config parameters for the specified destination name
---	--

## Syntax

```
public class RfcDebugger
```

## Members

 <b>bool</b> CallFailed	Indicates whether the call failed
 RfcDebugger( <b>IRfcFunction</b> func)	The constructor
 <b>string</b> Call( <b>RfcDestination</b> dest, <b>bool</b> escapeResultForSQL = <b>true</b> , <b>Encoding</b> encodingToUse= <b>null</b> )	Returns log data as xml
 <b>Exception</b> GetCallException()	Returns the call error if an exception occurred

## Sample

This sample gets the IP address as well as the log data of a specified function of a given destination.

```
#IYOPRO.Backend
import clr
clr.AddReferenceToFileAndPath(Session.GetAdapterPath("IYOPRO.SAP"))
from IYOPRO.SAP import *
```

```
dest = ConnectionManager.GetDestination("T11")
```

```
func = dest.Repository.CreateFunction("RFC_SYSTEM_INFO")
func.Invoke(dest)
si = func.GetStructure("RFC_SI_EXPORT")
print "SI.RFCIPADDR", si.GetValue("RFCIPADDR")
```

### print result:

```
SI.RFCIPADDR ...
```

### ALTERNATIVE:

```
func = dest.Repository.CreateFunction("RFC_SYSTEM_INFO")
logger = RfcDebugger(func)
logdata = logger.Call(dest, True)
si = func.GetStructure("RFC_SI_EXPORT")
print "SI.RFCIPADDR", si.GetValue("RFCIPADDR")
print logdata
```

### print result:

```
SI2.RFCIPADDR ...
<?xml version="1.0" encoding="utf-16"?>
```

```

<RfcCallDump Timestamp="2021-07-02T11:53:29.1387114+02:00"
FunctionModule="RFC_SYSTEM_INFO">
  <Destination AppServerHost="..." AppServerService="" Client="500"
Language="DE" MessageServerHost="" MessageServerService="" Name="T11"
SAPRouter="" SystemId="" SystemNumber="01" User="RFC_PLMDEV" />
  <PreCall>
    <Function Name="RFC_SYSTEM_INFO">
      <IMPORTING />
      <EXPORTING>
        <Parameter Name="CURRENT_RESOURCES" Type="INT4" Length="4" Value="0" /
>
        <Parameter Name="FAST_SER_VERS" Type="INT4" Length="4" Value="0" />
        <Parameter Name="MAXIMAL_RESOURCES" Type="INT4" Length="4" Value="0" /
>
        <Parameter Name="RECOMMENDED_DELAY" Type="INT4" Length="4" Value="0" /
>
      <Structure Name="RFCSI" Type="STRUCTURE">
        <Field Name="RFCPROTO" Type="CHAR" Length="3" Value="" />
        <Field Name="RFCCHARTYP" Type="CHAR" Length="4" Value="" />
        <Field Name="RFCINTTYP" Type="CHAR" Length="3" Value="" />
        <Field Name="RFCFLOTYP" Type="CHAR" Length="3" Value="" />
        <Field Name="RFCDEST" Type="CHAR" Length="32" Value="" />
        <Field Name="RFCHOST" Type="CHAR" Length="8" Value="" />
        <Field Name="RFCSYSID" Type="CHAR" Length="8" Value="" />
        <Field Name="RFCDATABS" Type="CHAR" Length="8" Value="" />
        <Field Name="RFCDBHOST" Type="CHAR" Length="32" Value="" />
        <Field Name="RFCDBSYS" Type="CHAR" Length="10" Value="" />
        <Field Name="RFCSAPRL" Type="CHAR" Length="4" Value="" />
        <Field Name="RFCMACH" Type="CHAR" Length="5" Value="" />
        <Field Name="RFCOPSYS" Type="CHAR" Length="10" Value="" />
        <Field Name="RFCTZONE" Type="CHAR" Length="6" Value="" />
        <Field Name="RFCDAYST" Type="CHAR" Length="1" Value="" />
        <Field Name="RFCIPADDR" Type="CHAR" Length="15" Value="" />
        <Field Name="RFCKERNRL" Type="CHAR" Length="4" Value="" />
        <Field Name="RFCHOST2" Type="CHAR" Length="32" Value="" />
        <Field Name="RFCSI_RESV" Type="CHAR" Length="12" Value="" />
        <Field Name="RFCIPV6ADDR" Type="CHAR" Length="45" Value="" />
      </Structure>
      <Parameter Name="S4_HANA" Type="CHAR" Length="1" Value="" />
    </EXPORTING>
    <CHANGING />
    <TABLES />
    <EXCEPTIONS />
  </Function>
</PreCall>
<Call DurationInSeconds="0.000494" Status="Success" />
<PostCall>
  <Function Name="RFC_SYSTEM_INFO">
    <IMPORTING />
    <EXPORTING>
      <Parameter Name="CURRENT_RESOURCES" Type="INT4" Length="4"
Value="10" />
      <Parameter Name="FAST_SER_VERS" Type="INT4" Length="4" Value="1" />
      <Parameter Name="MAXIMAL_RESOURCES" Type="INT4" Length="4"
Value="11" />
      <Parameter Name="RECOMMENDED_DELAY" Type="INT4" Length="4" Value="0" /
>
    <Structure Name="RFCSI" Type="STRUCTURE">
      <Field Name="RFCPROTO" Type="CHAR" Length="3" Value="011" />
      <Field Name="RFCCHARTYP" Type="CHAR" Length="4" Value="4103" />
    </Structure>
  </Function>
</PostCall>

```

```

<Field Name="RFCINTTYP" Type="CHAR" Length="3" Value="LIT" />
<Field Name="RFCFLOTYP" Type="CHAR" Length="3" Value="IE3" />
<Field Name="RFCDEST" Type="CHAR" Length="32" Value="..." />
<Field Name="RFCHOST" Type="CHAR" Length="8" Value="..." />
<Field Name="RFCSYSID" Type="CHAR" Length="8" Value="..." />
<Field Name="RFCDATABS" Type="CHAR" Length="8" Value="..." />
<Field Name="RFCDBHOST" Type="CHAR" Length="32" Value="..." />
<Field Name="RFCDBSYS" Type="CHAR" Length="10" Value="..." />
<Field Name="RFC SAPRL" Type="CHAR" Length="4" Value="750" />
<Field Name="RFCMACH" Type="CHAR" Length="5" Value=" 390" />
<Field Name="RFCOPSYS" Type="CHAR" Length="10" Value="Linux" />
<Field Name="RFCTZONE" Type="CHAR" Length="6" Value=" 3600" />
<Field Name="RFC DAYST" Type="CHAR" Length="1" Value="X" />
<Field Name="RFCIPADDR" Type="CHAR" Length="15" Value="..." />
<Field Name="RFC KERNRL" Type="CHAR" Length="4" Value="753" />
<Field Name="RFCHOST2" Type="CHAR" Length="32" Value="..." />
<Field Name="RFC SI_RESV" Type="CHAR" Length="12" Value="" />
<Field Name="RFC IPV6ADDR" Type="CHAR" Length="45" Value="..." />
</Structure>
<Parameter Name="S4_HANA" Type="CHAR" Length="1" Value="" />
</EXPORTING>
<CHANGING />
<TABLES />
<EXCEPTIONS />
</Function>
</PostCall>
</RfcCallDump>

```

## SharePoint

The SharePoint adapter provides a method with which a ClientContext can be created.

You can specify the URL and login data in the IYOPRO.SHAREPOINT.config. If no login data are specified, access is carried out as a service user.

You can find more information on the usage under the following links:

Examples: <https://docs.microsoft.com/de-de/sharepoint/dev/sp-add-ins/complete-basic-operations-using-sharepoint-client-library-code>

API: [https://docs.microsoft.com/en-us/previous-versions/office/sharepoint-csom/ee538685\(v=office.15\)](https://docs.microsoft.com/en-us/previous-versions/office/sharepoint-csom/ee538685(v=office.15))

## Syntax

```
public class Adapter : IDisposable
```

## Members

 Adapter()	The constructor
 ClientContext GetContext(string url)	Get the client context for the specified URL
 void Dispose()	Releases the resources used

## Sample

This sample is the Python version of the example found via the link above and gets the title of a SharePoint website.

```

import clr
clr.AddReferenceToFileAndPath(Session.GetAdapterPath("IYOPRO.SHAREPOINT"))
from IYOPRO.SHAREPOINT import *

```

```

with Adapter() as sharepoint:
    ctx = sharepoint.GetContext("<URL>")
    web = ctx.Web
    ctx.Load(web)
    ctx.ExecuteQuery()
    print web.Title

```

## Word

The Word adapter lets you modify the content of a Word document.

## Syntax

```
public class Adapter : IDisposable
```

## Members

 <code>bool</code> SetDocumentBytes( <code>byte[]</code> data)	Load document as binary data
 <code>MemoryStream</code> GetDocumentStream()	Read document in binary data from memory
 <code>bool</code> ReplaceBookmark( <code>string</code> bookmarkName, <code>string</code> bookmarkText)	Insert the specified text at the place of the given bookmark. (Replace placeholder text, if existing)
 <code>bool</code> ReplaceBookmarkKeepFormat( <code>string</code> bookmarkName, <code>string</code> bookmarkText)	Insert the specified text at the place of the given bookmark while keeping the existing format
 <code>bool</code> ReplaceBookmark2( <code>string</code> name, <code>string</code> text, <code>string</code> formats)	Insert the specified text at the place of the given bookmark. You may work with multiline text (e.g. the addressee)
 <code>bool</code> ReplaceBookmarkAsHyperlink( <code>string</code> name, <code>string</code> text, <code>string</code> url)	Insert the specified text as a hyperlink at the place of the given bookmark
 <code>bool</code> FillBookmarkTable( <code>string</code> name, <code>DataRows</code> datarows, <code>bool</code> keepfirstrow = <code>true</code> )	Insert the specified data table at the place of the given bookmark
 <code>void</code> Dispose()	Releases the resources used

## Syntax

```
public class DataRows : List<DataRow>
```

## Syntax

```
public class DataRow : List<DataColumn>
```

## Members

 <code>DataFormat</code> Format	Row format
--	------------

## Syntax

```
public class DataColumn
```

## Members

 <code>string</code> Content	Content of the column
 <code>DataColumnAlignment</code> Align	Alignment of the column content
 <code>DataFormat</code> Format	Format of the column content

## Syntax

```
public enum DataFormat
```

### Members

 None = 0	Normal content
 Bold = 1	Content in bold
 Italic = 2	Content in italics
 UnderlineSingle = 4	Content with a single underline
 UnderlineDouble = 8	Content with a double underline

## Syntax

```
public enum DataColumnAlignment
```

### Members

 Left = 0	Content is left-justified
 Right = 1	Content is right-justified
 Center = 2	Content is centered
 Decimal = 3	Content is aligned with the decimal point

## Sample

This sample modifies the content of a Word document. The variable myfile points to a file which may either be uploaded from a web form or which could be retrieved out of a data store.

```
import clr
clr.AddReferenceToFileAndPath(Session.GetAdapterPath("IYOPRO.WORD"))
from IYOPRO.WORD import *
data = ProcessInstance.ReadFileBytes(myfile)
with Adapter() as word:
    word.SetDocumentBytes(data)
    word.ReplaceBookmark('MyBookMark', 'Hello World')
    stream = word.GetDocumentStream()
    ProcessInstance.WriteFileFromMemoryStream(myfile, stream)
```

## Creating adapters

You may use other adapters than the already defined ones.

To do this, you need to add the desired external library to IYOPRO and then activate the new adapter.

To find out where to add them and how to activate them, see the following topic: [Activate adapters](#)

## Execute Workflows

To run and monitor workflows it is necessary to complete the following steps:

1. Define a Workflow as described [here](#)
2. [Run the workflow](#). Especially on the first time [monitor the process instance](#). Verify, if the process instance variables contain the expected values and read the process log to detect warnings and errors.
3. Verify even finished process instances. It's important to verify if there where errors or unexpected

warnings.

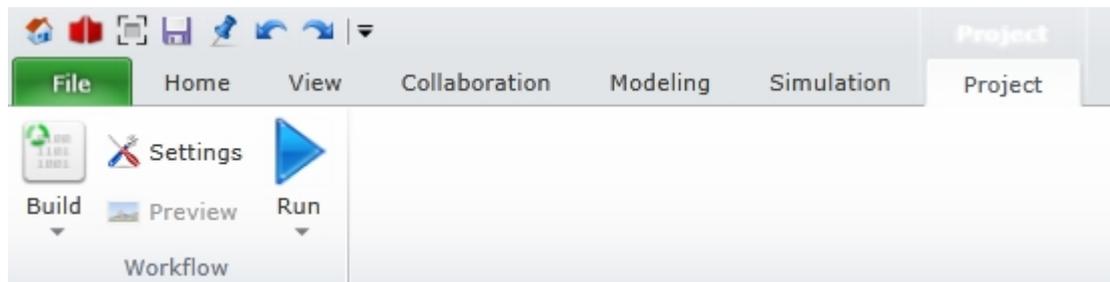
## Run a Workflow

There are multiple ways available to run a workflow

- Run Workflow from Project
- Run Workflow from the Repository
- Run Workflow from the Workflow Portal

## Run Workflow from Project

To run a Workflow from a project select the Project Tab out of the [Ribbon](#) menu. If the Project tab is not available, make sure that at least one document of your BPI-Project has the current input focus.



Press the Run button to start a workflow.

There are two possibilities to run a workflow. It can either start as a Development Version or as an Active Version. The default of the Run button is the Development Version.

Depending on the version specified as active, the Active Version that is started changes. For example, to start a workflow as a previously built Productive Version, it has to be set as the Active Version. To change the active version or to find out which version is currently active, press "Set Active Version". A dialog appears with all built versions. The currently active version is selected.



## Run Workflow from the Repository

Open the [Repository](#) or the [Portal](#) and expand the Workflow node. Click on the "Runnable" node to receive a list of all available workflows. The workflows displayed in the list are those that are set as Active Version.

Repository Tree		Refresh				
		Name	Team	Created at	Created by	Workflow Id
Workflow		ECR	Thomas Roehrig			10751
Tasks		ECR Develop sample	Thomas Roehrig			11187
Instances		Max Freigabe	Max Mustermann GmbH			10041
Runnable		Send Newsletter	Thomas Roehrig			10013
Recent Items						

Double-click on the appropriate workflow to start.

Then either a web version of your [start form](#) will appear or the IYOPRO Start Workflow Dialog, depending on the settings in the workflow definition.

If a start form has been declared, a process instance will be created when the form is submitted.

## Start Workflow Dialog

The Start Workflow Dialog is used for all types of workflows, which must be manually started.

It shows the name and some organizational information as well as a preview image and a description.

**Temperature Monitor**

Temperature Monitor

**Start**

This process is used to monitor the temperature of a heating enging. If the temperature exceeds 35 degree a warning mail will be send.

Type: Workflow

Size:

Created: Friday, September 07, 2012 2:20:25 PM by Thomas Roehrig

Modified: Friday, September 07, 2012 2:20:25 PM by Thomas Roehrig

```

    graph LR
      Start(( )) --> GetTemp[Get Temperature]
      GetTemp --> Wait[Wait 10 min]
      Wait --> Decision{Above 35 degrees?}
      Decision -- No --> GetTemp
      Decision -- Yes --> SendEmail[Send email]
      SendEmail --> End(( ))
  
```

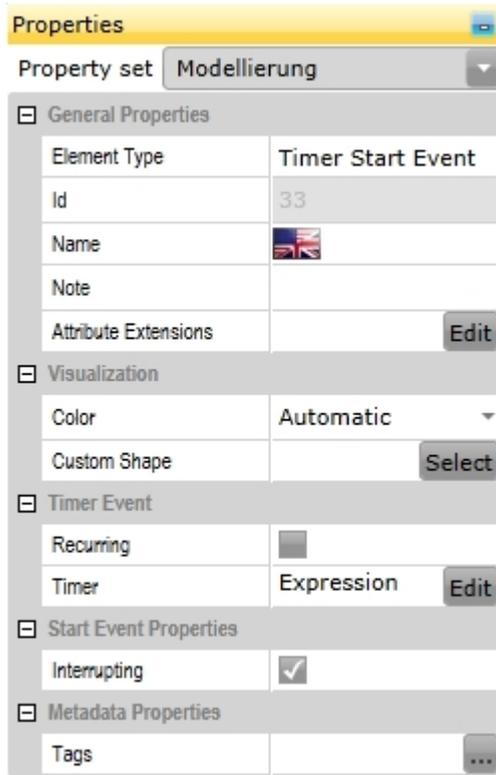
To start such a workflow hit the "Start" button.

## Run a Workflow by Timer

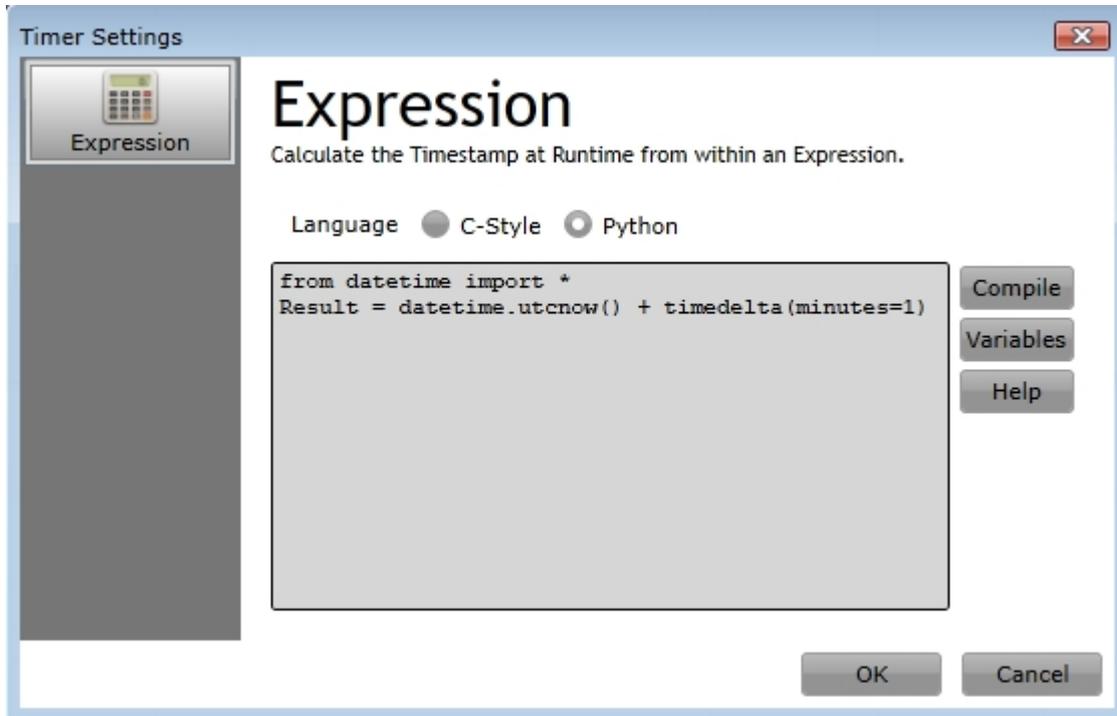
Some workflows are supposed to start at a defined time, possibly recurring. An example could be a monthly meeting. The Timer Start Event enables the automatic start of a workflow at the specified time.

Select the Timer Start Event as the start of the process.

Navigate to the property set of the Timer Start Event and click on the "Edit" button of the timer property. The timer settings will be displayed now.



Insert an expression to determine when the workflow should start and return the calculated date time value as "Result". The result must be a value in UTC (Universal Time Coordinated) format. Otherwise timezone related delays might appear while triggering timer events.



Click Compile and then Execute to check if there are errors in the expression. Click OK and the calculated time will be considered in the workflow.

Select "Recurring" in the properties to have the workflow start recurringly and not just once.

## Run the Workflow

To run a workflow by timer, proceed as described in "[Run Workflow from Project](#)".

The workflow can now be found in the list of scheduled workflows. Open the [Repository](#) and navigate to the [Instances](#) node. Click on "Technical Options" and choose "Show Queues".

Id	Name	User	Scheduled	Repeat	RepeatUnit	State
87699	Scheduled Workflow #116766	kristina.stockhaus	Thursday, November 29, 2018 1:21:01 PM	0	Off	Hold

## Stop Workflow marked as Recurring

To stop a workflow that has "Recurring" selected in the properties, press the "Run" button again (same type of version). Then press the "Stop" button in the Stop Workflow Dialog.



## Monitor Process Instance

After a process instance has been created it can be monitored.

Monitoring a process instance gives answers to the following questions:

- Which Workflow is a process instance assigned to
- Which Tasks are outstanding respectively have been completed
- What is the progress in the associated diagram
- When and in what order were which activities executed
- How is the performance of the current process instance compared to other process instances
- Were there errors or warnings
- Which attributes with which values are in the current global scope.

All these information can be obtained by the Process Instance Details dialog. There are several ways to access the Process Instance Details dialog:

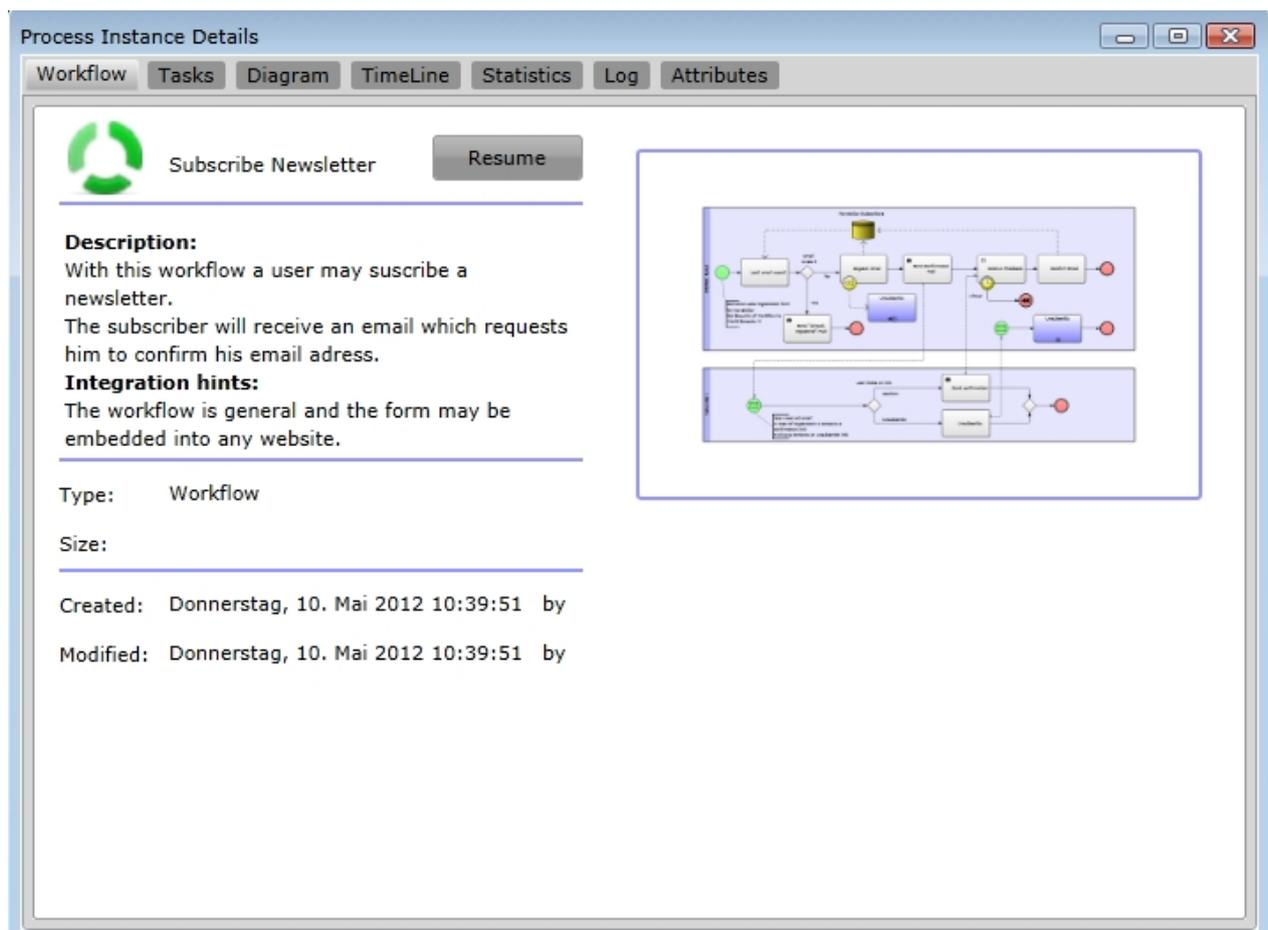
First of all you can choose between the Repository and the Portal as entry point. Then you can either

- navigate from the [dashboard](#) onwards
  - directly double-click on one of the displayed process instances in the running process instances list or press the "more" button below the running process instances list --> double-click on the desired process instance
  - press the "more" button below the pending tasks list --> click on the blue dot of the desired task on the far right of the task list
- expand the [Workflow node](#)
  - click on the Instances node and double-click on the desired process instance
  - click on the Tasks node and then on the blue dot of the desired task on the far right of the task list

The Process Instance Details dialog provides the following tabs:

## Workflow

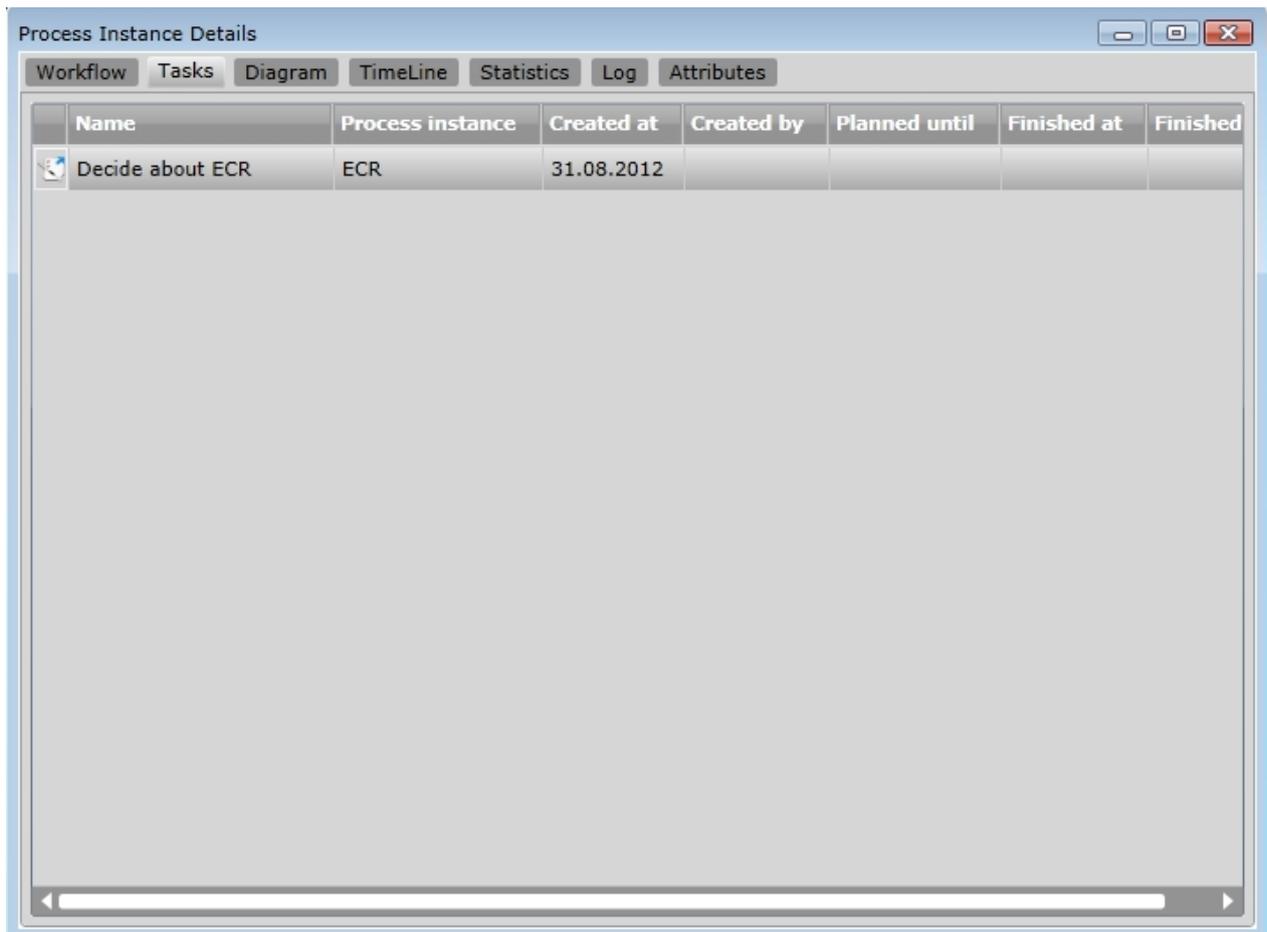
This tab shows basic information according the associated workflow of the process instance.



The screenshot shows a window titled "Process Instance Details" with several tabs: Workflow, Tasks, Diagram, TimeLine, Statistics, Log, and Attributes. The "Workflow" tab is selected. On the left, there is a green circular icon and a "Resume" button. Below this, the "Description" states: "With this workflow a user may suscribe a newsletter. The subscriber will receive an email which requests him to confirm his email adress." The "Integration hints" section says: "The workflow is general and the form may be embedded into any website." Below this, it shows "Type: Workflow" and "Size:". At the bottom, it lists "Created: Donnerstag, 10. Mai 2012 10:39:51 by" and "Modified: Donnerstag, 10. Mai 2012 10:39:51 by". On the right, there is a large diagram showing a complex workflow with various nodes and transitions.

## Tasks

This tab shows all current and already completed tasks performed by the process instance. To open a task, double-click on the row.

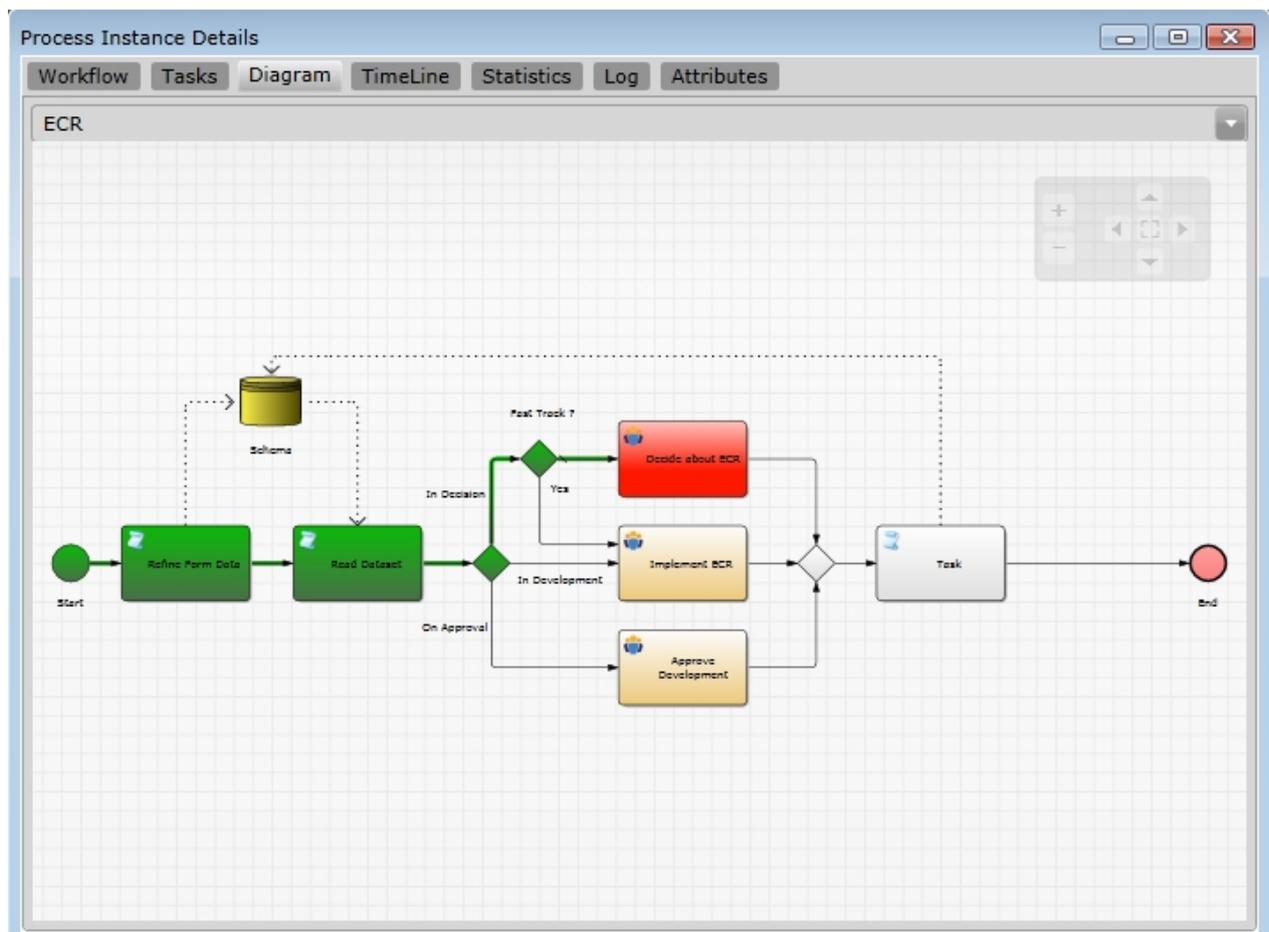


The screenshot shows a window titled "Process Instance Details" with several tabs: Workflow, Tasks, Diagram, TimeLine, Statistics, Log, and Attributes. The "Tasks" tab is active, displaying a table with the following data:

Name	Process instance	Created at	Created by	Planned until	Finished at	Finished
 Decide about ECR	ECR	31.08.2012				

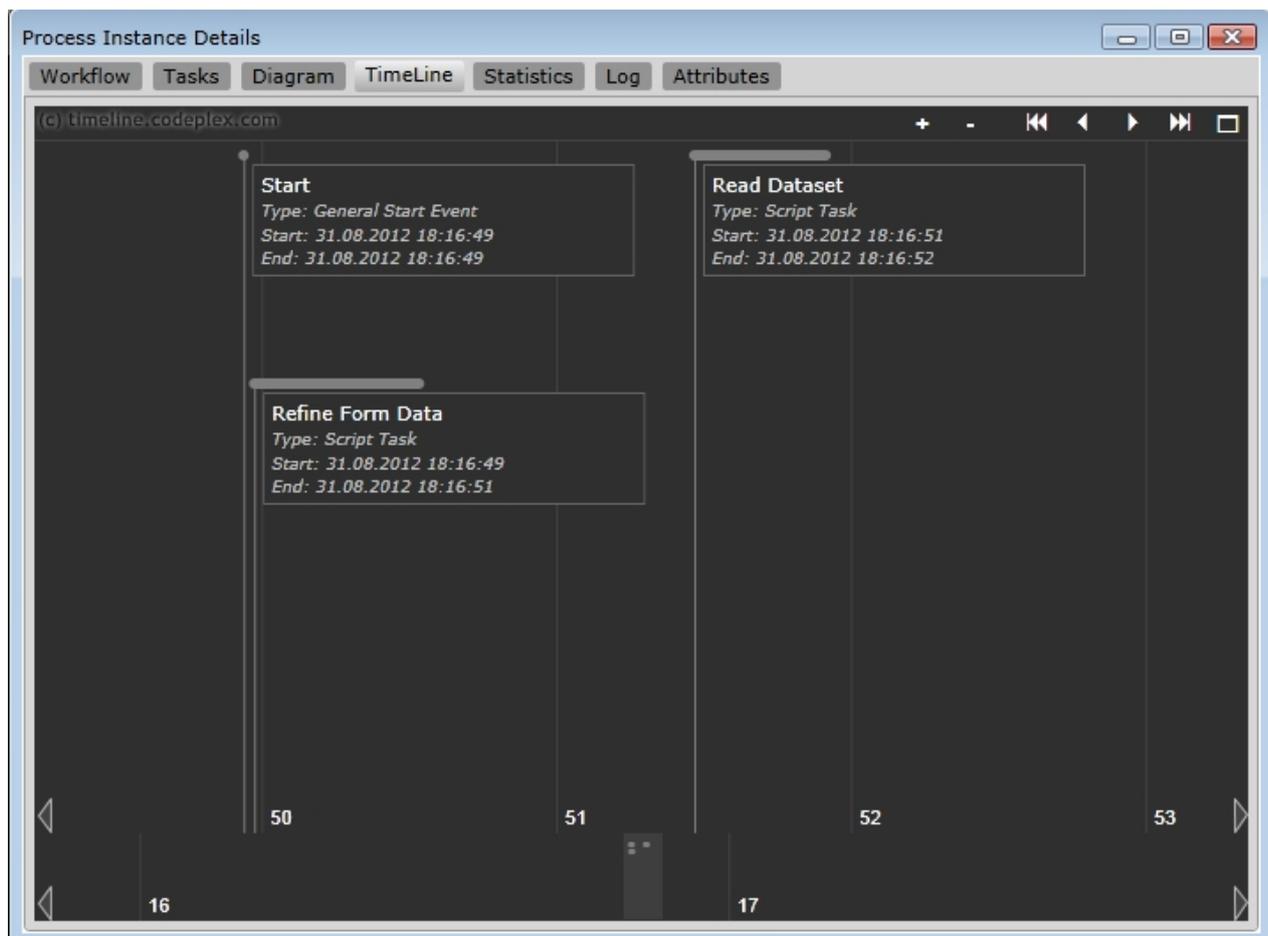
## Diagram

This tab shows the process along with its progress. Elements that have been passed (or completed Activities) are colored green, the current activities are colored red.



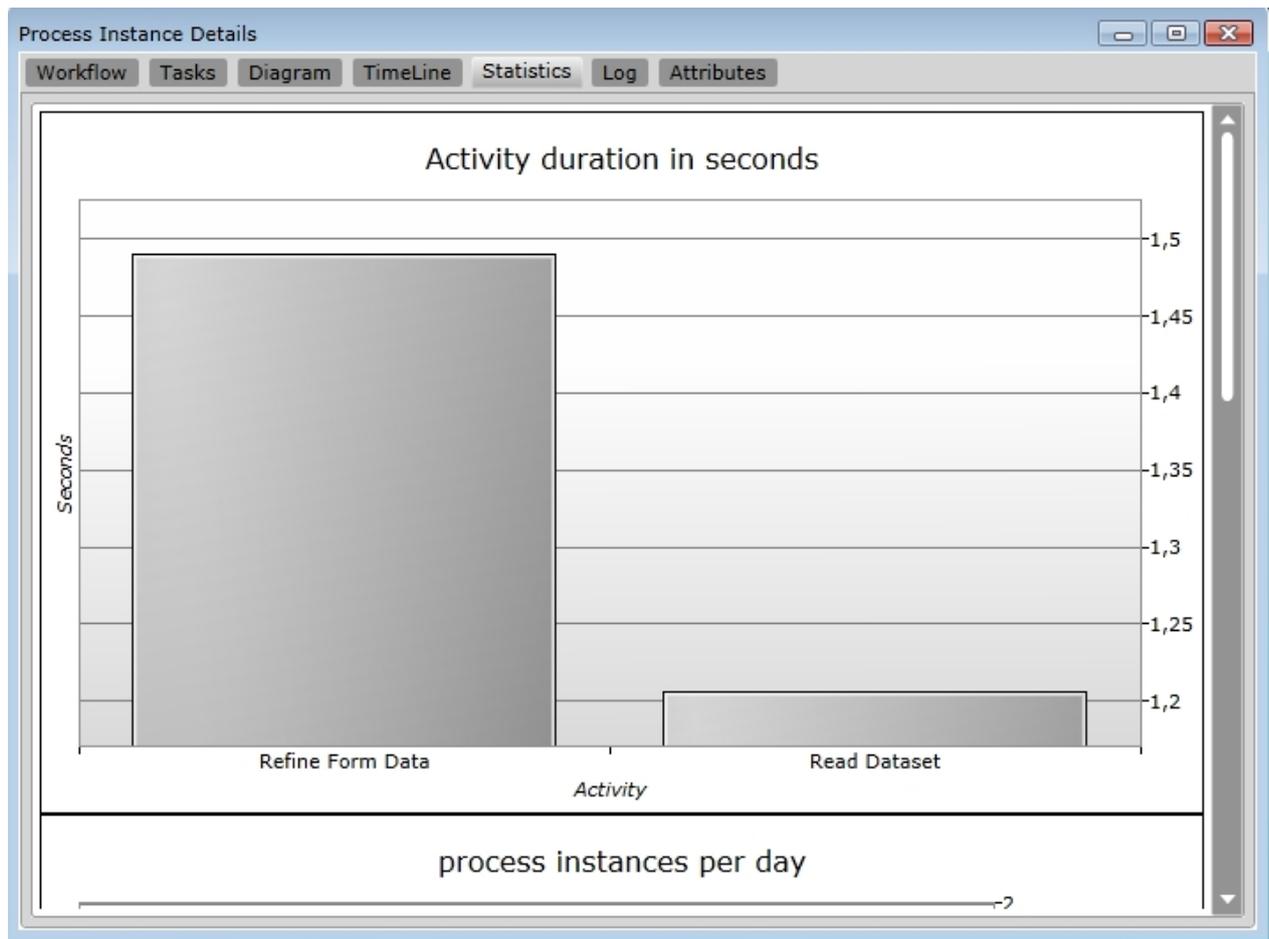
## Timeline

This tab shows the executed activities on a timeline with start/end and duration information.



## Statistics

This tab shows statistics of the current process instance as well as comparisons to previous process instances.



## Log

This tab shows detailed technical information about the execution of the workflow.

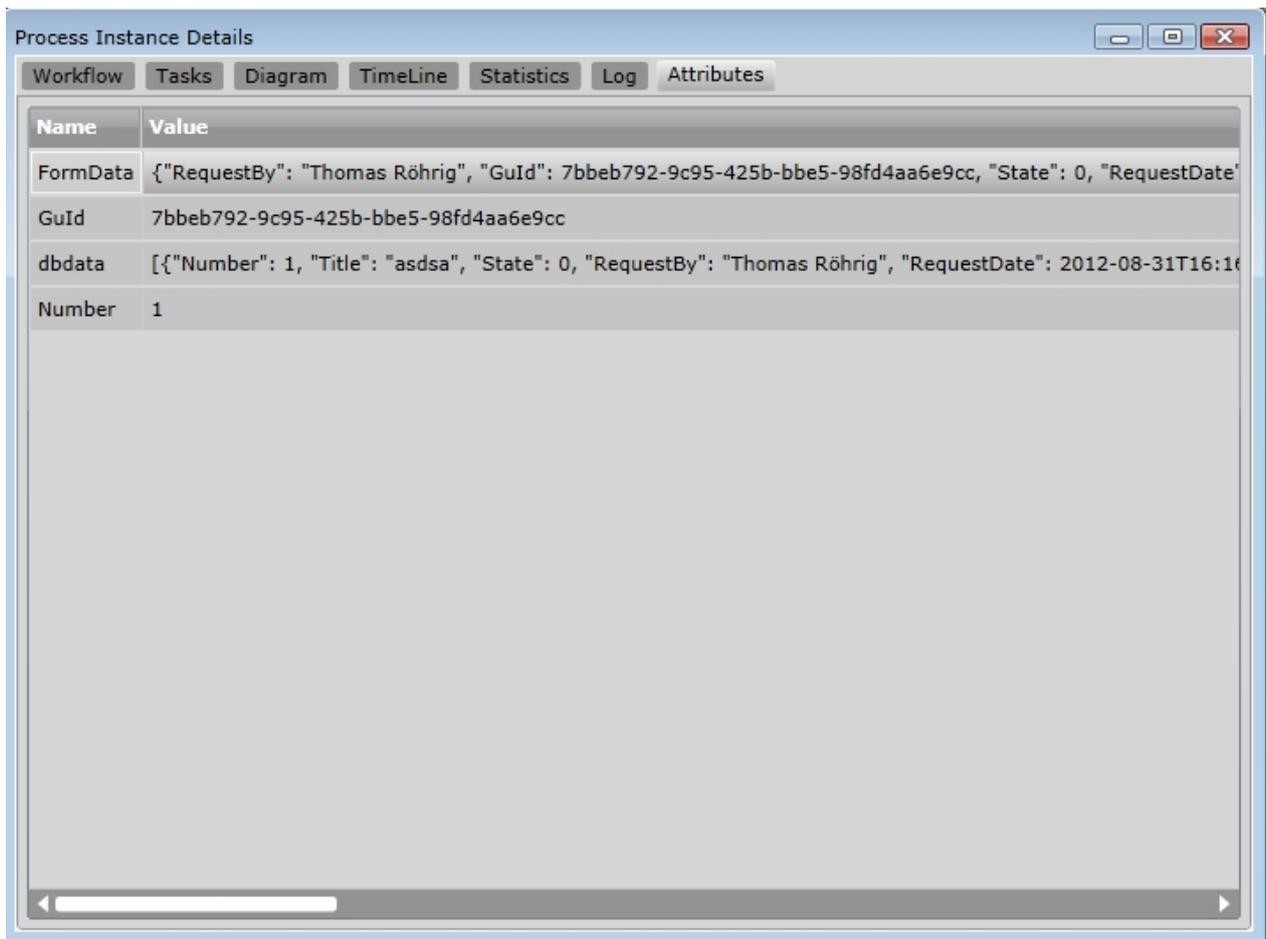
Process Instance Details

Workflow Tasks Diagram TimeLine Statistics Log Attributes

Time	Action	Control	Control Type	Control Id
8/31/2012 6:16:49 PM	Process instance execution triggered			0
8/31/2012 6:16:49 PM	Token created	Start	General Start Event	1
8/31/2012 6:16:49 PM	Token moves		Normal Sequence Flow	8
8/31/2012 6:16:49 PM	Token moved	Refine Form Data	Script Task	7
8/31/2012 6:16:51 PM	Token moves		Normal Sequence Flow	14
8/31/2012 6:16:51 PM	Token moved	Read Dataset	Script Task	13
8/31/2012 6:16:52 PM	Token moves		Normal Sequence Flow	24
8/31/2012 6:16:52 PM	Token moved		Exclusive Gateway	23
8/31/2012 6:16:56 PM	Token moves	In Decision	Normal Sequence Flow	42
8/31/2012 6:16:56 PM	Token moved	Fast Track ?	Exclusive Gateway	41
8/31/2012 6:16:58 PM	Token moves		Default Sequence Flow	48
8/31/2012 6:16:58 PM	Token moved	Decide about ECR	User Task	26
8/31/2012 6:16:58 PM	There is no message body defined	Decide about ECR	User Task	26
8/31/2012 6:16:58 PM	Process instance execution ended			0

## Attributes

This tab lists the process instance attributes of the global scope and their current values.



## View the process instance status outside IYOPRO

To view the status of a process instance outside IYOPRO, use one of the following URLs:

- `<IYOPRO WebClient>/wfstate/<process instance guid>`
- `<IYOPRO WebClient>/wfstate/<process instance id>`

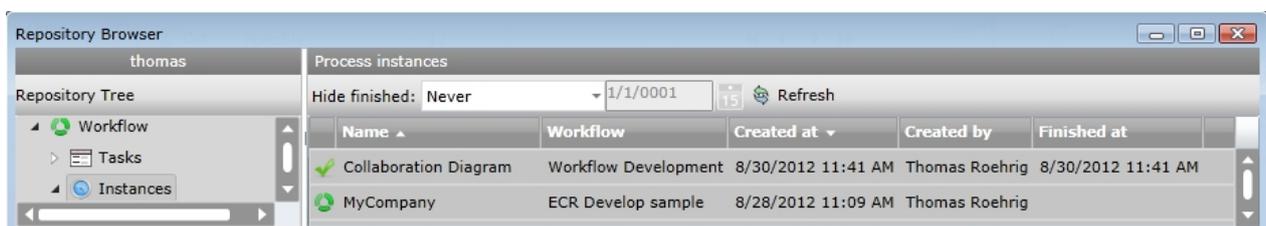
Example: `http://portal.iyopro.com/index.html#/wfstate/b6ac37b1-e1a5-42ac-8410-61ca0d827de6`

## Inspect finished Process Instances

To inspect finished process instances it may be important to control the time stamp of suppression.

There are two ways to access finished process instances: the [Repository](#) or the [Portal](#)

Open the Repository or Portal and expand the Workflow node. Click on the "Instances" node to receive a list of all available process instances.



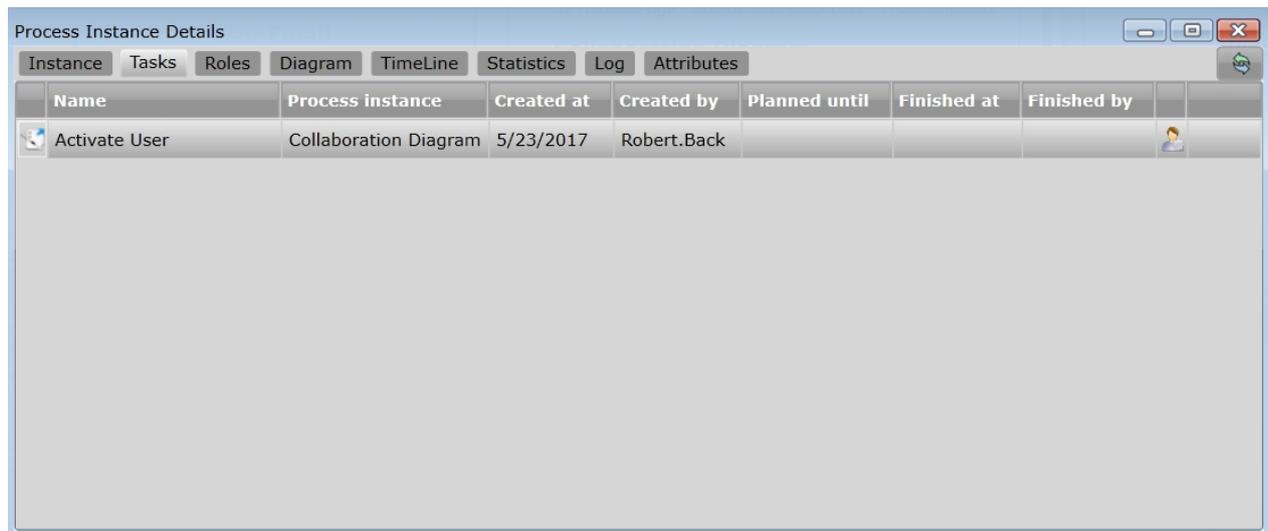
The Process Instances plug in provides the "Hide finished" setting in its toolbar where you can control the timespan of suppression.

## Configure Participants

Participants of workflow tasks can be configured during the run time of a process [instance](#). Only team administrators can add and remove task participants.

## Navigate to the Participant Configurator

The configuration dialog can be found on the tasks tab in the process instance details. The configurator opens after clicking on the user icon of the specific task.

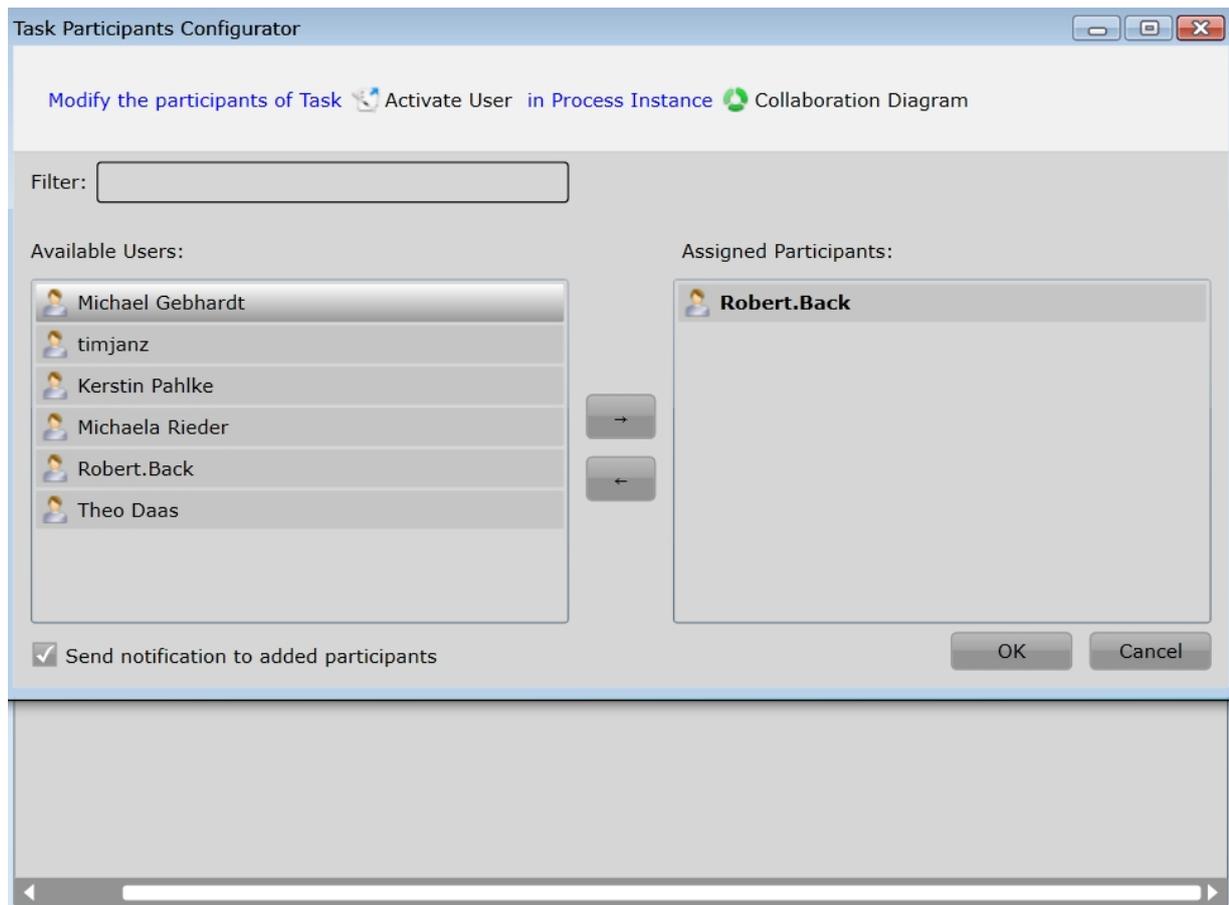


## Configure Participants of Workflow Tasks

The configurator consists of two lists. The left one shows all the available users of the team which can participate the specific tasks. The right list shows the currently assigned participating users of the task.

One or more list entries have to be selected. Afterwards these users can be moved to the opposite list by pressing the corresponding direction button in the middle of both lists.

Check the box "Send notification to added participants" to send the task notification to every added participant via email.



### Additional information

In the lists, the name of the account you are signed in with is shown in bold.

In IYOPRO Web you can also see whether and who has already accepted a task. The respective user has both a green circle with a tick in the icon and "accepted" next to the name.



### Best Practice

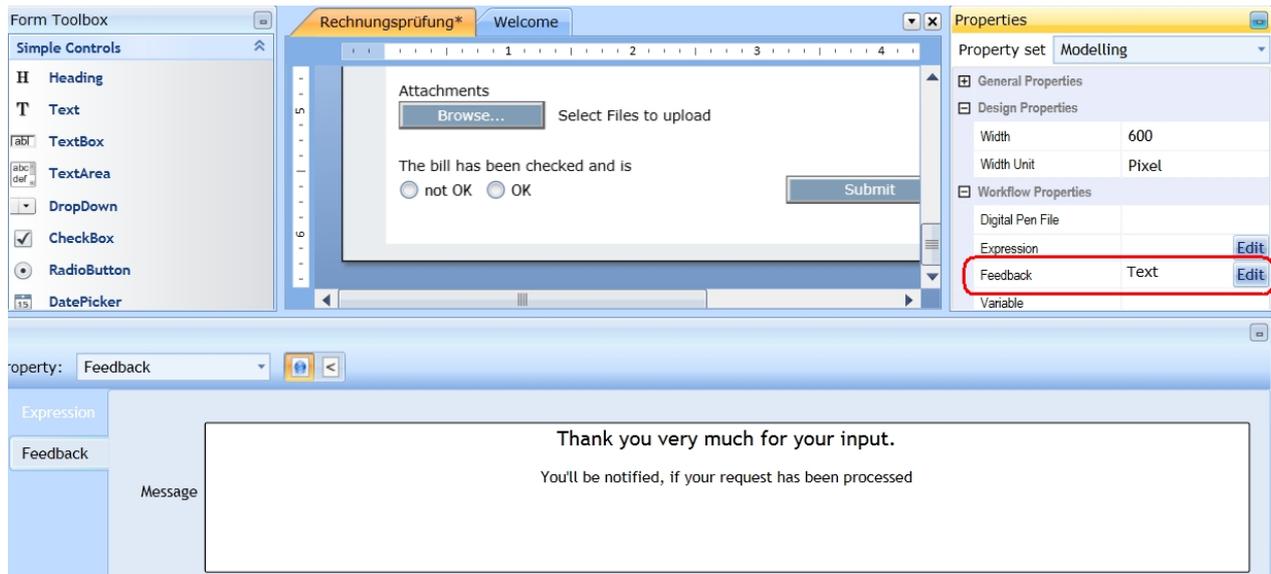
This section provides information about common scenarios and useful tips.

#### Feedback after Form Submission

When a user has processed a form you may notify him with a feedback. This could be something like:

[Thank you for your input, you'll be notified, if your request has been processed.](#)

To do so activate the properties of the Form and click on the Edit-Button of the Feedback property. You may then edit the feedback in the advanced property dialog.



It is also possible to include information from the workflow in the feedback. Please refer to [Referencing data in Notifications](#) for the details.

### Attach Message Event to Hyperlink

Nearly every workflow uses the email subsystem to inform users about outstanding tasks or actions that have been done. Sometimes it's useful to embed one or more hyperlinks in the email to allow the receiver to quickly respond to this mail or take a decision.

For example, if your email contains a newsletter you may embed a hyperlink to allow the receiver to unsubscribe from the subscription.

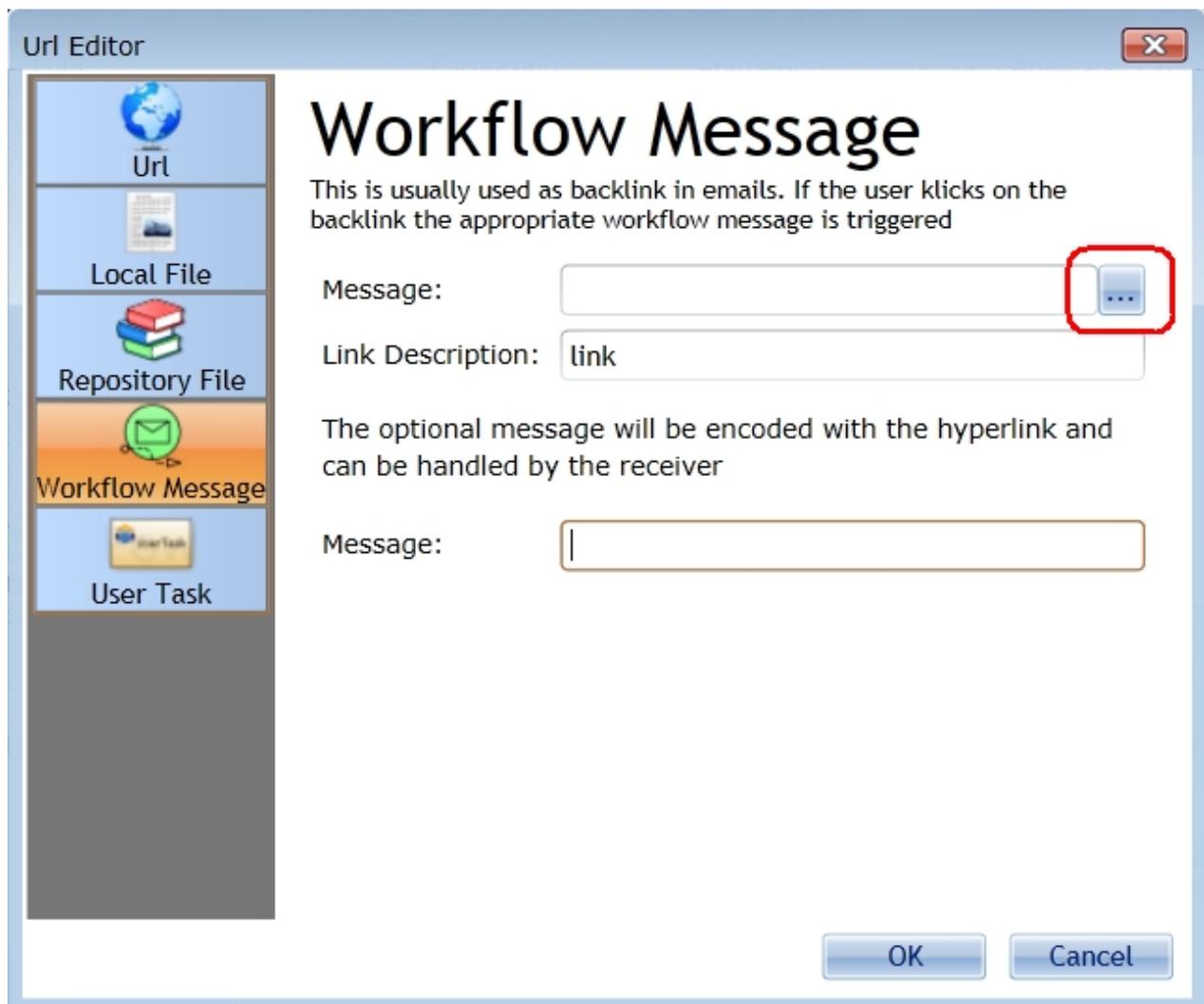
To do so, edit the email message of the sending task and mark the text which will become a hyperlink



Click on the "Insert Link" icon of the Font Ribbon Group to open the Url-Editor



In the Url-Editor activate the "Workflow Message" Option.



Click on the "... " Button to select the destination. The following window appears:



You may now select a message flow or any receive message event.

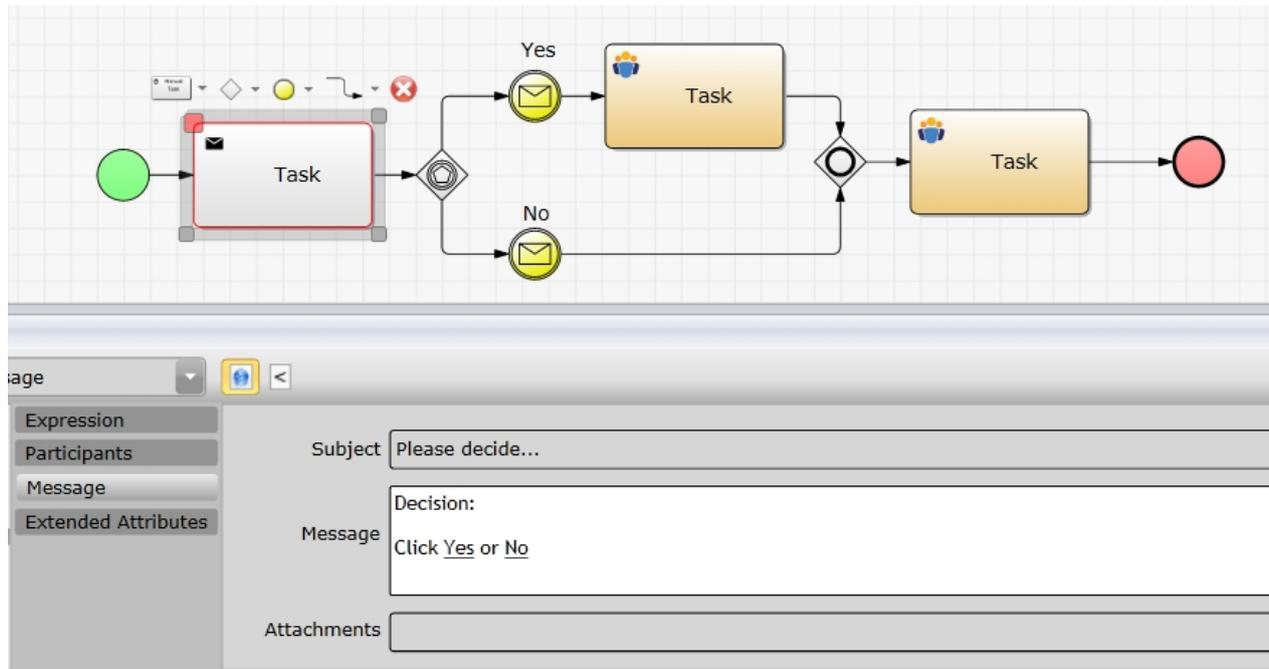


Caution: If you have chosen a message flow which spans over at least one none executing pool, you must activate the executable property of that message flow. If you don't, the targeted object will not wait for that element.

This event will then be assigned to the selected hyperlink.

In the case, the email receiver clicks on the hyperlink, the specified message will be triggered. If the target has been a receive message start event, a new process instance will be created.

Another common practice is to control the flow by an event based gateway. See the following model:



With the above method you can link the Yes/No hyperlinks to the intermediate events which will then trigger the flow.

### Attach Email Notification to a User Task

Typically when a user task is triggered the participants should be notified that the new task requires their attention.

This is achieved by composing a notification email in the [message editor](#).

In addition to static content the message editor is also capable of inserting dynamic information from the current process.

This is achieved with a special syntax to reference the desired information:

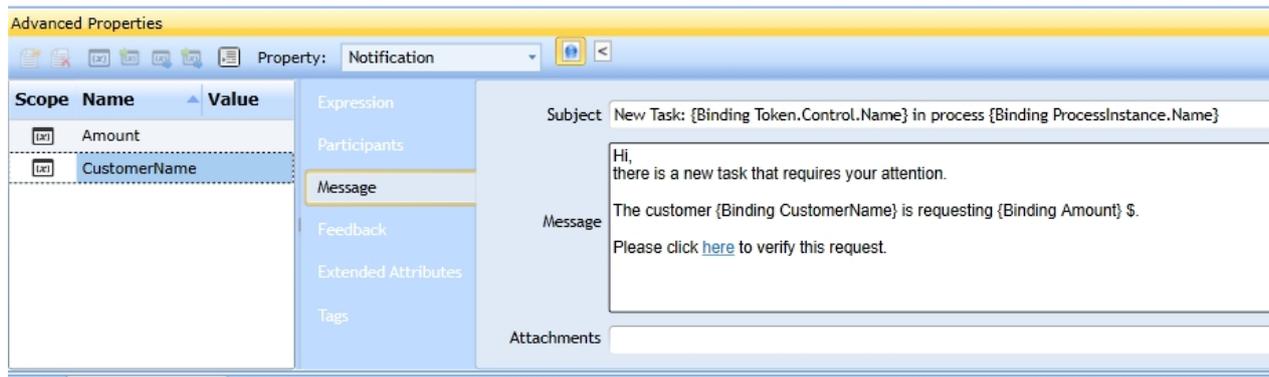
**{Binding Object}**

The "Object" is one of the predefined objects available or a workflow attribute of the process.

The predefined objects available are:

- [Session](#)
- [ProcessInstance](#)
- [Token](#)
- [Database](#)
- [Workflow](#)

A sample email notification could look like this (for a detailed description please refer to: [Referencing data in Notifications](#))



## Referencing data in Notifications

When sending an email notification to a user or displaying a form feedback it is possible to include information from the executed process instance.

### Basic Syntax

The basic syntax to include process instance information is:

**{Binding VariableName}**

Here the *VariableName* is a valid workflow variable. Examples are:

- {Binding FormData["Text"]}
 

Retrieve the content of the entry "Text" from the dictionary FormData. This could for example be a text field in a form used in the workflow
- {Binding ProcessInstance.Created.Name}
 

Get the display name of the person who has started this process instance.

In addition there is the possibility to use extra information in the context of an email notification. This extra information is referenced with:

**{Context Keyword}**

The available *Keywords* are Name, EEmail and ErrorMessage.

The Name and EMail keyword provide the respective attributes of the user receiving the particular notification.

The keyword ErrorMessage can be used in the context of a workflow failure notification only.

In this situation ErrorMessage contains the error message that stopped the process instance from continuing.

### Extended Syntax

While an information might be available in a workflow variable, it might not have a suitable format to be used in an email notification or form feedback.

To avoid additional script activities for the sole purpose of data conversion, the extended syntax provides an option to specify desired conversions.

When using the extended syntax the *VariableName* or *Keyword* is followed by a comma and the specification of the conversion.

**{Binding VariableName, Convert=ConverterSpecification}**

**{Context Keyword, Convert=ConverterSpecification }**

A converter specification is the name of the converter with optional converter parameters in parenthesis. Multiple converters can be concatenated with a vertical bar ("pipe character").

Examples:

- **{Binding ProcessInstance.Created.Name, Convert=ToHTML}**  
Ensure the user name is valid in a HTML context. No options are provided to the convert.
- **{Binding ProcessInstance.Created.Name, Convert=ToHTML()}**  
The same as above.
- **{Binding ProcessInstance.Created.Name, Convert=ToHTML(True)}**  
Same as above, but an option is provided to not convert spaces.
- **{Binding MyDate, Convert=ToTZ(„Europe/Berlin“) | Format(„dd.mm.yyyy“) }**  
Concatenation of two converters. The first one takes a datetime object with UTC time zone and converts it to the time zone for Berlin. This time stamp is then formatted to a string comprising day, month and year.

**Note:**

When concatenating converters care has to be taken regarding the supported input and output types. So while it is ok to convert the time zone of a time stamp using ToTz(...) first and convert the time using Format(...) to a string next, it will not work the other way around.

**Available Converters**

**No – prevent automated HTML conversion**

The automated conversion to valid HTML might not be desired in some cases.

For example, if the content of the workflow variable already contains HTML to be inserted in the notification.

Specifying this converter disables the automated HTML conversion.

No		
Input	Output	Parameter
Object	Object	No parameters

**ToHTML – manual HTML conversion**

This converter also prevents the automatic HTML conversion and allows you to specify the conversion options to be used.

ToHTML		
Input	Output	Parameter
String	String	<ul style="list-style-type: none"> <li>• Ignore white spaces - Bool (optional, default false): Do not convert spaces to "&amp;nbsp;".</li> <li>• Ignore Line Breaks - Bool (optional, default false): Do not convert line breaks to "&lt;br /&gt;".</li> <li>• Spaces for tab - Int (optional, default 4): Number of spaces to replace a tab character with</li> </ul>

**ToTZ – Time zone conversion for time stamps**

Typically all workflow timestamps are set as time zone UTC. When being displayed in the GUI they are converted to the current time zone of the user. To perform this conversion also within notifications, one

can use this converter.

**Note:**

When trying to specify a time zone one faces a number of issues. The first thought might be to provide simply the desired time zone abbreviation like UTC or CEST. Unfortunately these are not unique. For example AST denoted the Arabia Standard Time as well as the Atlantic Standard Time. Furthermore these two have different offsets from UTC. As an alternative one might consider country names, but this will fail for any country spanning across more than one time zone. As the IYOPRO Backend is executed on a windows server, one might consider the windows time zone names. However these are fairly long and one might not be sure which one to choose (e.g. W. Central Africa Standard Time versus South Africa Standard Time).

Luckily the Unicode project provides a mapping of Region/City texts to windows time zones. See <http://unicode.org/cldr/trac/browser/tags/release-30-0-3/common/supplemental/windowsZones.xml> for details.

ToTZ		
Input	Output	Parameter
Python datetime or C# DateTime	Python datetime	Conversion form UTC to another time zone: <ul style="list-style-type: none"><li>• Target location - String: Specification of the target location for the conversion e.g. „Europe/Berlin“</li></ul> Conversion from one time zone to another <ul style="list-style-type: none"><li>• Source location - String: Specification of source location of the conversion e.g. „America/Los Angeles“</li><li>• Target location - String Specification of the target location for the conversion e.g. „Europe/Berlin“</li></ul>

### Format – Generic object formatting

This converter allows to use the C# string.Format(...) capability to convert the data. See [https://msdn.microsoft.com/de-DE/library/26etazsy\(v=vs.110\).aspx](https://msdn.microsoft.com/de-DE/library/26etazsy(v=vs.110).aspx) for details.

**Note:**

The C# string.Format(...) method is capable of converting multiple values. This is achieved by referencing the values with {0:...}, {1:...} etc.

In the context of this converter there is however only one value that will be converted. Therefore the converter adds the required "{0:" and "}" automatically.

Format		
Input	Output	Parameter
Object	String	<ul style="list-style-type: none"><li>• Format string - String: Depends on the input. For example a float value can be converted to a string with two decimals and a separator for the thousands using the format string "#,###.00".</li><li>• Culture - String (optional, default Invariant): Key for the culture to be used for the formatting e.g. „de-DE“</li></ul>

### Custom conversions by python scripts

With this converter one can implement custom python scripts to perform the required conversion. Here is an example for a converter to make a string upper case:

```
def UC(v):  
    return unicode(v).upper()
```

Usage: **{Binding Var, Convert=Script("ScriptFileName","UC")}**

**Note:**

To allow the Script converter to access the required script file, the file has to be part of the workflow project.

This converter can't be used in the feedback of a process instance start form, as there is no process instance created yet, when the feedback is displayed.

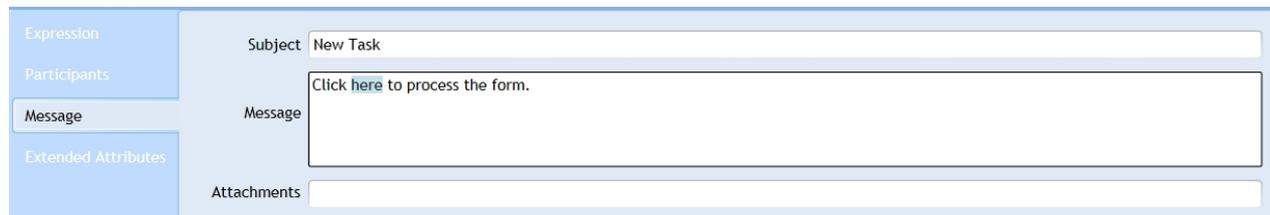
Script		
Input	Output	Parameter
Object	Object	<ul style="list-style-type: none"><li>• Script file name - String: Name of the script file in the project containing the conversion function.</li><li>• Function name - String: Name of the conversion function in the script file.</li></ul>

### Attach User Task to Hyperlink

Nearly every workflow uses the email subsystem to inform users about outstanding tasks that have to be done.

Sometimes it's useful to embed a Hyperlink in the notification mail to allow the receiver to quickly process the user task.

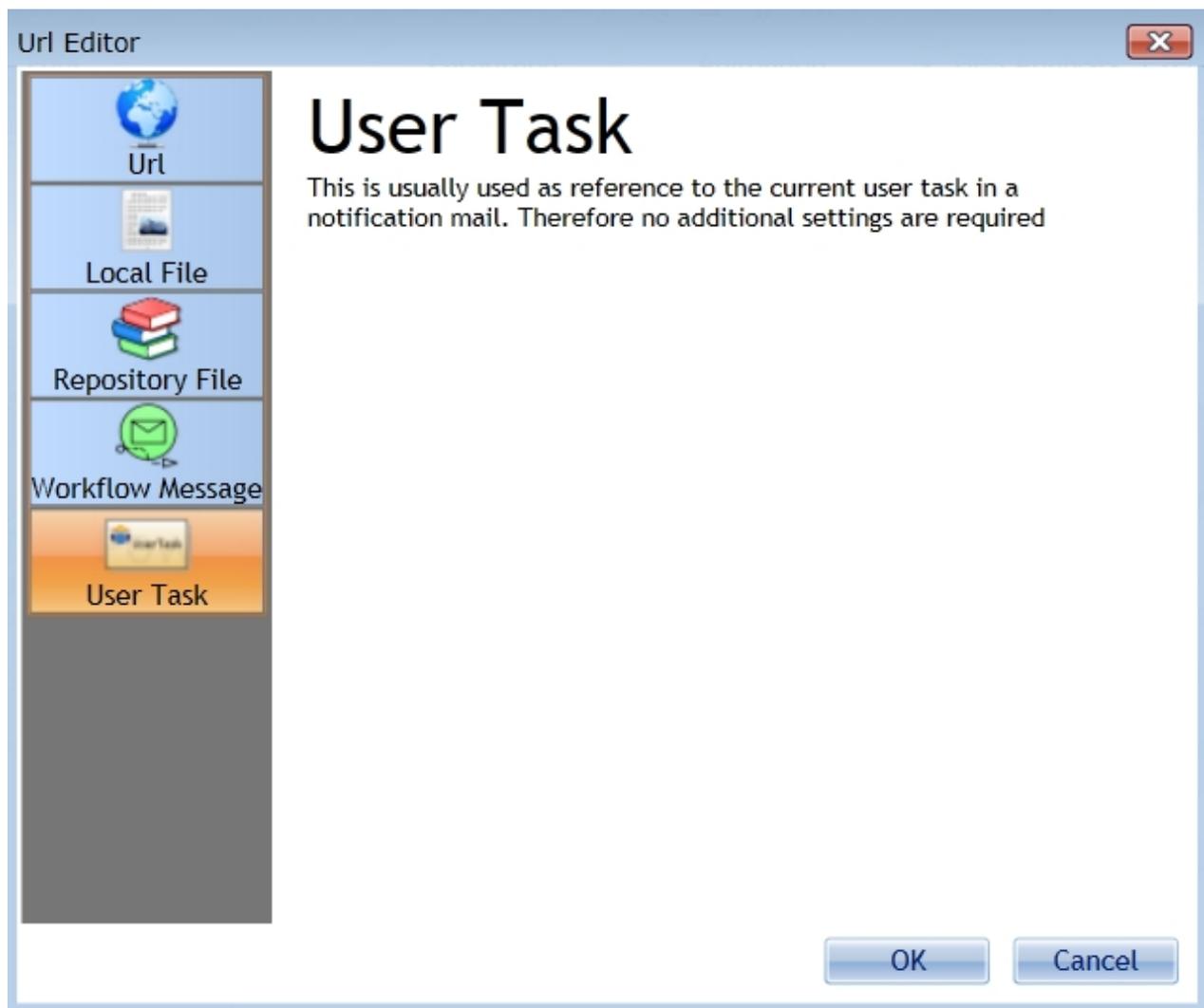
To do so, edit the email message of the sending task and mark the text which will become a hyperlink



Click on the "Insert Link" icon of the Font Ribbon Group to open the Url-Editor



In the Url-Editor activate the "User Task" Option.



Click OK and the task will be assigned to the selected Hyperlink.

In the case, the email receiver clicks on the Hyperlink, the specified task will be opened (i.e. the form will be displayed).

### Trigger message programmatically

It's often required to send workflow messages to IYOPRO process instances. To do that IYOPRO offers three mechanisms:

1. Send Workflow Message using the IYOPRO API
2. Send Workflow Message by Url
3. Send Workflow Message by Web Service

Possible [results](#) when sending a workflow message can be found at the bottom of this page.

### Send Workflow Message using the IYOPRO API

This method is described in the [IYOPRO API section](#).

## Send Workflow Message by Url

To send a Workflow Message by Url, IYOPRO provides the IYOPROSendWorkflowMessage.aspx component. You may specify the following sets of parameters to complete the message:

The target may be specified in three different ways:

### To new process instance

This mode is used to create a new process instance which immediately receives the specified message

You may specify the following url parameters:

projectid	Specify the id of the workflow project, to which the message will be send
diagramid	Specify the id of the workflow diagram of the project, to which the message will be send
ctrlid	Specify the id of the diagram control in the diagram, to which the message will be send
message	This is an optional, base64 encoded message which may be read by the workflow instance
backlink	This is an optional, base64 encoded email address which may be read by the workflow instance

Sample:

```
https://www.iyopro.com/iyopro/IYOPROSendWorkflowMessage.aspx?ProjectId=56443&DiagramId=9897&CtrlId=60
```

### To running process instance

This mode is used to send a workflow message to a known process instance

You may specify the following url parameters:

instance	The guid of the targeted process instance.
diagramid	Specify the id of the workflow diagram, to which the message will be send
ctrlid	Specify the id of the diagram control in the diagram, to which the message will be send
message	Optional, base64 encoded message which may be read by the workflow instance
backlink	Optional, base64 encoded email address which may be read by the workflow instance

Sample:

```
https://www.iyopro.com/iyopro/IYOPROSendWorkflowMessage.aspx?instance=1025dee0-4722-4121-914b-0c0425379202&DiagramId=9897&CtrlId=60
```

## Determine existing process instance

This mode is used to determine a running process instance by specifying a process instance attribute which matches a given value.

You may specify the following url parameters:

diagramid	Specify the id of the workflow diagram of the project, to which the message will be send
ctrlid	Specify the id of the diagram control in the diagram, to which the message will be send
message	This is an optional, base64 encoded message which may be read by the workflow instance
backlink	This is an optional, base64 encoded email address which may be read by the workflow instance
instancename	Specify the name of the process instance. All running instances with the given name will be searched
instanceattribu tename	Specify the name of the process instance attribute
instanceattribu tevalue	Specify the value, which must match to the instance attribute.

Sample:

```
https://www.iyopro.com/iyopro/IYOPROSendWorkflowMessage.aspx?  
DiagramId=9897&ControlId=60&InstanceName=Orders&InstanceAttributeName=OrderNo&  
InstanceAttributeValue=4711
```

## Send Workflow Message by Web Service

To send a Workflow Message by Url, IYOPRO provides the IYOPROSendWorkflowMessage.asmx component. You may specify the following sets of parameters to complete the message:

projectid	Specify the id of the workflow project, to which the message will be send
diagramid	Specify the id of the workflow diagram of the project, to which the message will be send
controlid	Specify the id of the diagram control in the diagram, to which the message will be send
message	This is an optional message which may be read by the workflow instance
backlink	This is an optional, base64 encoded email address which may be read by the workflow instance

instancename Specify the name of the process instance. All running instances with the given name will be searched

instanceattribu Specify the name of the process instance attribute

tename

instanceattribu Specify the value, which must match to the instance attribute.

tevalue

The meaning of the parameters are the same as when specified by the URL-Version of this component.

## Results when sending a workflow message

Depending on the system configuration IYOPRO responds with a HTML page containing a description of the call result or a structured result information.

The structured result information comprises of the HTTP status code and description, as well as the custom **X-IYOPRO-STATE** reply header.

If the call was successful the HTTP state 200 (OK) is returned. The X-IYOPRO-STATE header will contain one of the following values:

RESUMED The execution of the process instance was resumed

STARTED A new process instance was started

PROCESSED XML message was processed

If the call fails, it can either fail with status code 400 (bad request) or internal server error (500) in case of an exception during the request processing.

The X-IYOPRO-STATE header will contain one of the following values:

NOTUNIQUE The specified process instance is not unique

NOTFOUND.W No matching workflow found or the workflow is not available or was removed

ORKFLOW

NOTFOUND.I The appropriate process instance is not available

INSTANCE

NOTFOUND.D The appropriate process diagram could not be determined

IAGRAM

NOTFOUND.C The appropriate process element could not be determined

ONTROL

NOTFOUND.F The appropriate process instance has already been finished

FINISHED

NOTFOUND.M The specified link does not reference a message flow

MESSAGEFLOW

ERROR.CONT XML body invalid

ENTFORMAT

ERROR.MESS Invalid XML message data

AGEDATA

If the call is successful or if the X-IYOPRO-STATE contains the value "NOTFOUND.FINISHED", the structured result information additionally contains the **X-IYOPRO-INSTANCEID** and **X-IYOPRO-INSTANCEGUID** reply header.

The X-IYOPRO-INSTANCEID will contain the process instance ID and the X-IYOPRO-INSTANCEGUID will contain the process instance GUID.

## Start Workflow outside IYOPRO

An IYOPRO Workflow may be started by two methods:

1. A User completes a Form which is defined as workflow startup form
2. An event triggers a Receive Message Start Event

## Start Workflow by Form

To start a workflow by form, ensure that the workflow has been [build](#) and the desired form has been marked for [workflow startup](#).

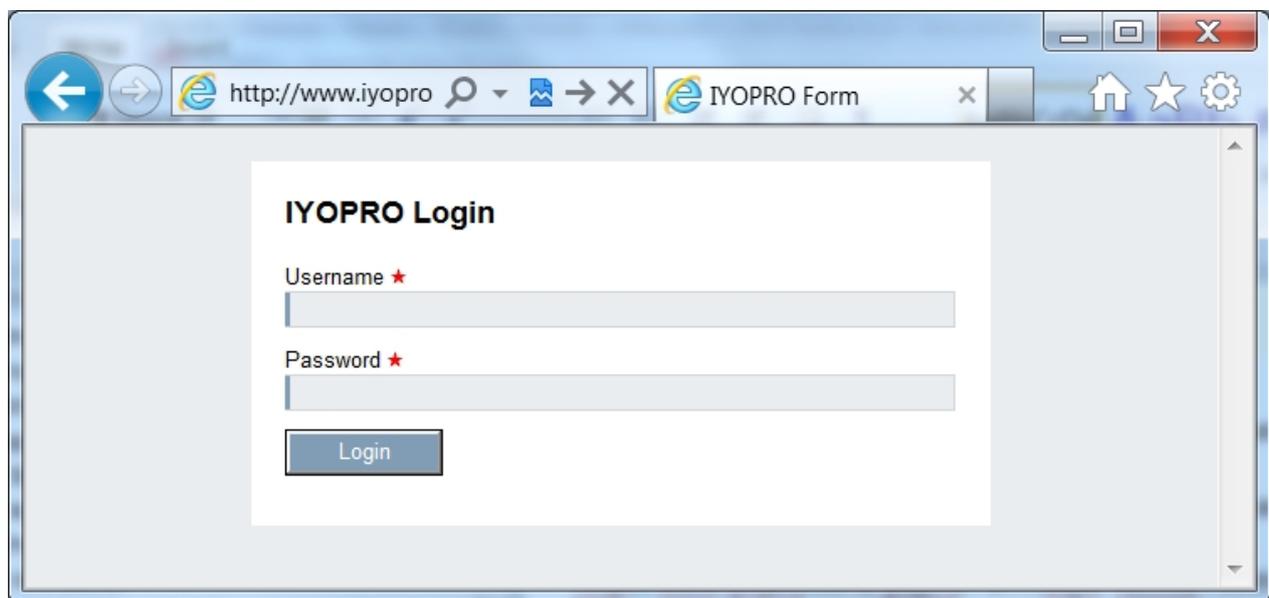
Then [run the workflow](#) out of IYOPRO. The form will appear. Copy the address line of the browser window to the clipboard. It should be something like this:

```
http://www.iyopro.com/iyopro/Form.aspx?formid=9999&sessionid=1bc489c7-39c4-48a4-9be4-8006dfd6700a&taskid=00000000-0000-0000-0000-000000000000&instanceid=4fb660a5-719c-4486-8a93-ccec711bae8e&wfid=11206
```

Paste the content into a notepad editor. Remove all tags after the question mark (?) except the wfid=xxxx entry. This entry specifies the target workflow id in IYOPRO. The result should look like this:

```
https://www.iyopro.com/iyopro/Form.aspx?wfid=11206
```

You may use this URL and embed it wherever you wish (i.e. as Bookmark in your browser). Whenever you activate the URL, you will be requested to log in to IYOPRO. Afterwards the requested form will be displayed.



If you want to work with an embedded form, you can just add "&embedded=true" to the URL.

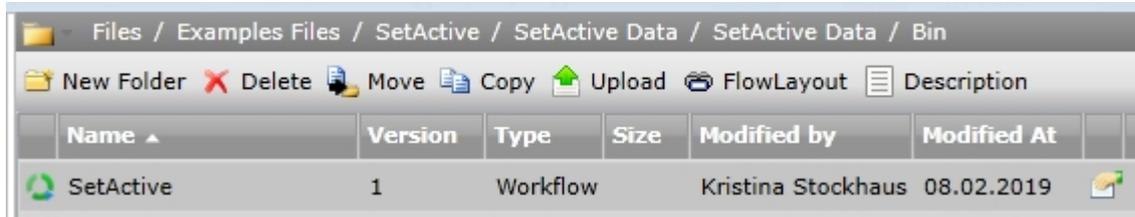
Attention: The URL redirects to the active version of a workflow upon activation, should the wfid used belong to a non-active version. If you want to start a non-active version you have to add "&forceVersion=true" to the URL.

## Start Workflow by Form - independent of the different versions

To start a workflow regardless of the built versions, the same short URL applies as above. The difference lies in the numerical value of the "wfid". Instead of using the version-specific Workflow ID, the numeric value of

the "wfid" is replaced by that of the Workflow File ID. The Workflow File ID is the same for each version. Activation of the URL will now always start the version that has been set as [Active Version](#).

The Workflow File ID can either be found in the corresponding column displayed in the Instance node of the Repository (if an instance of the workflow already exists) or in the File Properties of the "Bin" file of the workflow:



Name	Version	Type	Size	Modified by	Modified At
SetActive	1	Workflow		Kristina Stockhaus	08.02.2019

### Start Workflow by Form and provide initial field values

As described in the section above a workflow with a start form can be invoked by an URL. If desired initial values for form fields can be provided via the additional URL parameter "userdata".

The form initialization is performed in the same way as if these values were set in a from load expression.

As there can be multiple fields in the form that should be initialized the initial values need to be provided as a python dictionary.

To enable this data structure to be transported as part of the URL the dictionary needs to be base64 encoded.

The base64 encoded dictionary is then appended to the URL as parameter userdata.

Please note that both the web browser and the web server define a maximum length for an URL. Typically this is about 2048 Byte. If the required initialization data exceeds this limit the URL can also be invoked via HTTPS post.

#### Example:

Dictionary with the desired initial values:

```
{ "Form" : { "text1" : "Hallo", "text2": "World" } }
```

Base64 encoded dictionary:

```
eyAiRm9ybSIgOiB7ICJ0ZXh0MSIgOiAiSGFsbG8iLCAidGV4dDIiOiAiV29ybGQiIH0gfQ==
```

**Note:** If used within an URL parameters the "=" character (used for padding by the base64 encoding) needs to be URL encoded as %3d.

```
eyAiRm9ybSIgOiB7ICJ0ZXh0MSIgOiAiSGFsbG8iLCAidGV4dDIiOiAiV29ybGQiIH0gfQ%3d%3d
```

URL to invoke:

```
https://www.iyopro.com/iyopro/Form.aspx?wfid=11206&userdata=eyAiRm9ybSIgOiB7ICJ0ZXh0MSIgOiAiSGFsbG8iLCAidGV4dDIiOiAiV29ybGQiIH0gfQ%3d%3d
```

As result the start form will be invoked and the form field "text1" and "text2" will be set to the initial values "Hello" and "World".

### Start Workflow by Event

To start a workflow by event, ensure that the workflow has been [built](#) and it contains a Receive Message Start Event.

Follow the instructions given to [trigger a message programmatically](#)

## Start Workflow by Tag

To start a workflow by tag, ensure that the workflow has been [built](#) and it contains a unique tag. You use the same short URL as described in the "Start Workflow by Form" section. Instead of the wfid, use the tag as identifier:

```
https://www.iyopro.com/iyopro/Form.aspx?workflowtags=<name of the tag>
```

If you have several tags, you can also use more than one tag in the URL by separating the tags with commas:

```
https://www.iyopro.com/iyopro/Form.aspx?workflowtags=<name of the tag>,<name of the tag>
```

## Define waiting time

Some workflows are supposed to wait for a predetermined duration or until a point of time before taking an action. An example is the waiting time before sending a reminder for tasks. Intermediate timer events present such scenarios. They can be used as ordinary intermediate and as attached intermediate events.

Place an intermediate time event in the collaboration diagram to define at which position of the workflow a waiting time is required.

Navigate to the property set of the intermediate timer event and click on the "Edit" button of the timer property. The timer settings will be displayed now.

The screenshot shows a 'Properties' window for a 'Timer Intermediate' event. The 'Property set' is 'Modeling'. The 'General Properties' section includes: Element Type (Timer Intermediate), Id (1), Name (2 days), Note, and Attribute Extensions (with an 'Edit' button). The 'Visualization' section includes: Color (Automatic). The 'Intermediate Event Properties' section includes: Interrupting (checked). The 'Metadata Properties' section includes: Tags (with a menu icon). The 'Timer Event' section includes: Timer (with an 'Edit' button).

Insert an expression to determine the required waiting time and return the calculated date time value as "Result". The result must be a value in UTC (Universal Time Coordinated) format. Otherwise timezone related delays might appear while triggering timer events.

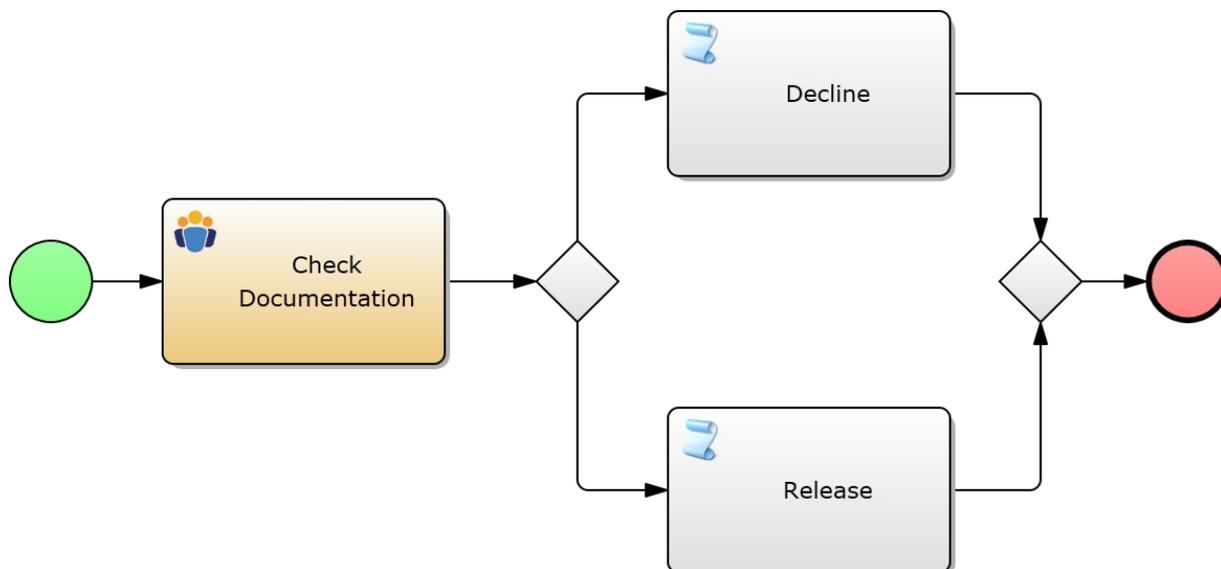


Click OK and the calculated time will be considered in the workflow.

## Sample Documentation Release Workflow

This section provides a step by step guide to create a sample documentation release workflow. This process is used to review and deploy a published documentation into the IYOPRO portal.

The sample process is described as follows:



## Usage

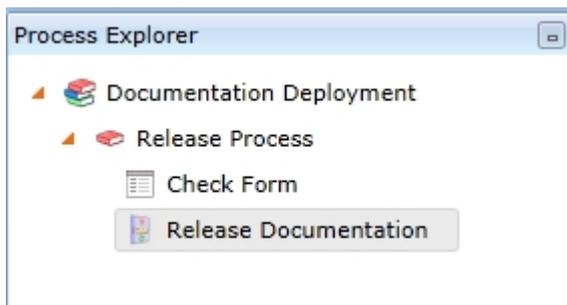
The author of a process documentation selects this workflow in the publishing dialog and completes the publishing task. IYOPRO will then start this release workflow.

The auditor receives a form where he can decline or release the documentation package. Depending on the decision either the decline or the release script task will be executed.

## Create a BPI Project

A workflow is always the result of a BPI Project. So the first step is to create a new BPI Project in the team of the process documentation. Detailed information how to create BPI projects can you find [here](#).

After you created the BPI-project add a form and a collaboration diagram to the project. The result should look like this:

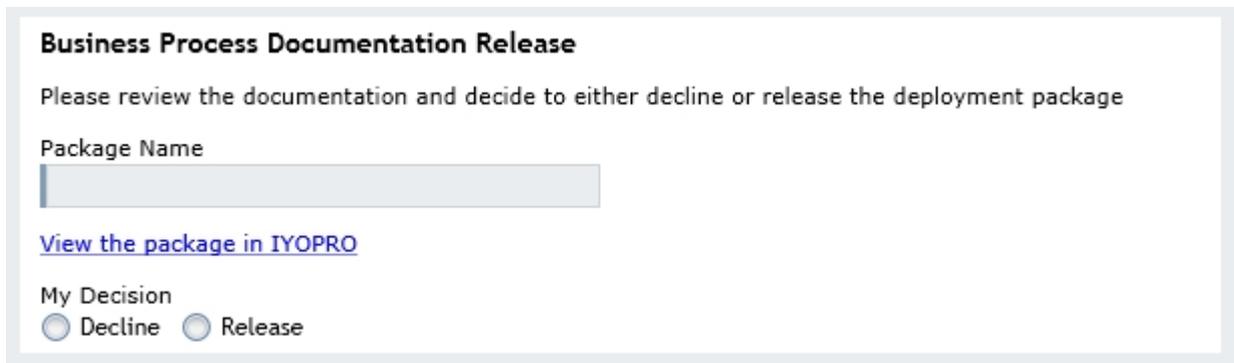


Now edit the [workflow properties](#) and ensure, that this workflow is a documentation release workflow.

In the next chapter we edit the [check form](#)

## Check Form

Open the Check Form and add the following elements to the form:

A screenshot of a form titled "Business Process Documentation Release". The form contains the following elements: a heading "Business Process Documentation Release", a paragraph "Please review the documentation and decide to either decline or release the deployment package", a text input field labeled "Package Name", a blue hyperlink "View the package in IYOPRO", and a section titled "My Decision" with two radio buttons labeled "Decline" and "Release".

Edit the properties of the form elements as follows:

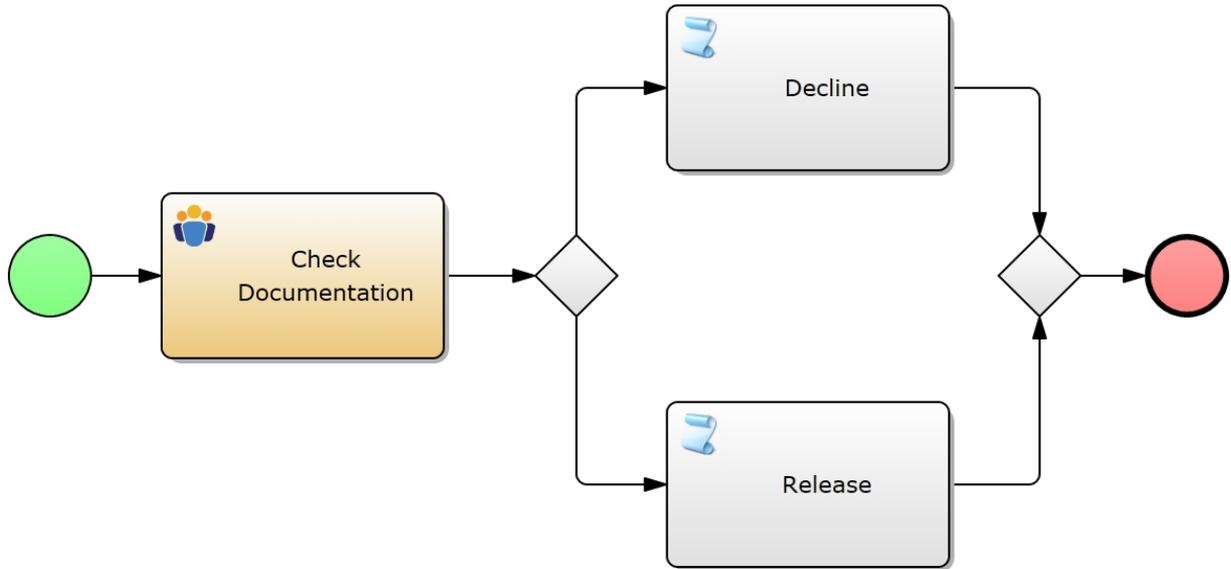
- Package Name TextBox  
Assign the VariableName "PackageName" to the input properties
- View the package in IYOPRO Hyperlink  
Assign the VariableName "PackageUrl" to the link properties
- My Decision RadioButton  
Assign the VariableName "Released" to the input properties

The variable names will be referred in the process diagram to respond to the users input.

In the next chapter we edit the [release documentation process](#)

## Release Documentation Process

Open the collaboration diagram and add the following shapes to the diagram:



In the next step it's important to adjust the properties of the shapes so that they work as expected.

In any order edit the properties of [Check Documentation Activity](#), [Decline Activity](#), [Release Activity](#), [Decision Path 1](#) and [Decision Path 2](#)

### Check Documentation Activity

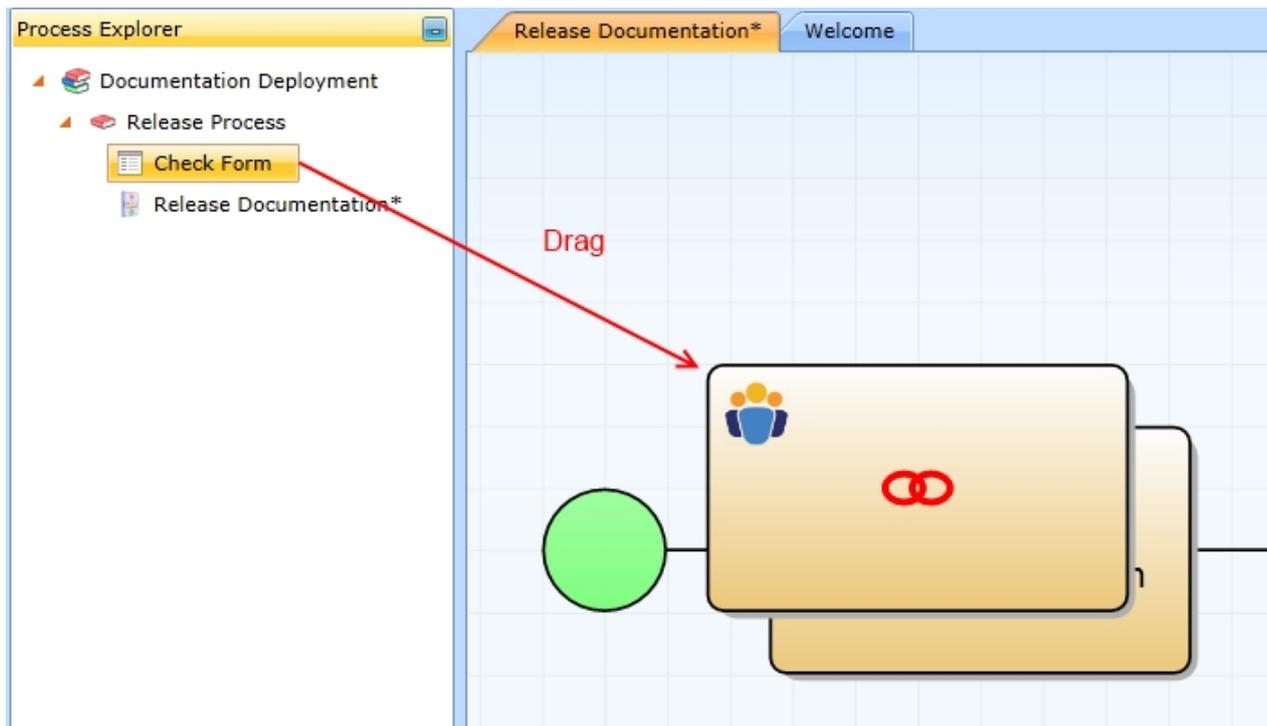
The check documentation activity is a user task which will request a decision from the auditor.



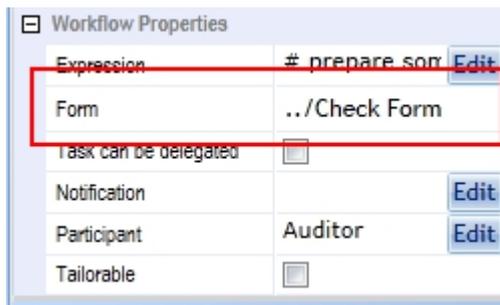
To complete this perform the following steps:

### Assign Form

To assign a form to the user task drag the form out of the process explorer over the user task in the collaboration diagram. Drop the form as soon as IYOPRO shows the marriage rings.



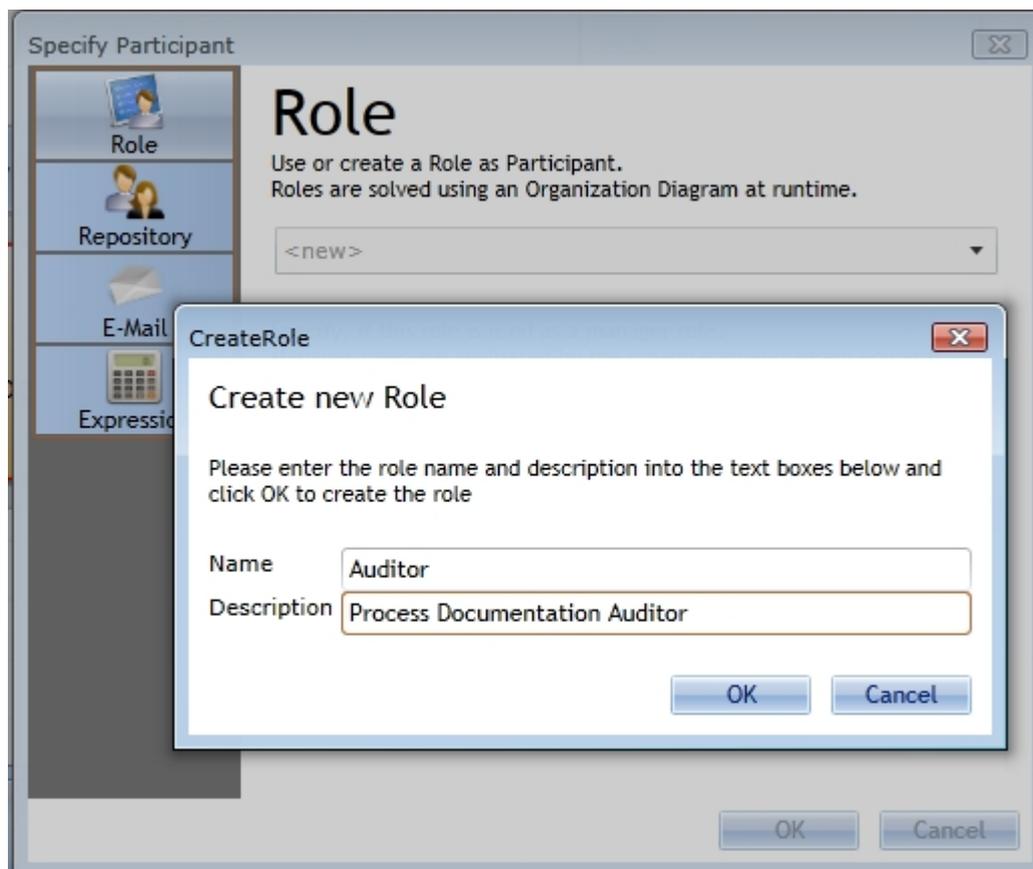
This will create a relation between the form and the task. The link is stored in the user task properties. You may also edit the link by your own if required.



## Assign Participant

There are different ways how to define a participant. In this example we will use a role and assign all members of the [package reviewer group](#) to that role.

To create a role open the Participant property. In the [participant editor](#) create a role named "Auditor" and assign it to the user task.

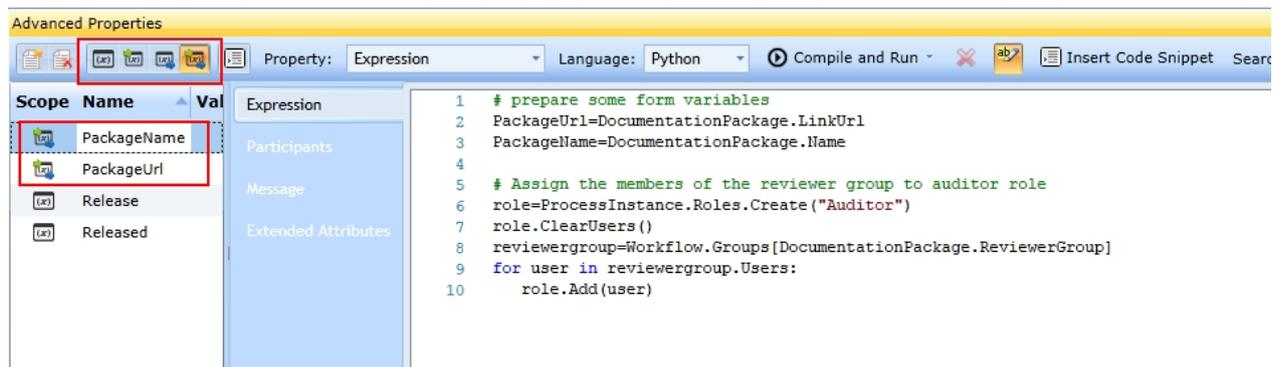


## Prepare user task

The last step is to prepare the user task. To do this open the user task [expression property](#). Switch the language to Python and enter the following python code:

```
1 # prepare some form variables
2 PackageUrl=DocumentationPackage.LinkUrl
3 PackageName=DocumentationPackage.Name
4
5 # Assign the members of the reviewer group to auditor role
6 role=ProcessInstance.Roles.Create("Auditor")
7 role.ClearUsers()
8 reviewergroup=Workflow.Groups[DocumentationPackage.ReviewerGroup]
9 for user in reviewergroup.Users:
10     role.Add(user)
```

In line 2+3 the script assigns the LinkUrl and Name attributes of the DocumentationPackage object to process instance attributes. This is required because we wish to provide these information in our form later. Ensure, that you at least Export the attributes out of your script into the [global scope](#). To do this select the attributes and choose Export.

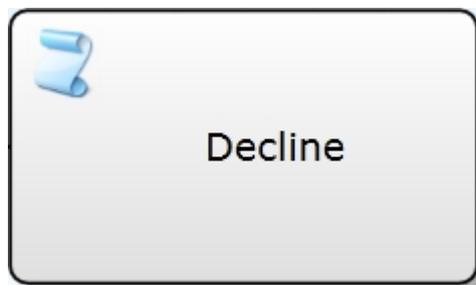


In line 6 we create a new role Auditor or receive the role if it already exists. In line 7 we remove possible members from the role.

Line 8 gets an instance of the reviewer group. Line 9+10 adds all members of the reviewer group to the role.

### Decline Activity

The Decline Activity is used to decline the documentation package.



### Edit script task

The only step is to edit the script task. To do this open the script task [expression property](#). Switch the language to Python and enter the following python code:

```

1 # Delete documentation package
2 DocumentationPackage.Delete ()

```

### Release Activity

The Release Activity is used to release the documentation package.



### Edit script task

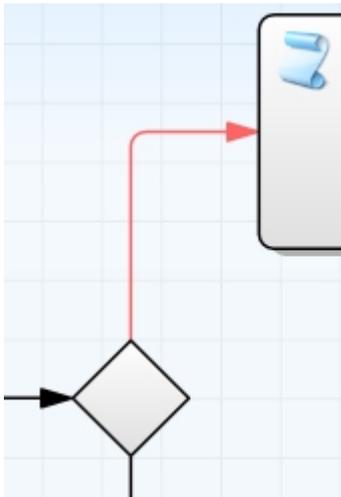
The only step is to edit the script task. To do this open the script task [expression property](#). Switch the language to Python and enter the following python code:

- 1 # Release documentation package
- 2 DocumentationPackage.Release()

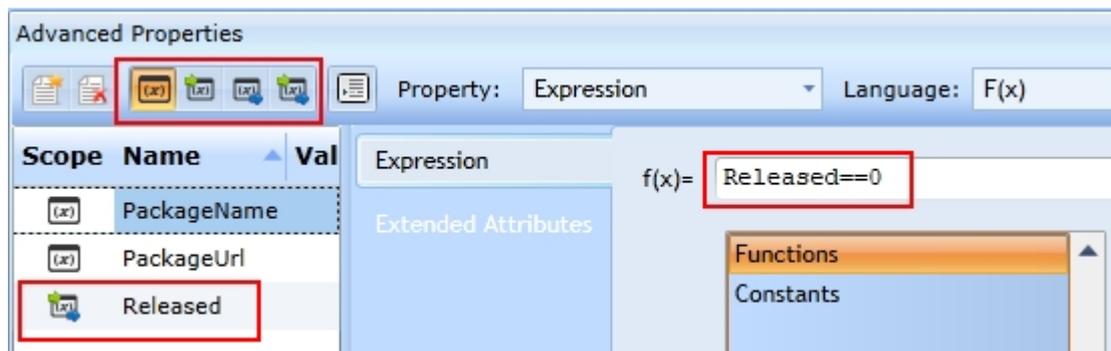
### Decision Path 1

After the Auditor has completed the form, the process instance attribute "Released" contains either "0" for Decline or "1" for release.

This must be mapped to the outgoing sequence flows of our decision gateway. Select the sequence flow which directs to the Decline Activity.



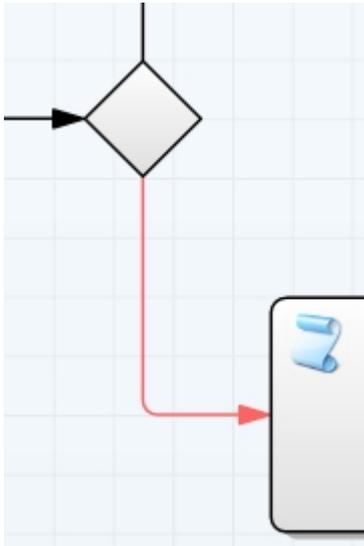
Enter "Released==0" in the F(x) input field. Also indicate, that the attribute "Released" has to be imported from the global scope into our formula.



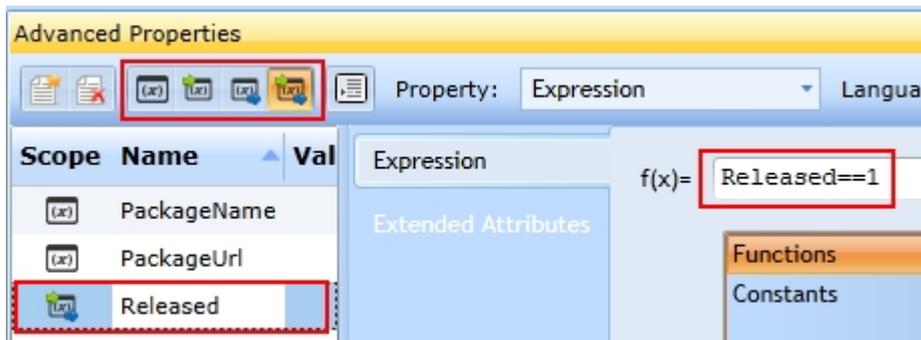
### Decision Path 2

After the Auditor has completed the form, the process instance attribute "Released" contains either "0" for Decline or "1" for release.

This must be mapped to the outgoing sequence flows of our decision gateway. Select the sequence flow which directs to the Release Activity.



Enter "Released==1" in the F(x) input field. Also indicate, that the attribute "Released" has to be imported from the global scope into our formula.



## Build Workflow

If you completed all steps you can now build the workflow. Information on how to build a workflow can be found [here](#).

If you now [publish](#) a business process documentation the workflow is available and can be chosen to control the deployment process.

## Monitor Workflow Queues

You can monitor two types of workflow queues:

- workflow tasks
- scheduled workflows

The scheduled workflow queue contains workflows that have a timer start event, which starts the workflow on a specific date. This queue is the custom queue. The workflow tasks queue, on the other hand, is the default queue and contains all other workflows.

In the properties of the [Workflow Build Options](#) of a workflow, you may change the queue from default to custom and vice versa.

The table provides you with information about the workflow / process instance for which there is a list entry and about the user who caused this list entry. Furthermore, you receive information about the time it will be triggered as well as the number of repeats, whether it is to be repeated, the state and whether it was aborted.

Depending on your permissions to execute system management functions, you can disable or enable the processing of jobs using the two buttons above the table.

In addition, you can see whether IYOPRO is in normal service operation or in recovery mode. The possible

statuses are:

- Normal service operation
- Service recovery mode active
- Recovery mode stop in progress

Id	Name	User	Scheduled	Repeat	RepeatUnit	State	Abort
24042	Process Instance #1454	Batch	Sunday, July 31, 2022 10:00:00 PM	0	Off	Hold	
25284	Process Instance #1458	Batch	Sunday, May 15, 2022 11:58:39 AM	0	Off	Hold	
25285	Process Instance #1458	Batch	Sunday, May 15, 2022 11:58:39 AM	0	Off	Hold	
25286	Process Instance #1458	Batch	Sunday, May 15, 2022 11:58:39 AM	0	Off	Hold	
25422	Process Instance #1510	Batch	Friday, May 20, 2022 8:58:48 AM	0	Off	Hold	

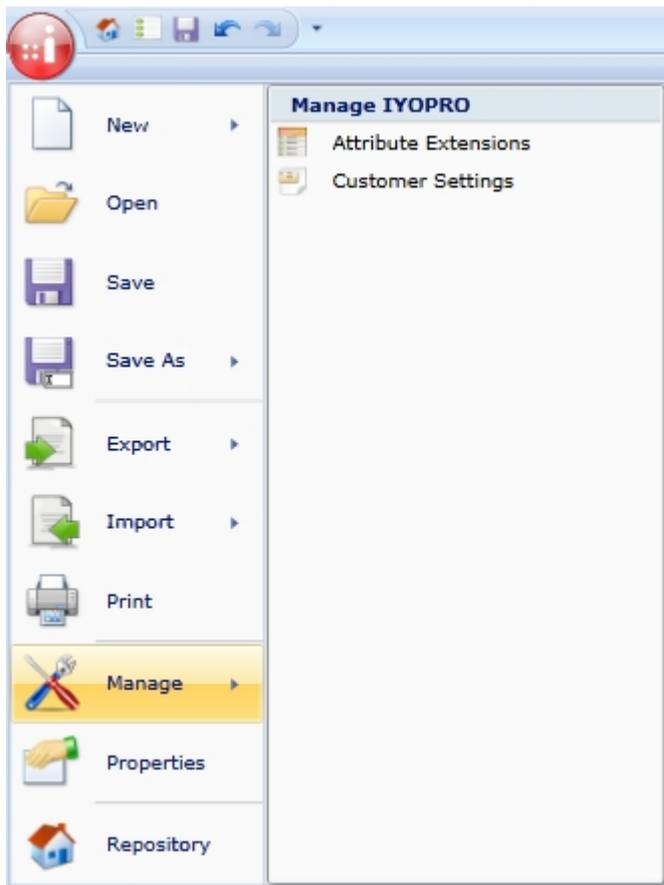
## Customization

---

IYOPRO provides the following customization options for business process modelling

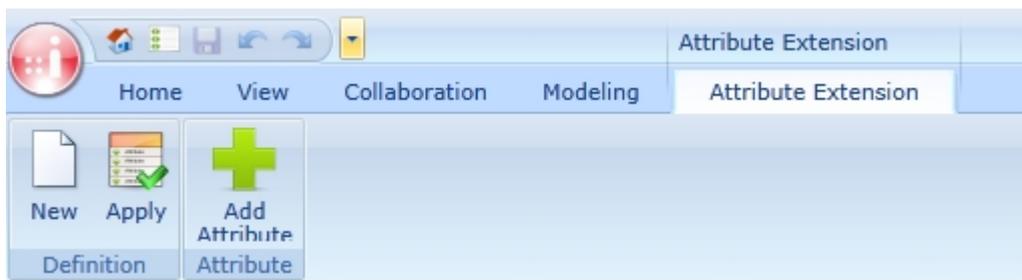
- Attribute Extensions  
Attribute Extensions provide a way to extend all available shapes with individual customer attributes.
- Customer Settings  
Use customer settings to adjust shape preferences like color, description template and more

Both features can be found in the Application Menu



### Attribute Extensions

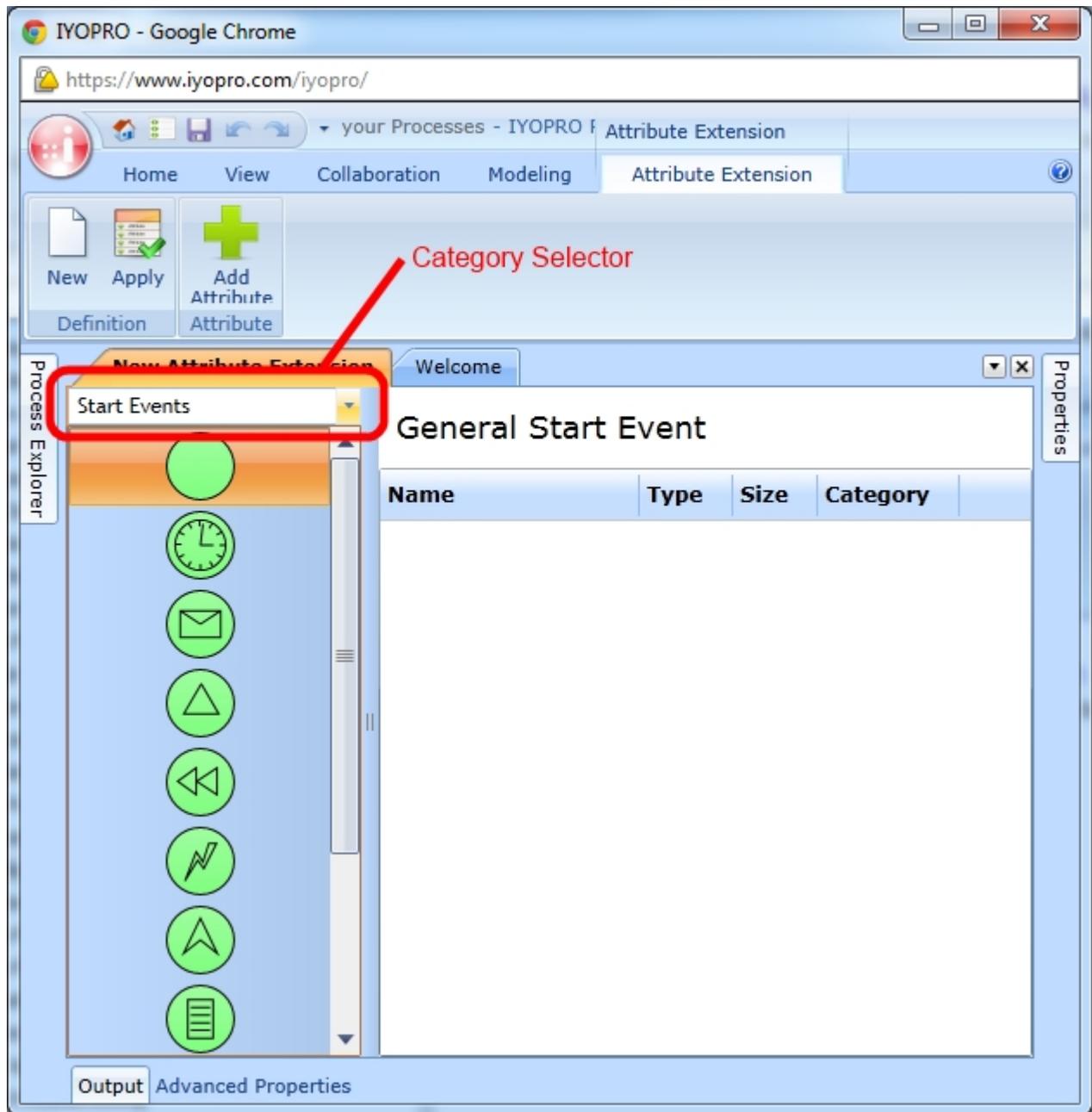
To activate the Attribute Extension Editor click on Manage->Attribute Extensions from the Application Menu



### Create new Attribute Extension definition

To create a new Attribute Extension definition click on the New button of the Definition ribbon group. A

new empty Attribute Extension definition will appear.  
 The associated Editor is splitted into two Areas. The left pane shows all available shapes, grouped by categories which may be chosen by the category selector.  
 The right pane shows the extended attributes for the selected shape.



To add a new Attribute click on the Add Attribute Button in the Ribbon menu. A new Row will appear.

### General Start Event

Name	Type	Size	Category
<new Attribute>	String	0	

Type in the Name of the Attribute and select a type. The following types are available:

- String

Used to enter any type of text

- Integer  
Used to enter only integer values
- Float  
Used to enter floating point values
- Boolean  
used to mark the Value with a Check Box
- Date  
Used to enter the value with a date picker
- Option  
Used to choose a value from a list

The String-Type allows to specify a size. The entered attribute value may not exceed this size. 0 means that the size is unlimited.

If you have chosen the Option-type you need to specify the options. To specify the options click on the properties-symbol of the desired line. In the appearing dialog enter one option per row.

If You are finished save the Attribute Extension with the Save Button

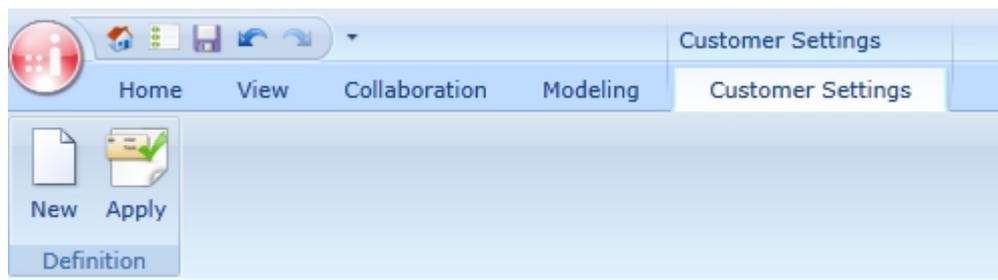
## Apply Attribute Extension

To apply an attribute extension click on the Apply Button of the Ribbon. In the upcoming dialog select on or more attribute extensions and hit OK. If you wish to deactivate attribute extensions you may uncheck the appropriate line.

Deactivating attribute extensions will not remove assigned attributes from existing shapes. They will only not be shows on new shapes.

## Customer Settings

To activate the Customer Settings Editor click on Manage->Customer Settings from the Application Menu



### Create new Customer Settings definition

To create a new Customer Settings definition click on the New button of the Definition ribbon group. A new empty Customer Settings definition will appear.

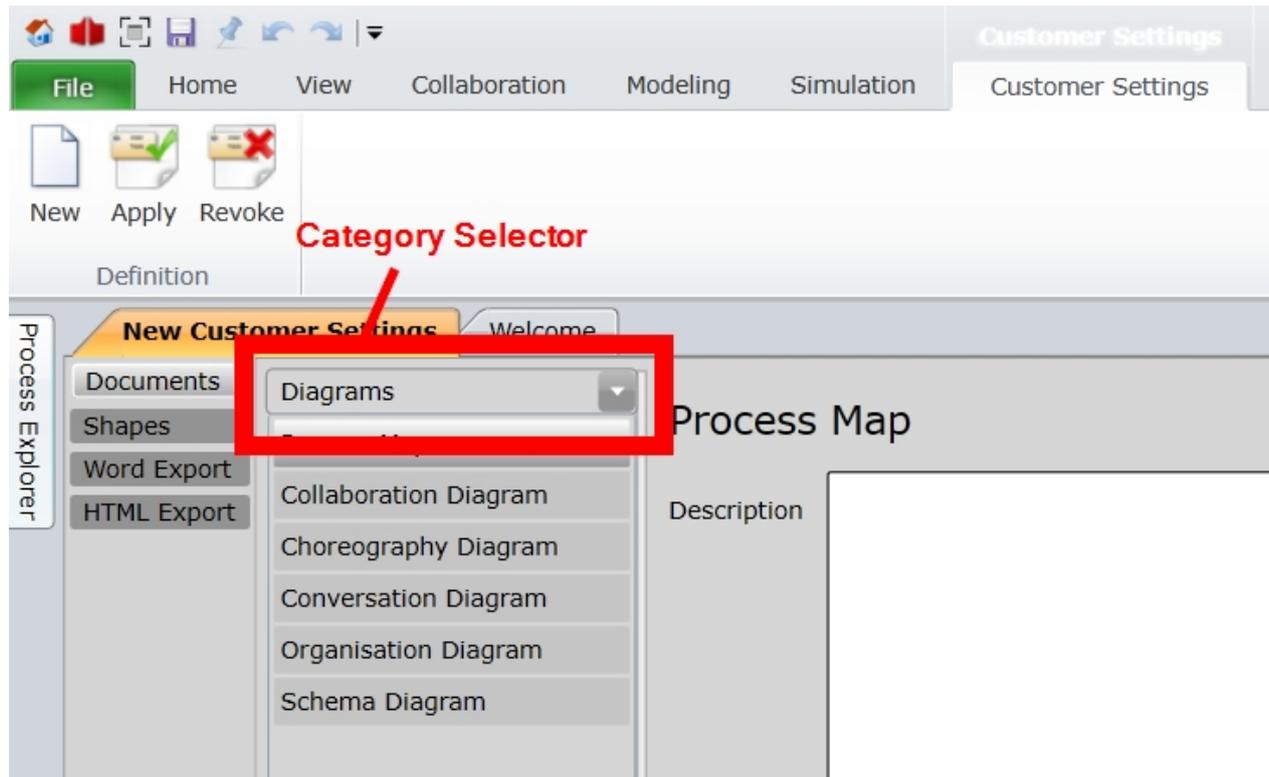
Customer settings are divided into the following sections:

- Documents  
Used to edit document specific settings
- Shapes  
Used to edit shape specific settings
- Word Export  
Used to change settings related to the Word export

- HTML Export  
Used to change settings related to the HTML export

## Document Settings

The Document Settings editor is splitted into two Areas. The left pane shows all available document types, grouped by categories which may be chosen by the category selector. The right pane shows the available settings for the documents.

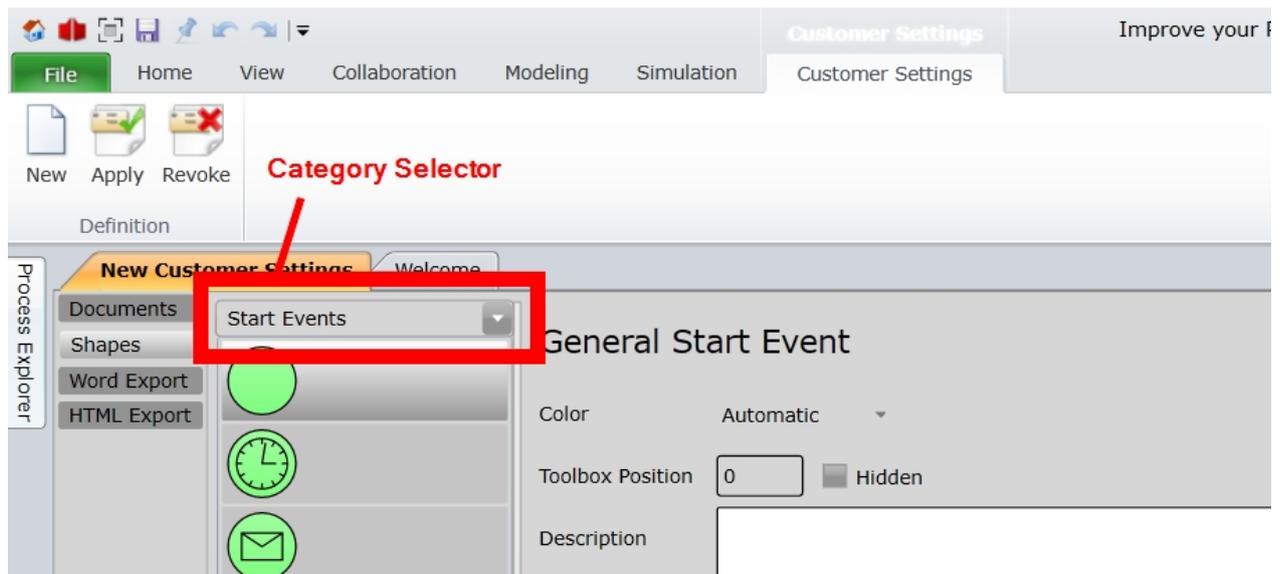


The following settings may be adjusted:

Description    Enter the description template which should appear if the diagram properties are shown in the modeler

## Shape Settings

The Shape Settings editor is splitted into two Areas. The left pane shows all available shapes, grouped by categories which may be chosen by the category selector. The right pane shows the available settings for the shapes

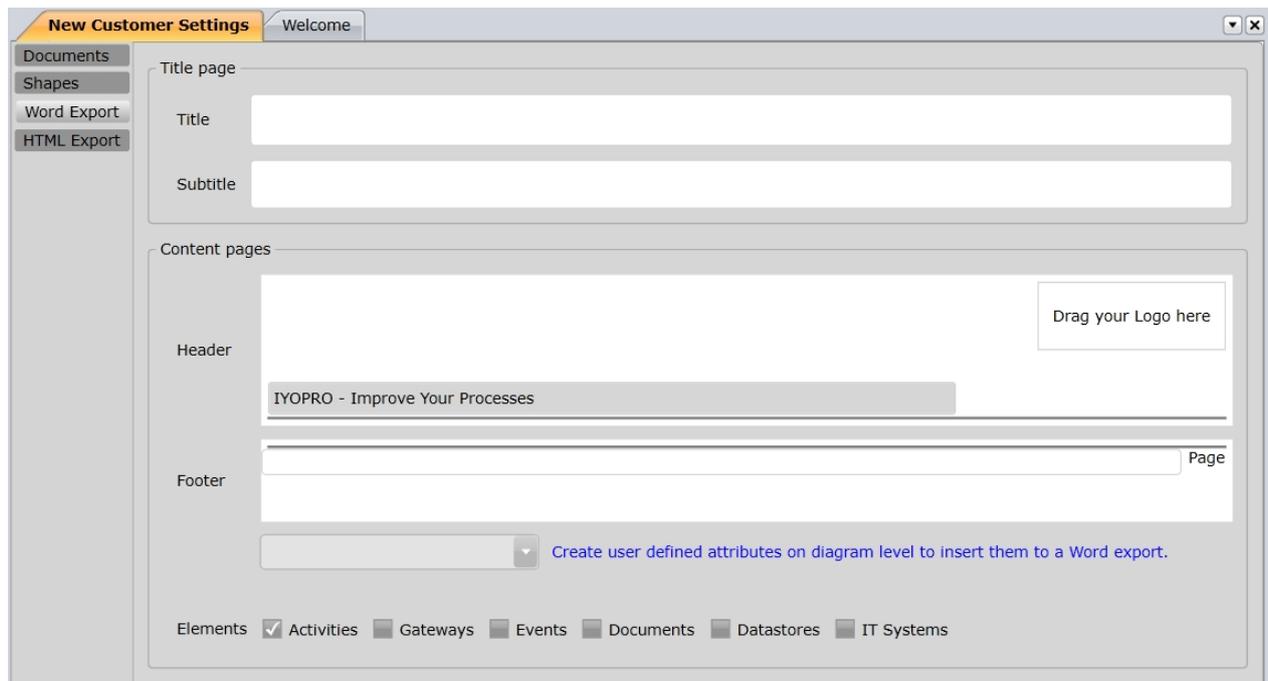


The following settings may be adjusted:

Color	Choose a base color for the shape. This color will be passed to all diagram shapes, which use have their local shape color set to automatic.
Toolbox Position	Specify the position, where the shape will be positioned in the <a href="#">modelers toolbox</a> . A value of 0 specifies the default position. 1 specifies the first position, 2 the second and so forth.
Hidden	Toggle this to hide the shape in the appropriate toolbox. Use this to simplify the toolboxes for restricted users.
Description	Enter the description template which should appear if the shape properties are shown in the modeler
Documentation	Define if element descriptions are shown in the portal documentation.
Library	Define if the element can be used as library element.
Custom Shapes	Manage individual appearances of the element.

## Word Export Settings

The Word Export Settings are used to configure the output of the word export to follow your corporate identity.

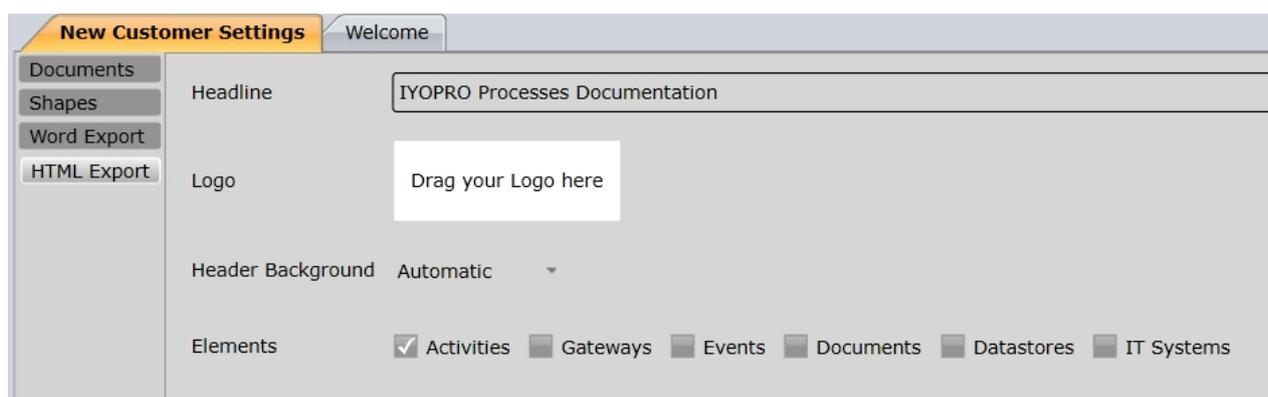


The following settings may be adjusted:

Title	Specify the title on the title page.
Subtitle	Specify the subtitle on the title page.
Header	Drag an Image of your company logo from your file explorer over the desired logo position. Enter the Header text along your needs.
Footer	Define which advanced attributes of the diagram are shown in the footer.
Elements	Specify, which shape types will be exported in the word export. Per default only activities will be documented.

## HTML Export Settings

The HTML Export Settings are used to configure the output of the HTML export to follow your corporate identity.



The following settings may be adjusted:

Headline	Enter the header text along your needs
Logo	Drag an Image of your company logo from your file explorer over the desired logo position
Header Background	specify the color of the colored HTML areas
Elements	Specify, which shape types will be exported in the HTML export. Per default only activities will be documented.

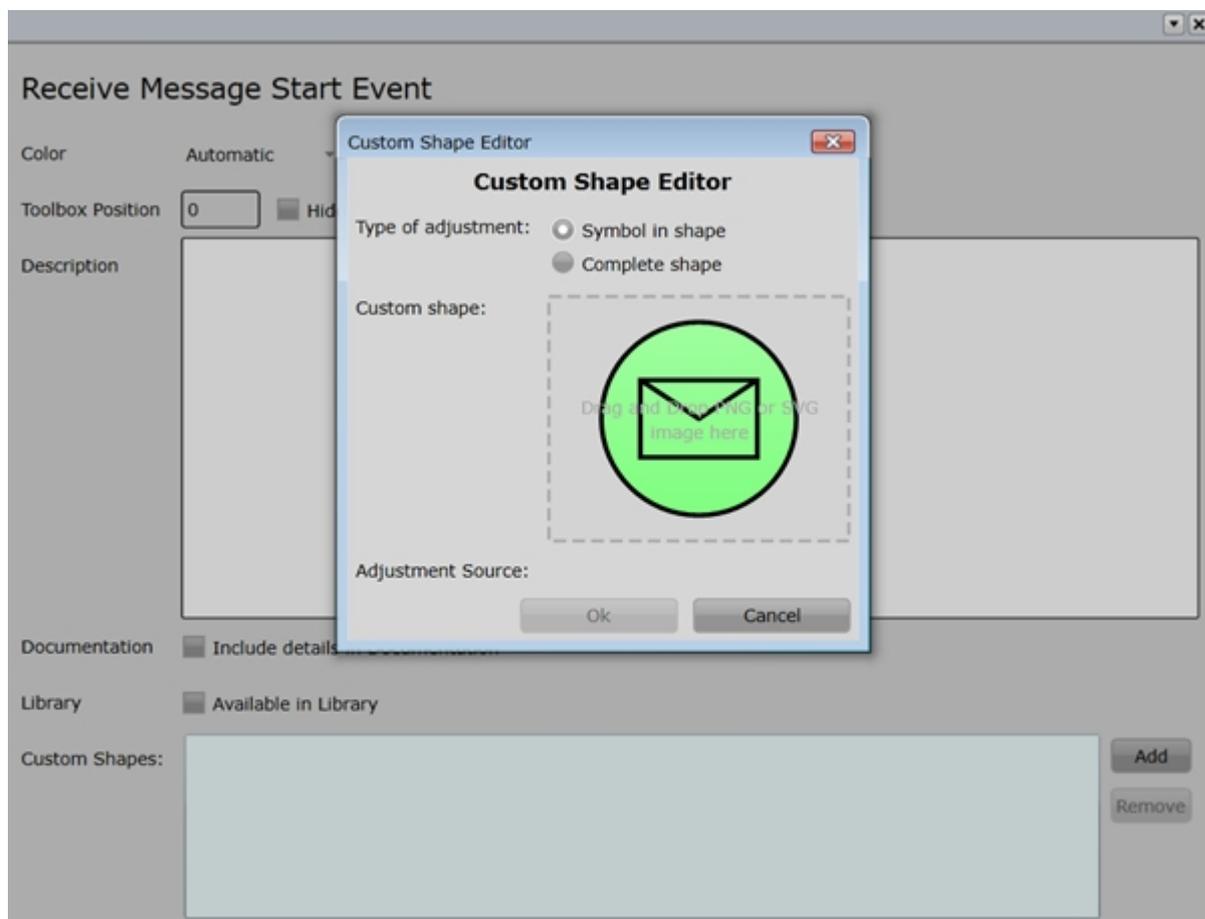
## Custom Shapes

Custom shapes allow an individual presentation of elements which are provided by BPMN 2.0. This feature only applies to the appearance of shapes and will not affect the elements behavior in [process automations](#) and [process simulations](#).

### Manage a new custom shape

#### Create a custom shape

A custom shape is created in the shape settings of the [customer settings](#). Use the category selector to open the settings of the desired shape. Press the "Add" button afterwards to open the custom shape editor.



The following settings may be adjusted:

Type of adjustment	Decide whether the whole shape will be replaced by the new imported image or only the symbol in the shape.
Custom shape	Drag and Drop the image of the custom shape into the preview dialog. Only SVG and PNG can be used. A preview of the custom shape is shown.
Adjustment source	The name of the imported file is displayed here.

The formats SVG and PNG can be used to import images as custom shapes. SVG's are highly recommended since this data format is scalable and won't cause any quality losses when resizing the element later on.

### Edit an existing custom shape

Custom shapes are edited in the customer settings. Use the category selector to open the settings of the desired shape. Double click the icon of the custom shape to open the custom shape editor. The custom shape can now be adjusted as it is described in the paragraph above.

### Remove a custom shape

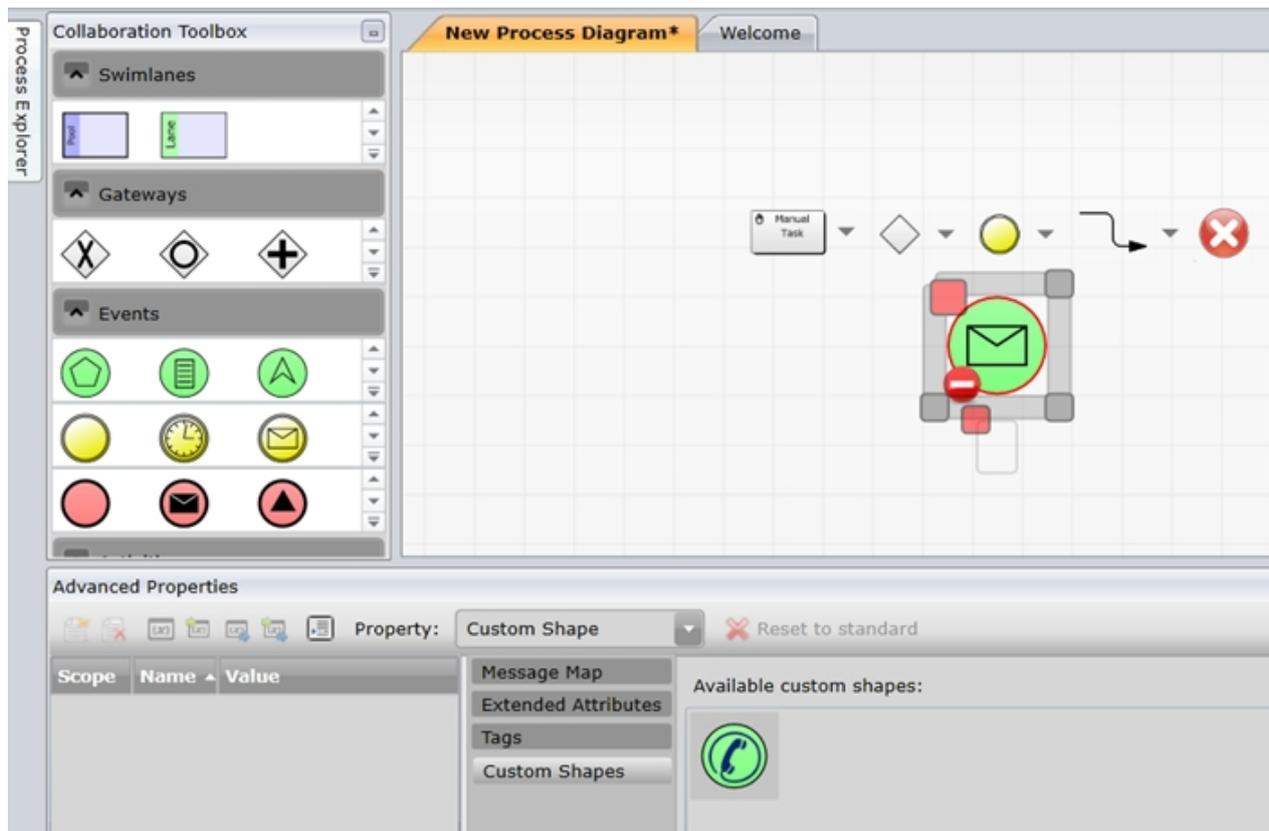
Custom shapes are removed in the customer settings. Use the category selector to open the settings of the desired shape. Selected the custom shape and press the "Remove" button afterwards to open the custom shape editor. When a custom shape is removed every applied element in diagrams will be set back to standard.

### Use a custom shape

During modeling only those custom shapes can be used which are created for the currently activated customer settings. Nevertheless users are able to see any custom shape that are already used in a diagram. They just won't be able to create a new element with such custom appearance.

### Apply a custom shape

To use a custom shape the corresponding BPMN 2.0 element must be dragged into the diagram first. Afterwards its appearance can be customized in the advanced properties. Open the tab "Custom Shapes" and select the desired custom shape for the modeled element. The appearance is changed immediately.



### Reset a custom shape

The appearance of a customized element can be set back to standard in the advanced properties. Make sure the element is selected. Press the button "Reset to standard" in the "Custom Shapes" tab of the advanced properties. The appearance is changed immediately.

### Integration of the web portal into applications

The YOPRO web portal can not only be opened as any browser page, but can also be integrated into applications. You can find examples below:

#### The whole web portal

The easiest way to integrate the web portal is to place a link on your intranet page, for example. You could then just have it open as a new tab.

The URL of the web portal is:

<https://www.iyopro.com/iyopro.portal/>

(Of course, if you are using YOPRO as an on-premises installation, the URL will differ from this one.)

Depending on your intranet site, you can simply use html as shown below:

```
<a href="https://www.iyopro.com/iyopro.portal/" target="_blank">YOPRO Web</a>
```

#### SVG export of a diagram

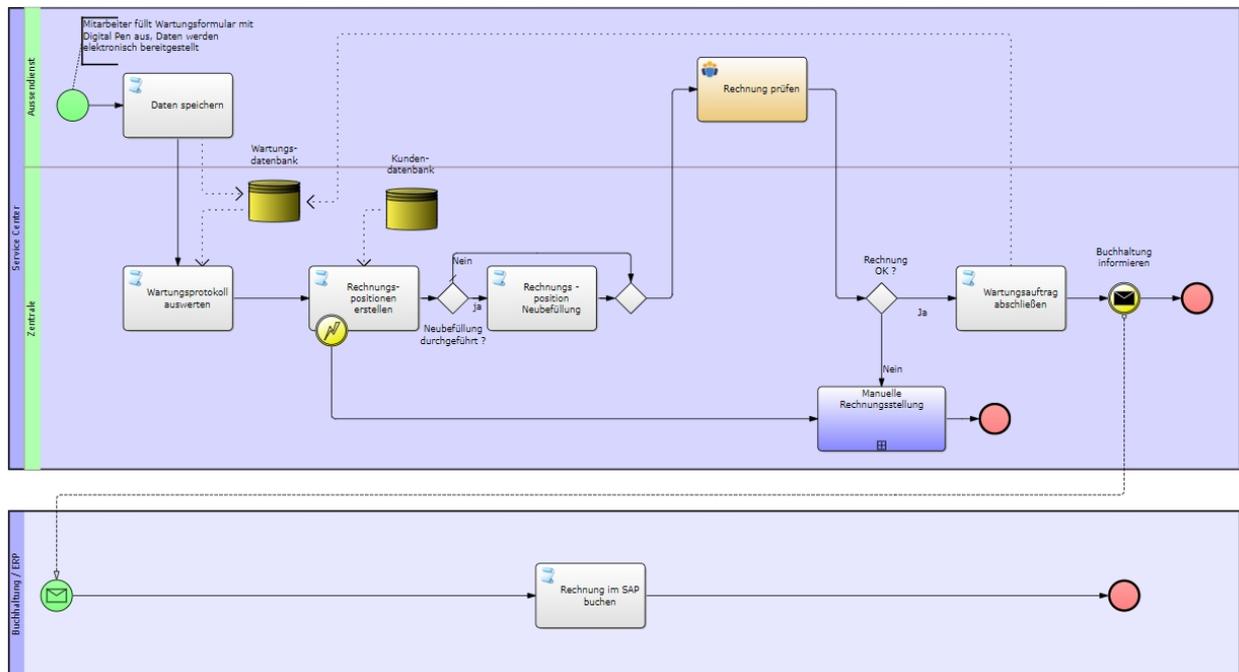
In YOPRO App you have the option of exporting a diagram as SVG. If you also want to provide this outside of the app, you can use the URL as shown below and simply modify it so that it opens the diagram you want.

<https://www.iyopro.com/iyopro.portal/svg?diagramid=37637>

The following image shows what the page looks like when you click on the link.

## Wartungseinsatz

[Download as SVG Image](#) [Open PNG Image](#)



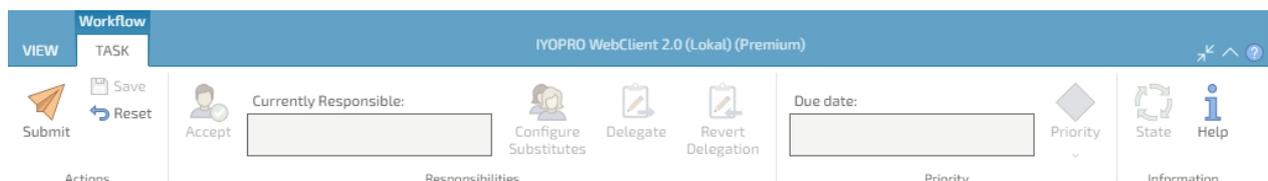
## Workflow form / task

Integrate start forms or existing tasks into other applications. The URLs needed for this look like this:

Start form: <https://www.iyopro.com/iyopro/Form.aspx?wfid=173065>

Analogous for tasks: [https://www.iyopro.com/iyopro/Form.aspx?taskid=<task\\_guid>](https://www.iyopro.com/iyopro/Form.aspx?taskid=<task_guid>)

As you can see in the image below, you will then only get the menu relevant to the task.



## Engineering Change Request (ECR)

ECR Number	Title *	Status
<input type="text"/>	<input type="text"/>	In Decision <input type="button" value="v"/>
Requestor	Request Date *	Decision By
Kristina	17. 1. 2023 <input type="button" value="calendar"/>	<input type="text"/>
	DecisionDate	
	DD. MM. YYYY <input type="button" value="calendar"/>	

For further information regarding the start of workflows, see the topic: [Start Workflow outside IYOPRO](#)  
 You can also find more information about embedding forms and the associated setting options under this topic: [Embedding Forms in another application](#)

## Process instance details of a workflow

The process instance details are also available via an URL. The URL looks like this:

<https://www.iyopro.com/iyopro/Form.aspx?instanceid=1877>

Process Instance   Tasks   Roles   Diagram   Log   Attributes   



### ECR

Launched by Demo User at 17.11.2015, 21:57:41

[Resume](#)

## Engineering Change Request

Description not available.

[Technical information](#)



Note:

Links that refer to "https://www.iyopro.com/iyopro/Form.aspx" automatically redirect to the current portal (analogous to opening a task in IYOPRO App). We recommend using this version, where available, to avoid having to adjust the URLs in the future.

If you are using IYOPRO as an on-premises installation, the beginning of your URLs will differ from the displayed ones.

## Embedding Forms in another application

### Embedding Forms in another application

IYOPRO forms can be displayed embedded in other applications. When using this mode additional functions are available to control the processing of the form by the hosting application.

The functions will only be invoked, if the form is displayed inside another application using the Internet Explorer control.

### Controlling form submission

Prior to the submission of the form a "pre-submit handler" can be invoked in the hosting application. This handler can check, amend or modify the form data. If an exception occurs during execution of the handler the form submission will be canceled.

As the name indicates the handler is invoked prior to any potentially defined form end action.

The pre-submit handler can be specified in the workflow properties of the form. It needs to be enabled and allows to specify a text string which is transferred to the hosting application. Which information is passed to the hosting application depends on the implementation. Please note, that the information is always transferred as text. Usually the data defined as "Host application parameter" will be a string representation of a data structure the application understands. But simple text commands like e.g. "CHECKDATA" can be used as well.

**Enable pre-submit handler for application embedded forms**

Host application parameter:

When the user clicks "submit" on the form, it is checked, if the form is  
a) embedded in another application  
b) the hosting application provided an accessible function "OnFormPreSubmit"  
If both conditions are met, the function is invoked in the hosting application and a representation of the form is passed.

## Exchange information between the form and the hosting application

Information about specific form properties can be obtained by the hosting application by calling a function via the InvokeScript method of the browser control. IYOPRO provides a set of functions allowing to obtain information about the pre-submit handler, the form variable values and to set the form variable values.

 string WPFGetFormDataProperty(string name)	Get the value form property <name>.
--	-------------------------------------

Within the hosting application the pre-submit handler property values can be read by querying the properties PreSubmitExpression.IsEnabled and PreSubmitExpression.Info. The first returns the value of the above checkbox, while the second returns the string provided as pre-submit handler parameter.

The values of individual form variable values can be read in a similar way, using the function:

 string WPFGetObjectPropertyAsJson (string name)	Get the value form form variable <name> as JSON.
---	--

Please note: that all form variable names provided need to be prefixed with "FormValues."

To return data to IYOPRO another function is available.

 void WPFSetFormDataProperty(string name, string value)	Set form variable <name> to the value <value>. Value has to be a JSON representation of the desired value.
--	--

## Controlling action button processing

A form with a button of the type "action" has an identical feature as the pre-submit handler. The button pre-click handler allows to specify an information which is transferred to the hosting application prior to executing the action script of the button.

Enable pre-click handler for application embedded forms

Host application parameter:

Similar to the processing for the pre-submit handler, if the action button is clicked, IYOPRO checks that the form is:

- a) embedded in another application
- b) the hosting application provided an accessible function "OnActionButtonClicked"

If both conditions are met, the function is invoked in the hosting application and the host application parameter is provided to it.

With the available functions described above form variable values can be read and also changed.

Please refer to the [pre-submit](#) sample for an example implementation.

### Pre Submit Example

#### Pre-Submit Handler Example

If the form is embedded in IYOPRO.PLM.Components as a workflow start form, a function in IYOPRO.PLM.Components can be invoked to process the form data prior to the form submission.

The protocol used with IYOPRO.PLM.Components expects a string that contains a python dictionary as data.

This dictionary contains a key to define the function to executed and a parameter to control which form variable should be provided to function.

#### Pre-Submit Example Data

```
{ "TYPE" : "FB", "NAME" : "IYOPRO.PLM.TEST.PRESUBMIT", "FBPARAMETER" : "Text" }
```

This would instruct IYOPRO to call the hosting IYOPRO.PLM.Components application when the submit

function of the form is invoked.

As part of the of this call the above text is transferred.

IYOPRO.PLM.Components decodes the information and obtains the value of the form variable specified by "FBPARAMETER".

This value is passed to the function module (TYPE="FB") with the name "IYOPRO.PLM.TEST.PRESUBMIT".

The function module itself processes the data and optionally returns information to the form.

For example the value of the text variable can be changed and an information, indicating the the form data was already processed, is set.

## IYOPRO.PLM.Components Example function module

```
! IYOPRO.PLM.TEST.PRESUBMIT
!
-----
! Example of a synchronous call form the IYOPRO web client via the pre-submit
handler
! Parameters:
! P1 = (IN) Dictionary of the form data
!           Not all expected key might be present, depending on the form
values
! P2 = (IN) Variable name
!           Set this variable with a string representing a Python dictionary.
!           Keys are the form variable names and their respective values (as
JSON) to be modified.

LOCAL FORMDATA[]
CREATE ARRAY FORMDATA FROM P1 OPT 'PYTHON'
SET ARR FORMDATA ROW -1 COL ACOLUMNS(FORMDATA)

LOCAL TXT
TXT = FORMDATA[0='Text'][1]
TXT = 'Pre-Submit-Handler(' + TXT + ')'

LOCAL RET
RET = '{ "Text" : "\"'+TXT+'\""}, "PROCESSED" : "\"Yes\""}'
EXEC P2+=RET'
```

Another typical usage is that within the form there is a checkbox, allowing the user to specify if he wants to have the form

data processed directly in the application embedding the form or by the workflow.

Depending on the value of the checkbox the function triggered by the pre-submit handler would either perform the processing or do nothing.

Likewise the workflow either perform the processing or skip it.

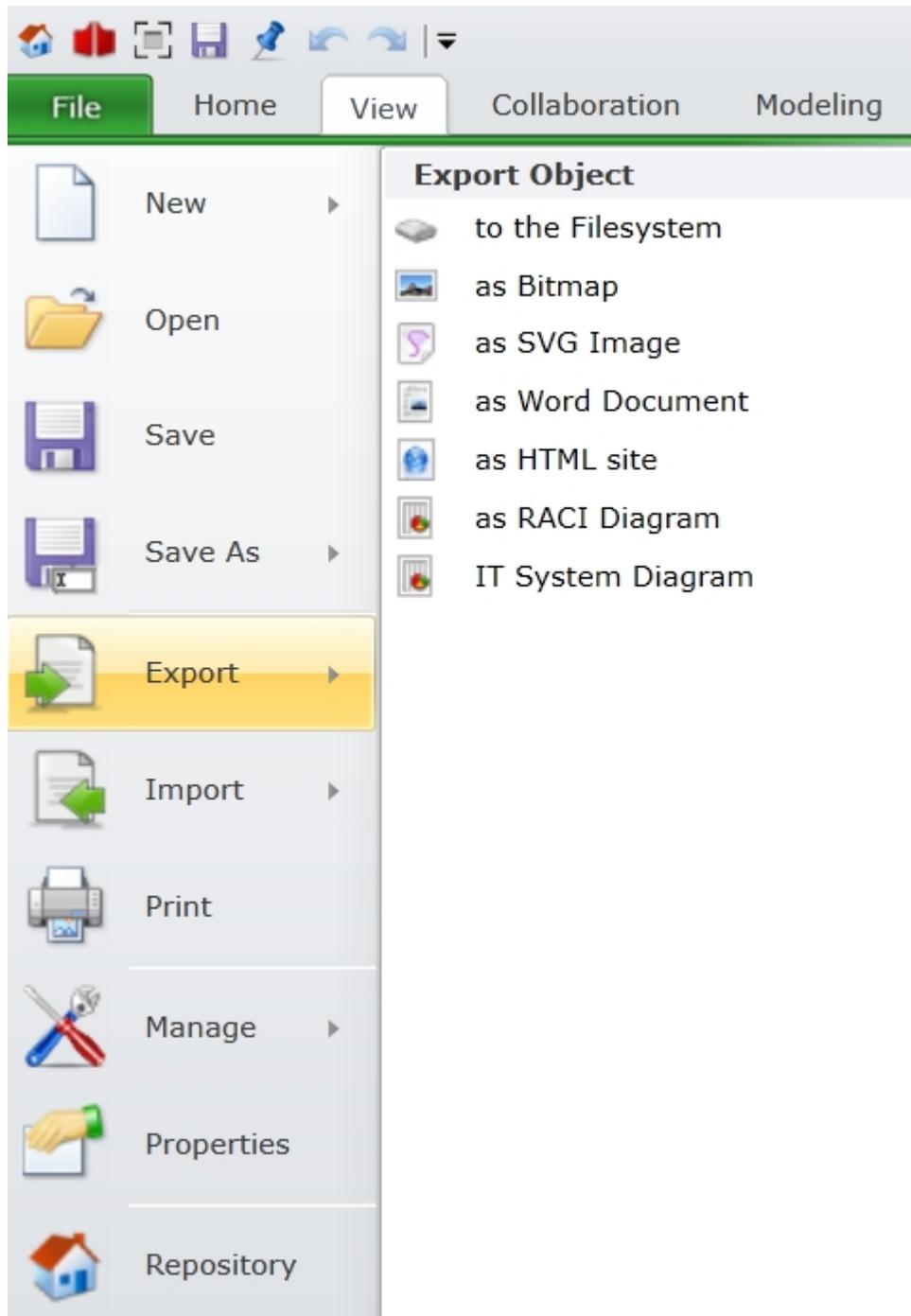
Thus the user can decide if a time consuming operation is to be performed while he waits (within the pre-submit handler of the

hosting application) or asynchronous by the workflow.

## Export and Import

---

Export and Import Operations can be found within the Application menu



### Export

#### Export to the Filesystem

To export IYOPRO process models to the file system use

Application Menu -> Export -> to the file system.

You may choose between the following file types:

1. IYOPRO File  
This export format uses an XML structure which allows to export all IYOPRO data types.
2. BPMN 2.0 File  
This export format is used to exchange data between different modeling systems. Use this if you wish to transfer the process model to another BPM-System.  
If you wish to export to the BPMN 2.0 format your data must be organized within a BPI project.

To import the data back to IYOPRO use [Import from file system](#)

## Export as Bitmap

To export a Bitmap from the active document use

Application Menu -> Export -> as Bitmap. This will create a high resolution bitmap of the document content.

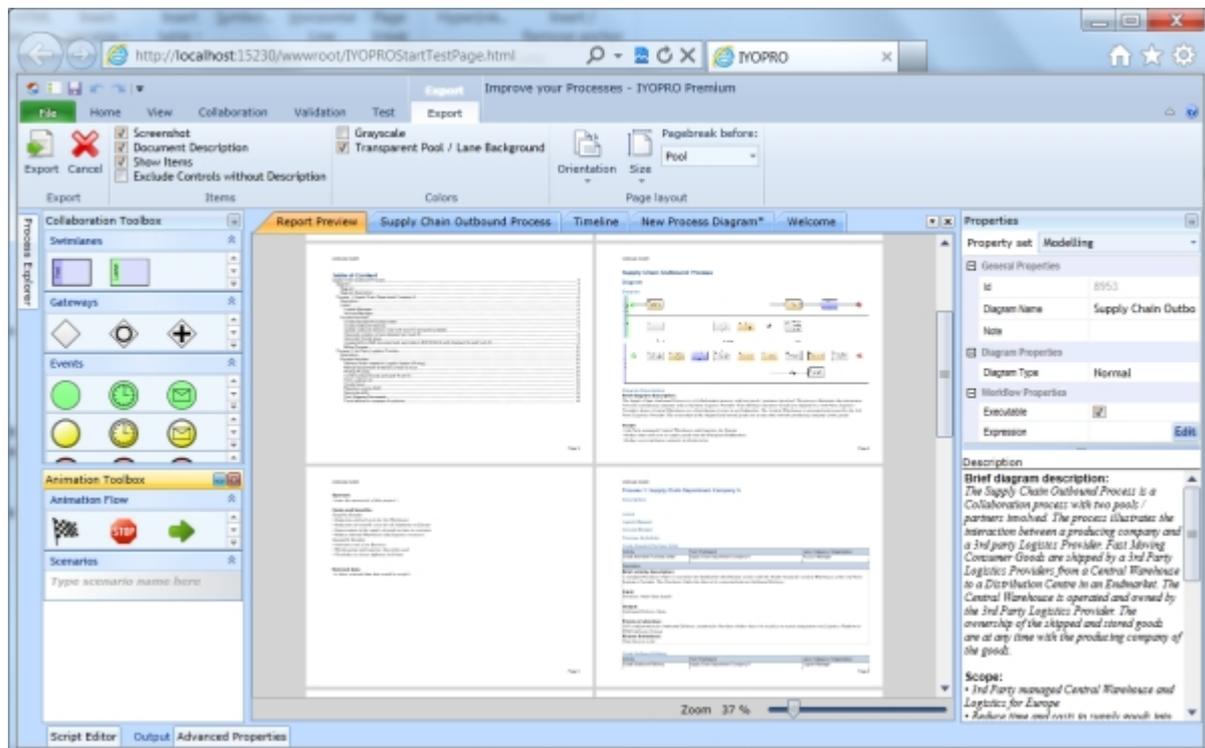
You may store it as png or jpg, depending on the extension you specify.  
If no extension is specified the default png is used.

Note: IYOPRO Basic users will see a watermark that this diagram is made by IYOPRO. If you wish to get rid of the watermark please upgrade to at least [IYOPRO Professional](#).

## Export as Word Document

To generate a process documentation in Word Format use

Application Menu -> Export -> as Word Document. This will create complete process documentation from the current Collaboration diagram.



A preview of the process documentation is shown.

The associated Ribbon tab provides the following options:

Export	Export the documentation into a Word document
Cancel	Cancel export and close preview
Order	Order the elements
Screenshot	Toggle to show / hide a screenshot of the diagram
Document Description	Toggle to show / hide the description provided on diagram level
Show Items	Toggle to show / hide detailed description of the diagram elements
Exclude Controls without Description	Toggle to show / hide elements where no description has been entered
Grayscale	Toggle to enable / disable colors
Transparent Pool / Lane Background	Create a printer friendly representation of Pools and lanes
Orientation	Toggle between Portrait and Landscape
Size	Choose a paper format
Pagebreak before:	Choose between Diagram - Each Diagram paragraph will start on a new page Pool - Each pool paragraph will start on a new page Activity - Each activity description will start on a new page
Different first page	Toggle so that the title page doesn't count as the first page

## Export as HTML site

To generate a process documentation in HTML Format use

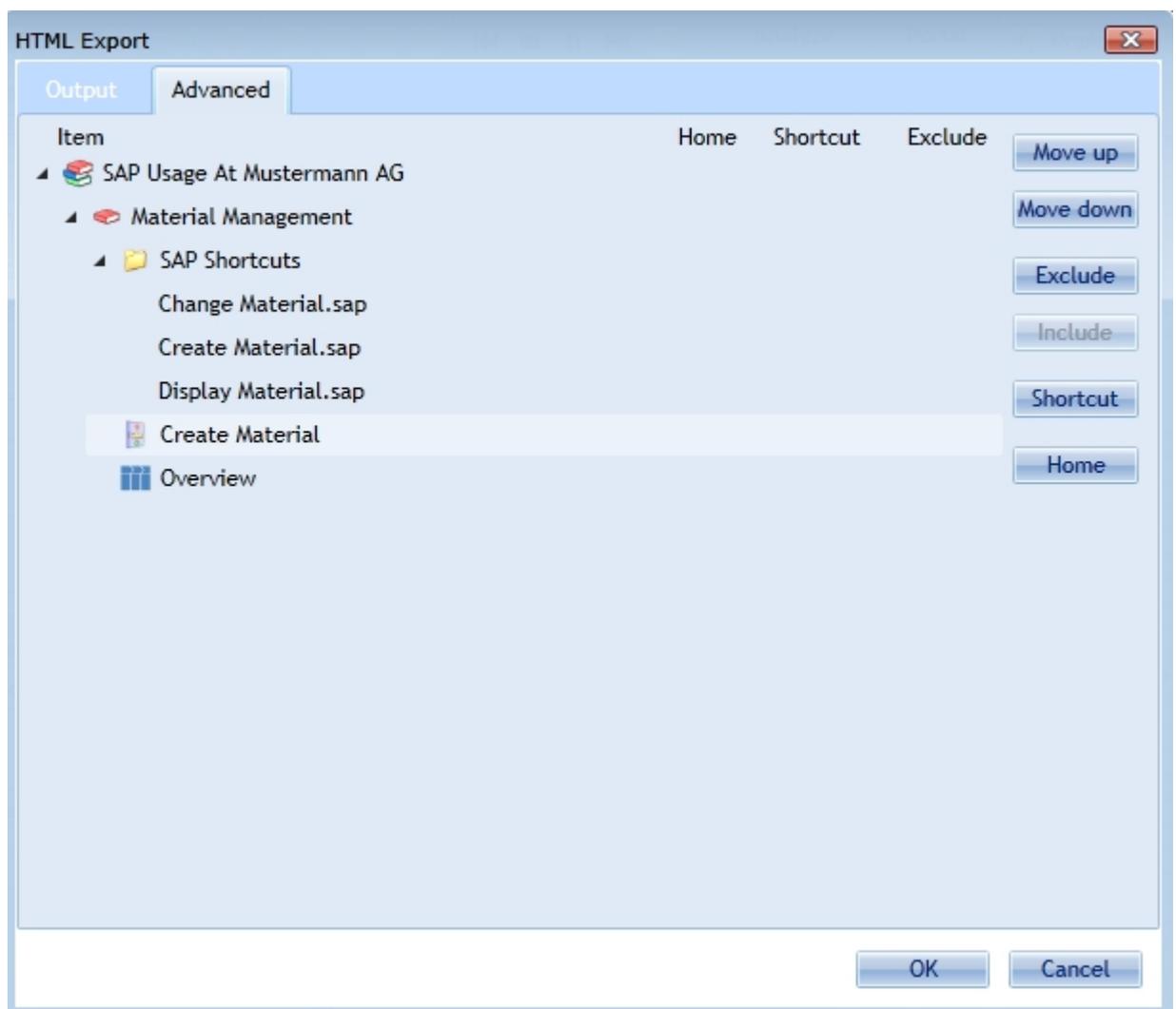
Application Menu -> Export -> as HTML site. This will create complete process documentation from the current BPI project.

Because this export type consists out of multiple files, the export is provided as packed zip file. As soon as you select a Zip container the required zip file will be created. Be aware that this file has to be deleted manually when you abort your export. Since IYOPRO is an web application it isn't granted to delete files automatically from your device.

Only if you use IYOPRO in [OOB mode](#) you may specify a destination directory where to save the data to.



On the advanced tab you have a better control on the items exported as well as their order:



To change the order of Elements click on the element to move and then choose the Move up / Move down buttons to move the element.  
 To exclude nodes from export select the node and click on Exclude. To Include excluded nodes select the node and click on Include.

To specify the Home Page ( the page which will be shown if the documentation is entered) select the node and click Home.

To provide shortcuts to important pages select the node and click on shortcut. You will be asked for the shortcut name.

Shortcuts will be listed in the upper right corner of your documentation.

After creating a process documentation involved projects and folders are marked with an asterisk (for changes). This happens because your export settings are saved. Further HTML exports will initially display these last used settings.

## Export as RACI Diagram

## Assign RACI Information

To assign RACI Information to participants select an Activity and click on the participants property. The [participants editor](#) dialog appears where you may add participants and choose the responsibility for each participant.

The screenshot displays the IYOPRO Premium software interface. The main workspace shows a process diagram with a start node, a task 'Check if material exist', a decision diamond, and a task 'Create Material in SAP'. The 'Advanced Properties' dialog is open, showing the 'ResponsibleUsers' property. The dialog contains a table with the following data:

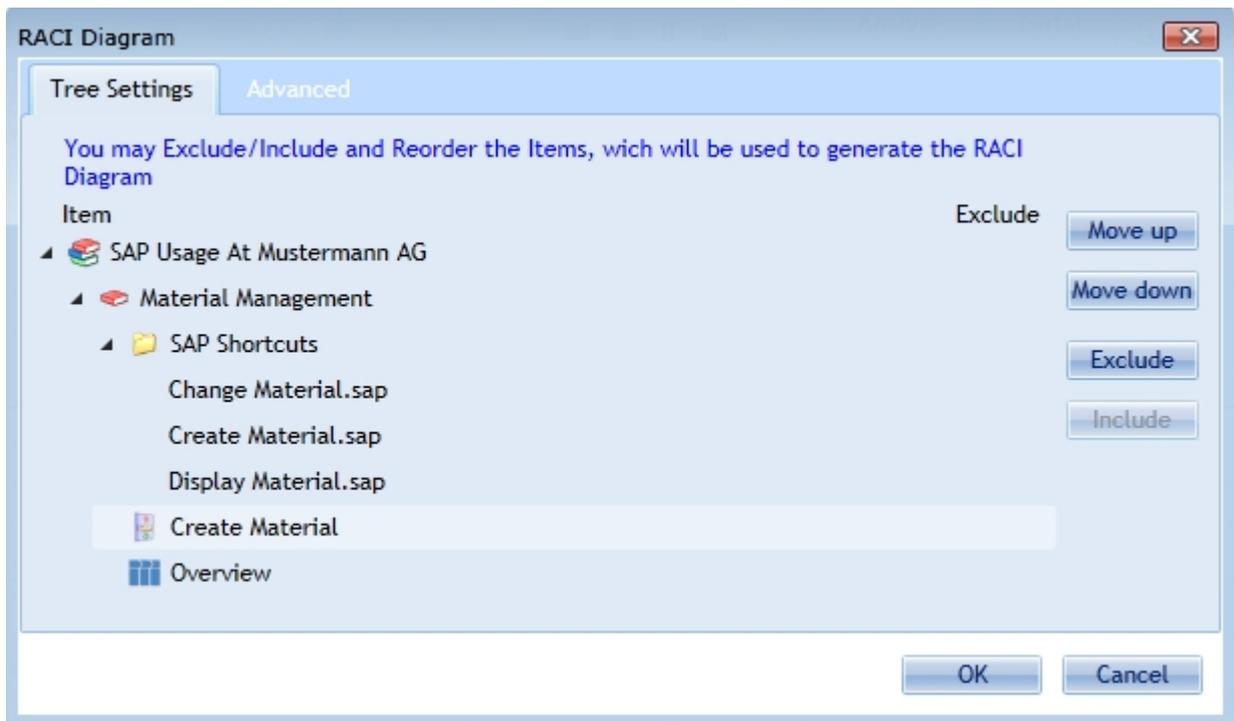
Scope	Name	Value	Participant Name	Responsibility	Amount
Participants			Developer	R	×
Message			Development Manager	C,I	×

## Create RACI Diagram

To generate a RACI diagram use

Application Menu -> Export -> as RACI diagram. This will create a RACI diagram from the current BPI project.

In the upcoming Dialog you may specify the following export options:

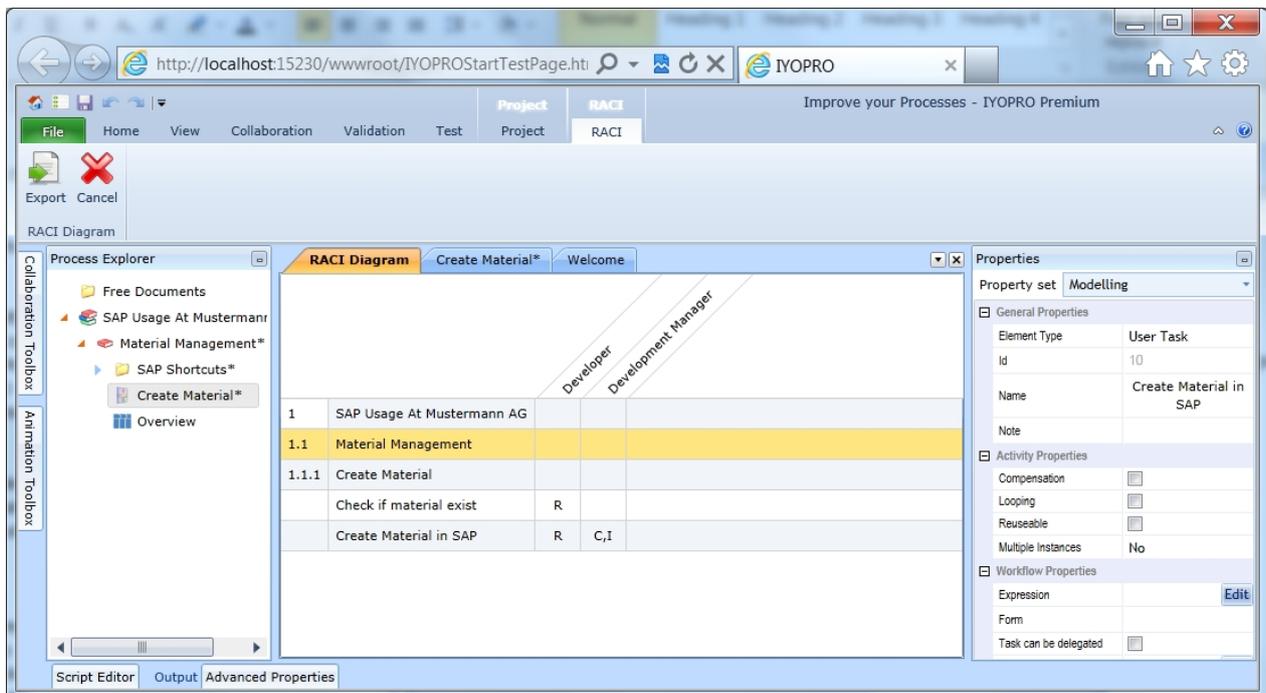


To change the order of Elements click on the element to move and choose the Move up / Move down buttons to move the element.

To exclude nodes from export select the node and click on Exclude. To Include excluded nodes select the node and click on Include.

In the "Advanced" ribbon you are able to decide if you want to resolve the roles according to an organisation diagram.

To proceed click on OK.

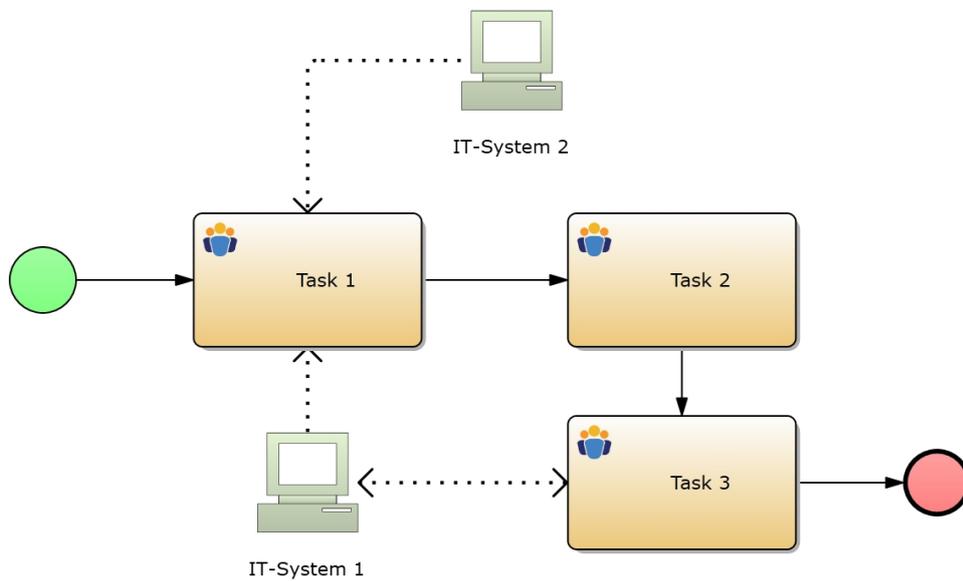


This will show a preview of the RACI diagram. To export the diagram to Excel click on the Export button in the ribbon.

## Export IT System Diagram

### Assign IT-Systems

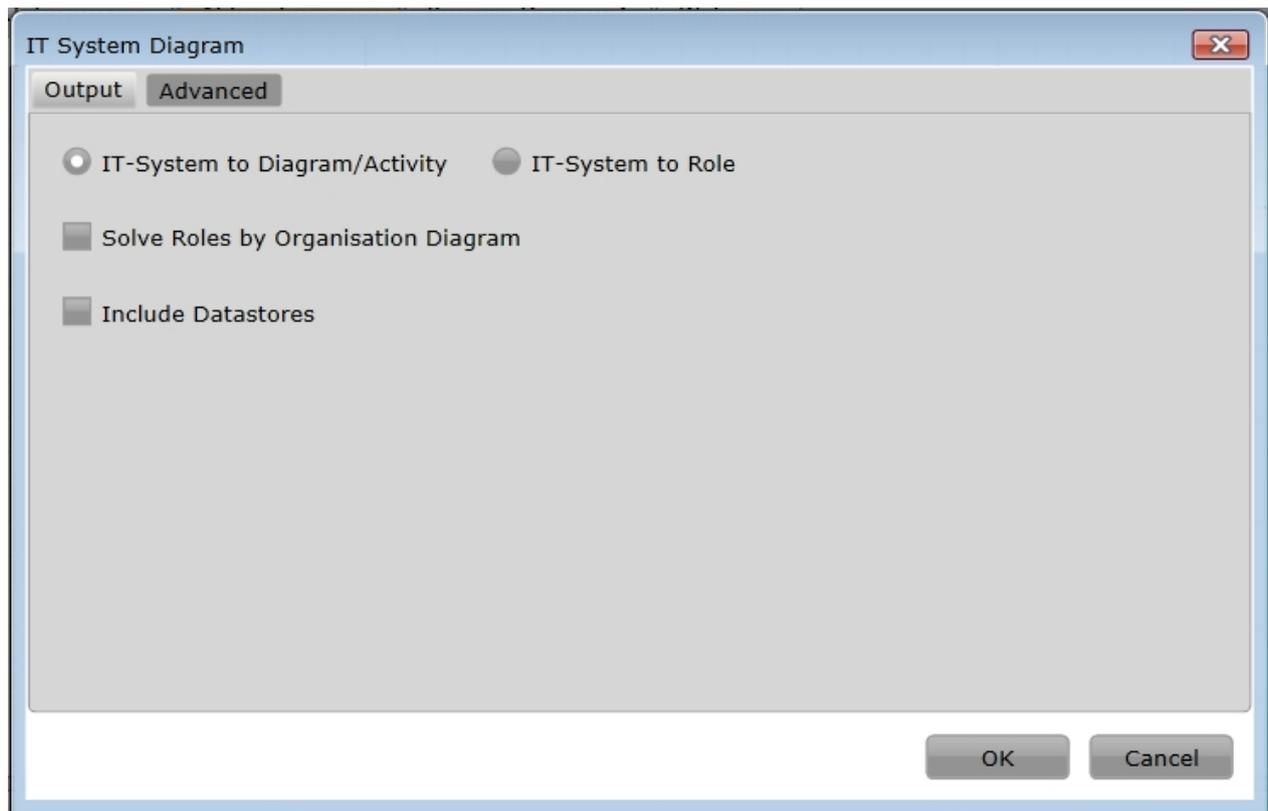
To assign IT-Systems to activities drag the IT-System shape from the Toolbox onto your diagram. The IT-System will automatically link to the nearest activity while dragging.



### Create IT System Diagram

To generate an IT-System diagram use

Application Menu -> Export -> IT System Diagram. In the upcoming Dialog you may specify the following export options:



### **IT-System to Diagram Activity**

This will create a table where the diagrams and activities describe the rows and the IT-Systems describe the columns.

### **IT-System to Role**

This will create a table where the Roles describe the rows and the IT-Systems describe the columns.

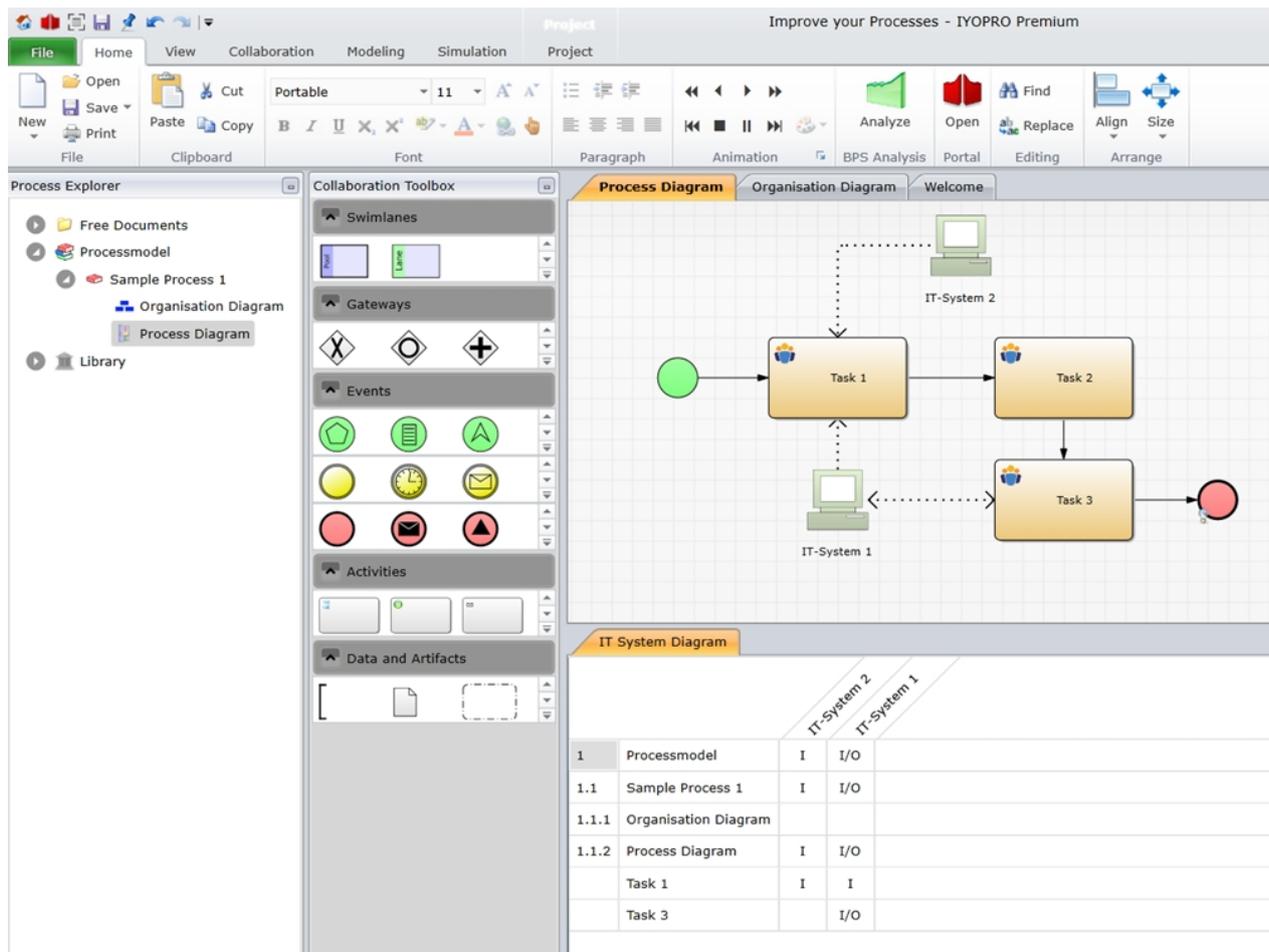
### **Solve Roles by Organization Diagram**

This will solve the roles and map them to real users / groups. There must be an organization diagram linked to the project.

### **Include Datastores**

IT-Systems and Datastores will be considered in the table.

Click OK to create a preview of the table. To Export the table to Excel click on the Export button in the ribbon.



## Export as SVG Image

SVG images provide a lossless re-sizable presentation of the active document. To generate a SVG image use

Application Menu -> Export -> as SVG image.

Note: SVG images present element labels in the default font format. Individual font settings can not be transferred into SVG images because of differences between the graphic specifications.

## Import

### Import from Filesystem

To import a previously exported IYOPRO document from the filesystem use

Application Menu -> Import -> from the Filesystem. This will load the document content into the memory.

# IYOPRO Api Developer Guide

---

## Introduction

The IYOPRO API gives you access to most of the repository elements used in workflow environments.

## Conceptual Overview

The IYOPRO API provides a set of classes which let you read and modify IYOPRO workflow data from within your applications.

First it is essential to establish a session to IYOPRO using the Session class. Once the Session has been established the session object is required for further API calls.

The API allows to enumerate running workflow instances and tasks as well as the creation of new instances.

In general all service calls are asynchronous. In order to receive the requested service result the programmer has to register for the specified event which provides the result data or even an error information.

## System Requirements

The IYOPRO API is based on the .Net Framework 4.0.

The API has been tested with various sample applications written with Visual Studio 2010.

## Download

The IYOPRO API Library and may be downloaded at <https://www.iyopro.com/iyopro/api/IYOPRO.Api.zip>

## API User

If you want to use API calls in your workflow, you need to set an API User in the [Workflow Build Options](#).

## EventArgs classes

### BackendEventArgs class

The BackendEventArgs provides the result of a Backend call.

**Namespace:** IYOPRO.Api

**Assembly:**

IYOPRO.Api (in IYOPRO.Api.dll)

**Y:**

### Syntax

```
public class BackendEventArgs : EventArgs
```

### Members

 <b>Exception</b> Error	If an error occurred this property will contain a reference to the exception, otherwise null
 <b>object</b> UserData	All asynchronous backend calls provide the possibility to attach user data to the call which will be provided in this property on service result.

### CreateWorkflowFinishedEventArgs class

The CreateWorkflowFinishedEventArgs provides the result of a CreateWorkflowFinished() call of the [Session](#) class.

**Namesp** IYOPRO.Api  
**ace:**  
**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)  
**y:**

## Syntax

```
public class CreateWorkflowFinishedEventArgs : BackendEventArgs
```

## Members

 <a href="#">Exception</a> Error	Inherited from <a href="#">BackendEventArgs</a>
 <a href="#">object</a> UserData	Inherited from <a href="#">BackendEventArgs</a>
 <a href="#">Workflow</a> Workflow	Specifies the associated workflow
 <a href="#">ProjectTasks</a> Tasks	Specifies the associated project tasks

## DeleteIstanceCompletedEventArgs

The DeleteIstanceCompletedEventArgs provides the result of a Delete() call of the [ProcessInstance](#) class.

**Namesp** IYOPRO.Api  
**ace:**  
**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)  
**y:**

## Syntax

```
public class DeleteIstanceCompletedEventArgs : BackendEventArgs
```

## Members

 <a href="#">Exception</a> Error	Inherited from <a href="#">BackendEventArgs</a>
 <a href="#">object</a> UserData	Inherited from <a href="#">BackendEventArgs</a>

## GetAccountCompletedEventArgs

The GetAccountCompletedEventArgs provides the result of a GetAccountForSession() call of the [Session](#) class.

**Namesp** IYOPRO.Api  
**ace:**  
**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)  
**y:**

## Syntax

```
public class GetAccountCompletedEventArgs : BackendEventArgs
```

## Members

 <a href="#">Exception</a> Error	Inherited from <a href="#">BackendEventArgs</a>
 <a href="#">object</a> UserData	Inherited from <a href="#">BackendEventArgs</a>
 <a href="#">string</a> UserName	Specifies the name of the user

## GetDownloadUrlCompletedEventArgs

The LoadAttributesCompletedEventArgs provides the result of a GetDownloadUrl() call of the [ProcessInstanceFile](#) class.

**Namespace** IYOPRO.Api

**Assembly:**

**Assembly** IYOPRO.Api (in IYOPRO.Api.dll)

**Y:**

### Syntax

```
public class GetDownloadUrlCompletedEventArgs : BackendEventArgs
```

### Members

 <a href="#">Exception</a> Error	Inherited from BackendEventArgs
 <a href="#">object</a> UserData	Inherited from BackendEventArgs
 <a href="#">string</a> Url	Specifies the url

## GetProjectTasksFinishedEventArgs class

The GetProjectTasksFinishedEventArgs provides the result of a GetProjectTasks() call of the [Session](#) class.

**Namespace** IYOPRO.Api

**Assembly:**

**Assembly** IYOPRO.Api (in IYOPRO.Api.dll)

**Y:**

### Syntax

```
public class GetProjectTasksFinishedEventArgs : BackendEventArgs
```

### Members

 <a href="#">Exception</a> Error	Inherited from BackendEventArgs
 <a href="#">object</a> UserData	Inherited from BackendEventArgs
 <a href="#">List</a> <ProjectTask> Data	List of project tasks

## LoadAttributesCompletedEventArgs class

The LoadAttributesCompletedEventArgs provides the result of a LoadAttributes() call of the [ProcessInstance](#) class.

**Namespace** IYOPRO.Api

**Assembly:**

**Assembly** IYOPRO.Api (in IYOPRO.Api.dll)

**Y:**

### Syntax

```
public class LoadAttributesCompletedEventArgs : BackendEventArgs
```

### Members

 <b>Exception</b> Error	Inherited from BackendEventArgs
 <b>object</b> UserData	Inherited from BackendEventArgs

## LoginCompletedEventArgs class

The LoginCompletedEventArgs provides the result of a Login() call of the [Session](#) class.

**Namespace** YOPRO.Api

**Assembly**

**Assembly** YOPRO.Api (in YOPRO.Api.dll)

**Y:**

## Syntax

```
public class LoginCompletedEventArgs : BackendEventArgs
```

## Members

 <b>Exception</b> Error	Inherited from BackendEventArgs
 <b>object</b> UserData	Inherited from BackendEventArgs

## ProcessContinuedEventArgs class

The ProcessContinuedEventArgs provides the result of a Run() call of the [Task](#) class.

**Namespace** YOPRO.Api

**Assembly**

**Assembly** YOPRO.Api (in YOPRO.Api.dll)

**Y:**

## Syntax

```
public class ProcessContinuedEventArgs : BackendEventArgs
```

## Members

 <b>Exception</b> Error	Inherited from BackendEventArgs
 <b>object</b> UserData	Inherited from BackendEventArgs

## ProcessInstanceLoadedEventArgs class

The ProcessInstanceLoadedEventArgs provides the result of a Load() call of the [ProcessInstances](#) class.

**Namespace** YOPRO.Api

**Assembly**

**Assembly** YOPRO.Api (in YOPRO.Api.dll)

**Y:**

## Syntax

```
public class ProcessInstanceLoadedEventArgs : BackendEventArgs
```

## Members

 <a href="#">Exception</a> Error	Inherited from BackendEventArgs
 <a href="#">object</a> UserData	Inherited from BackendEventArgs
 <a href="#">ProcessInstance</a> Instance	Specifies the associated process instance

## ProcessInstanceListLoadedEventEventArgs class

The ProcessInstanceListLoadedEventEventArgs provides the result of a GetProcessInstances() call of the [Workflow](#) class.

**Namespace** IYOPRO.Api

**Assembly:**

**Assembly** IYOPRO.Api (in IYOPRO.Api.dll)

**Y:**

## Syntax

```
public class ProcessInstanceListLoadedEventEventArgs : BackendEventArgs
```

## Members

 <a href="#">Exception</a> Error	Inherited from BackendEventArgs
 <a href="#">object</a> UserData	Inherited from BackendEventArgs
 <a href="#">List</a> < <a href="#">ProcessInstance</a> > Data	List of found process instances

## ProcessInstancesLoadedEventEventArgs class

The ProcessInstancesLoadedEventEventArgs provides the result of a Load() call of the [ProcessInstances](#) class.

**Namespace** IYOPRO.Api

**Assembly:**

**Assembly** IYOPRO.Api (in IYOPRO.Api.dll)

**Y:**

## Syntax

```
public class ProcessInstancesLoadedEventEventArgs : BackendEventArgs
```

## Members

 <a href="#">Exception</a> Error	Inherited from BackendEventArgs
 <a href="#">object</a> UserData	Inherited from BackendEventArgs
 <a href="#">List</a> < <a href="#">ProcessInstance</a> > Data	List of found process instances

## ProcessStartedEventArgs class

The ProcessStartedEventArgs provides the result of a Run() call of the [ProcessInstance](#) class.

**Namespace** IYOPRO.Api

**Assembly:**

**Assembly** IYOPRO.Api (in IYOPRO.Api.dll)

y:

## Syntax

```
public class ProcessStartedEventArgs : BackendEventArgs
```

## Members

 <a href="#">Exception</a> Error	Inherited from BackendEventArgs
 <a href="#">object</a> UserData	Inherited from BackendEventArgs
 <a href="#">Workflow</a> Workflow	Specifies the associated workflow

## ReadLogCompletedEventArgs class

The ReadLogCompletedEventArgs provides the result of a ReadLog() call of the [ProcessInstance](#) class.

**Namesp** IYOPRO.Api

**ace:**

**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)

y:

## Syntax

```
public class ReadLogCompletedEventArgs : BackendEventArgs
```

## Members

 <a href="#">Exception</a> Error	Inherited from BackendEventArgs
 <a href="#">object</a> UserData	Inherited from BackendEventArgs
 <a href="#">ProcessInstanceLog</a> Log	Specifies the associated process instance log

## TasksLoadedEventArgs class

The TasksLoadedEventArgs provides the result of a Load() call of the [Tasks](#) class.

**Namesp** IYOPRO.Api

**ace:**

**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)

y:

## Syntax

```
public class TasksLoadedEventArgs : BackendEventArgs
```

## Members

 <a href="#">Exception</a> Error	Inherited from BackendEventArgs
 <a href="#">object</a> UserData	Inherited from BackendEventArgs
 <a href="#">List&lt;Task&gt;</a> Data	The list of tasks found

## TaskUpdatedEventArgs class

The TaskUpdatedEventArgs provides the result of a Update() call of the [Task](#) class.

**Namesp** YOPRO.Api

**ace:**

**Assembl** YOPRO.Api (in YOPRO.Api.dll)

**y:**

## Syntax

```
public class TaskUpdatedEventArgs : BackendEventArgs
```

## Members

 <a href="#">Exception</a> Error	Inherited from BackendEventArgs
 <a href="#">object</a> UserData	Inherited from BackendEventArgs

## WorkflowFormLoadedEventEventArgs class

The WorkflowFormLoadedEventEventArgs provides the result of a LoadStartForm() call of the [Workflows](#) class.

**Namesp** YOPRO.Api

**ace:**

**Assembl** YOPRO.Api (in YOPRO.Api.dll)

**y:**

## Syntax

```
public class WorkflowFormLoadedEventEventArgs : BackendEventArgs
```

## Members

 <a href="#">Exception</a> Error	Inherited from BackendEventArgs
 <a href="#">object</a> UserData	Inherited from BackendEventArgs
 <a href="#">string</a> FormXml	Specifies the XML of the loaded form

## WorkflowLoadedEventEventArgs class

The WorkflowLoadedEventEventArgs provides the result of a Load() call of the [Workflows](#) class.

**Namesp** YOPRO.Api

**ace:**

**Assembl** YOPRO.Api (in YOPRO.Api.dll)

**y:**

## Syntax

```
public class WorkflowLoadedEventEventArgs : BackendEventArgs
```

## Members

 <a href="#">Exception</a> Error	Inherited from BackendEventArgs
 <a href="#">object</a> UserData	Inherited from BackendEventArgs
 <a href="#">Workflow</a> Workflow	Specifies the associated workflow

## WorkflowsLoadedEventEventArgs class

The WorkflowsLoadedEventEventArgs provides the result of a Load() call of the [Workflows](#) class.

**Namespace:** IYOPRO.Api

**Assembly:**

IYOPRO.Api (in IYOPRO.Api.dll)

**Public:**

### Syntax

```
public class WorkflowsLoadedEventEventArgs : BackendEventArgs
```

### Members

 <b>Exception</b> Error	Inherited from BackendEventArgs
 <b>object</b> UserData	Inherited from BackendEventArgs
 <b>List</b> <Workflow> Workflows	List of found workflows

## Session class

The creation of a Session object is always the first step in the communication between a client application and the IYOPRO backend. It is required for all following API calls and objects.

**Namespace:** IYOPRO.Api

**Assembly:**

IYOPRO.Api (in IYOPRO.Api.dll)

**Public:**

### Syntax

```
class Session : Util.APIEventBase
```

### Members

 <b>Session</b> (object wrapper, bool throwExceptions = true)	Initializes a new instance of the Session class with either a provided workflow or process instance. If you set the throwExceptions parameter to False, the Error attribute can be queried in the return value of the API. Note: If executed on backend
 <b>Session</b> (string servicepath, string username, string password)	Initializes a new instance of the Session class. The servicepath specifies the location of the IYOPRO backend webservice. For the SaaS version of IYOPRO you must specify <code>https://www.iyopro.com/iyopro/</code> Username and password represent the credentials which were used on user registration
 <b>void</b> Login(object Userdata=null)	Performs the Login process. You may provide user data which will be provided in the resulting <a href="#">LoginCompletedEventEventArgs</a>
 <b>void</b> Login(Guid sessionId, object Userdata=null)	Validate session id. Perform this operation to validate if a previously received session is still valid
 <b>void</b> AutoLogin(object userdata = null)	Perform automatic login. This method uses NTLM to perform the login at the IYOPRO Server

 <code>bool</code> IsLoggedIn()	Returns true if the user was logged in, false otherwise. This method does not verify if the session is still valid
 <code>void</code> Logout()	Performs the Logout process
 <code>string</code> GetServiceUrl()	Gets the service URL that is used to connect to IYOPRO
 <code>string</code> GetNtlmServiceUrl()	Gets the service URL that is used to connect to IYOPRO with NTLM credentials
 <code>string</code> GetWebSocketServiceUrl()	Gets the service URL that is used to connect to IYOPRO via WebSocket
 <code>void</code> GetProjectTasks( <code>int</code> workflowid, <code>object</code> userdata = <code>null</code> )	Read all tasks of a workflow definition which are marked as project task. Userdata will be provided in the <a href="#">GetProjectTasksFinishedEventArgs</a>
 <code>void</code> CreateWorkflow( <code>ProjectTasks</code> tasks, <code>Roles</code> roles, <code>object</code> userdata = <code>null</code> )	Creates a workflow from a given hierarchy of tasks and roles. Userdata will be provided in the <a href="#">CreateWorkflowFinishedEventArgs</a>
 <code>void</code> GetAccountForSession( <code>Guid</code> sessionId, <code>object</code> userdata = <code>null</code> )	Gets the account for the specified session id. Userdata will be provided in the <a href="#">GetAccountCompletedEventArgs</a>
 <code>string</code> UserName	Gets the user name which was specified in the constructor
 <code>string</code> ServicePath	Gets the service url which was specified in the constructor
 <code>Guid</code> SessionId	Gets the session id which is provided by the Login() method
 <code>Workflows</code> Workflows	Gets a reference to the Workflows class which provides managing workflow objects.
 <code>Tasks</code> Tasks	Gets a reference to the Tasks class which provides managing task objects.
 <code>ProcessInstances</code> ProcessInstances	Gets a reference to the ProcessInstances class which provides managing process instance objects
 <code>event</code> LoginCompletedEvent LoginCompleted	The LoginCompleted event is triggered if the Login has finished
 <code>event</code> GetProjectTasksFinishedEvent GetProjectTasksFinished	The GetProjectTasksFinished event is triggered if the GetProjectTasks method has finished
 <code>event</code> CreateWorkflowFinishedEvent CreateWorkflowFinished	The CreateWorkflowFinished event is triggered if the CreateWorkflow method has finished
 <code>event</code> GetAccountCompletedEvent GetAccountCompleted	The GetAccountCompleted event is triggered if the GetAccountForSession method has finished

A typical way to log in is

1. Verify if a previously received sessionid is still valid To do this use `void Login(Guid sessionId,object Userdata=null)`
2. If the sessionid is not valid try AutoLogin `void AutoLogin(object userdata = null)`
3. If Autologin failed ask for Username and password and perform a manual login `Session(string servicepath, string username, string password)`

### Sample to log in

```
Session session = new Session("https://www.iyopro.com/iyopro/" , "username" , "password")
;
session.LoginCompleted += (s, args) =>
{
    if (args.Error != null)
    { string msg=args.Error.Message;
      MessageBox.Show(msg);
      return;
    }
}
```

```

    }
};
session.Login();

```

## Workflows class

The Workflows object specifies a collection of loaded workflow objects. An instance of the Workflows collection is provided by the [Session](#) object.

**Namesp** YOPRO.Api

**ace:**

**Assembl** YOPRO.Api (in YOPRO.Api.dll)

**y:**

## Syntax

```
class Workflows()
```

## Members

 <code>void Load(int wfid, object userdata = null)</code>	Retrieves a Workflow definition. Wfid specifies the id of the workflow. Userdata will be provided in the <a href="#">WorkflowLoadedEventArgs</a>
 <code>void Load(string tag, object userdata = null)</code>	Retrieves a Workflow definition. Userdata will be provided in the <a href="#">WorkflowsLoadedEventArgs</a>
 <code>void LoadStartForm(int wfid, object userdata = null)</code>	Retrieves a Workflow definition. Wfid specifies the id of the workflow. Userdata will be provided in the <a href="#">WorkflowFormLoadedEventArgs</a>
 <code>event WorkflowLoadedEvent WorkflowLoaded</code>	The WorkflowLoaded event is triggered if the Load method has finished
 <code>event WorkflowsLoadedEvent WorkflowsLoaded</code>	The WorkflowsLoaded event is triggered if the Load method has finished
 <code>event WorkflowFormLoadedEvent WorkflowFormLoaded</code>	The WorkflowFormLoaded event is triggered if the Load method has finished

## Sample

```

// Load Workflow definition
Workflows wf = session.Workflows;

Workflows.WorkflowLoadedEvent handler = null;
handler = (s, args) =>
{
    wf.WorkflowLoaded -= handler;
    Workflow w = args.Workflow;
    if (w == null) return;
};
wf.WorkflowLoaded += handler;
wf.Load(workflowid);

```

## Workflow class

The Workflow object represents a workflow definition.

**Namesp** YOPRO.Api

**ace:**

Assembly: IYOPRO.Api (in IYOPRO.Api.dll)

Namespace:

## Syntax

```
class Workflow()
```

## Members

 <b>ProcessInstances</b> CreateProcessInstance()	Creates a new process instance based on the workflow definition
 <b>void</b> GetProcessInstances(List<string> names, DateTime startDate, DateTime endDate, ProcessInstanceState states, List<string> attributes, object userdata = null)	Retrieves process instances for the given workflow. Names is an optional filter for process instance names. StartDate and endDate are filters and specify the time range for the finishedDate of these process instances, which are finished. States specifies, at what state the retrieved instances must be. Attributes specify a list of process instance variable names, which should be retrieved. UserData will be provided in the <a href="#">ProcessInstanceListLoadedEventArgs</a>
 <b>int</b> Id	Gets the Id of the workflow
 <b>string</b> Name	Gets the name of the workflow
 <b>string</b> PlainDescription	Gets the description of the Workflow
 <b>event</b> ProcessInstanceListLoadedEvent ProcessInstanceListLoaded;	The ProcessInstanceListLoaded event is triggered if the GetProcessInstances method has finished

## Sample

```
// Load Workflow definition and create process instance
Workflows wf = session.Workflows;

// STEP 1: Load the Workflow
Workflows.WorkflowLoadedEvent handler = null;
handler = (s, args) =>
{
    wf.WorkflowLoaded -= handler;
    Workflow w = args.Workflow;
    if (w == null) return;

    // STEP 2: Create a Process Instance
    ProcessInstance pi = w.CreateProcessInstance();
    pi.Attributes.SetValue("email", "noreply@iyopro.com");

    // STEP 3: Run the Process Instance
    pi.Run();
};
wf.WorkflowLoaded += handler;
wf.Load(workflowid);

// Load process instances of that workflow

Workflows wf = session.Workflows;
Workflows.WorkflowLoadedEvent handler = null;
handler = (s, args) =>
```

```

{
    wf.WorkflowLoaded -= handler;
    Workflow w = args.Workflow;
    Workflow.ProcessInstanceListLoadedEvent wlhandler = null;
    wlhandler = (wls, wle) =>
    {
        w.ProcessInstanceListLoaded -= wlhandler;
        List<ProcessInstance> instances = wle.Data;
        ...

    };
    w.ProcessInstanceListLoaded += wlhandler;
    // Load only process instances with the name "my process instance name"
    List<string> names=new List<string>(){ "my process instance name"};
    w.GetProcessInstances(names, DateTime.MinValue, DateTime.MaxValue,
ProcessInstanceState.All, new List<string>() { "email", "myuuid"});
};
wf.WorkflowLoaded += handler;
wf.Load(workflowid);

```

## ProcessInstances class

The ProcessInstances object specifies a collection of loaded ProcessInstances objects. An instance of the ProcessInstances collection is provided by the [Session](#) object.

**Namesp** IYOPRO.Api

**ace:**

**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)

**y:**

## Syntax

```
class ProcessInstances()
```

## Members

 <b>void</b> Load(Guid piid, object userdata = null)	Retrieves the process instance for the given id. Userdata will be provided in the <a href="#">ProcessInstanceLoadedEventArgs</a>
 <b>void</b> Load(DateTime startDate, DateTime endDate, ProcessInstanceState state, List<string> names, List<string> variables, object userdata = null)	Provides a list of the available process instances Userdata will be provided in the <a href="#">ProcessInstancesLoadedEventArgs</a>
 <b>event</b> ProcessInstanceLoadedEvent ProcessInstanceLoaded	The ProcessInstanceLoaded event is triggered if the Load method has finished
 <b>event</b> ProcessInstancesLoadedEvent ProcessInstancesLoaded	The ProcessInstancesLoaded event is triggered if the Load method has finished

## Sample

```

// Load the process instance with the specified id
Session session = ...;
Guid myid = ...;

ProcessInstances.ProcessInstanceLoadedEvent handler = null;
handler = (s, args) =>
{
    session.ProcessInstances.ProcessInstanceLoaded -= handler;
    Data = args.Data;

```

```

    ...
};
session.ProcessInstances.ProcessInstanceLoaded += handler;
session.ProcessInstances.Load(myid);

```

This sample is a script that can be run directly in IYOPRO. It shows how to load process instances that have a specific name and status and are within a specific time period

```

#IYOPRO.Backend
import clr
clr.AddReference("ServerBackend")
import IYOPRO.Api

import System

s = IYOPRO.Api.Session(Workflow)

start = System.DateTime(2020,1,1)
end = System.DateTime(2021,12,31)
names = System.Collections.Generic.List[str](["<process instance name>"])
vars = System.Collections.Generic.List[str]([])
e = s.ProcessInstances.Load
(start,end,IYOPRO.Api.ProcessInstanceState.Finished,names,vars)

```

## ProcessInstance class

The ProcessInstance object represents a process instance of a workflow. A process instance does not have a public constructor. It can be created using the CreateProcessInstance method of the [Workflow](#) class.

**Namesp** IYOPRO.Api

**ace:**

**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)

**y:**

## Syntax

```
class ProcessInstance()
```

## Members

 <b>ProcessInstanceFile</b> AddFile( <b>string</b> name, <b>string</b> path)	Add a local file to the process instance. Keep in mind that you should also assign the internal filename to a process instance variable to be able to access the file later
 <b>void</b> RemoveFile( <b>string</b> internalname)	Removes a file from the process instance. Remember to remove the file also from the referencing process instance variable
 <b>void</b> Run( <b>object</b> userdata = null)	Continue / Start the process instance. User data will be provided in the resulting <a href="#">ProcessStartedEventArgs</a>
 <b>void</b> LoadAttributes( <b>List</b> < <b>string</b> > attributes, <b>object</b> userdata=null)	Load process instance attribute values and append them to the Attributes collection. Userdata will be provided in the <a href="#">LoadAttributesCompletedEventArgs</a>
 <b>void</b> ReadLog( <b>bool</b> deep=false, <b>object</b> userdata=null)	Read the process instance log. Specify with "deep" whether logs from subprocesses should also be loaded. User data will be provided in the <a href="#">ReadLogCompletedEventArgs</a>
 <b>void</b> Delete( <b>object</b> userdata = null)	Delete the process instance. User data will be provided in the resulting <a href="#">DeleteIstanceCompletedEventArgs</a>
 <b>int</b> Id	Gets the Id of the process instance

 <b>Guid</b> GuId	Gets the unique identifier of the process instance
 <b>string</b> Name	Gets the name of the process instance
 <b>int</b> WorkflowId	Gets the Id of the describing Workflow
 <b>int</b> WorkflowFileId	Gets the Id of the workflow file
 <b>string</b> WorkflowName	Gets the name of the describing Workflow
 <b>int</b> WorkflowVersion	Gets the version of the workflow
 <b>string</b> WorkflowVersionName	Gets the name of the workflow version
 <b>ProcessInstanceState</b> State	Gets the state of the process instance
 <b>string</b> InstanceInfo	Gets the information of the process instance
 <b>string</b> CreatedBy	Gets the name of the process instance creator
 <b>DateTime</b> CreatedAt	Gets the timestamp when the process instance has been created
 <b>DateTime</b> FinishedAt	Gets the timestamp when the process instance is finished
 <b>ProcessInstanceAttributes</b> Attributes	Gets a collection of process instance attributes
 <b>TailoredActivities</b> Tailor	Gets a collection of tailored activities
 <b>Roles</b> Roles	Gets a collection of roles defined in the process instance
 <b>event</b> <b>ProcessStartedEvent</b> ProcessStarted	The ProcessStarted event is triggered if the Run method has finished
 <b>event</b> <b>LoadAttributesCompletedEvent</b> LoadAttributesCompleted	The LoadAttributesCompleted event is triggered if the LoadAttributes method has finished
 <b>event</b> <b>ReadLogCompletedEvent</b> ReadLogCompleted	The ReadLogCompleted event is triggered if the ReadLog method has finished
 <b>event</b> <b>DeleteInstanceCompletedEvent</b> DeleteInstanceCompleted	The DeleteInstanceCompleted event is triggered if the Delete method has finished

## Sample

```
// Add the Pdf to our process instance
ProcessInstanceFile pif=Task.AddFile("My Documentation", "c:\\temp\\File.pdf");
// Add the Pdf to our process attribute, that contains the file list
List<object> files = Task.Attributes.Get("Files") as List<object>;
files.Add(pif.InternalName);
// Finish the task
Task.Run(true);
```

This sample shows how to delete process instances that have a specific name and status and are within a specific time period

```
#IYOPRO.Backend
import clr
clr.AddReference("ServerBackend")
import IYOPRO.Api

import System

s = IYOPRO.Api.Session(Workflow, False)

start = System.DateTime(2020,1,1)
end = System.DateTime(2021,12,31)
names = System.Collections.Generic.List[str](["<process instance name>"])
vars = System.Collections.Generic.List[str]()
e = s.ProcessInstances.Load
(start,end,IYOPRO.Api.ProcessInstanceState.Finished,names,vars)
```

```

if e.Error:
    raise e.Error
for inst in e.Data:
    ProcessInstance.Info("Removing instance " + unicode(inst.Id))
    d = inst.Delete()
    if d.Error:
        raise d.Error

```

## ProcessInstanceAttributes class

The ProcessInstanceAttributes object represents a collection of Process Instance Attributes. An instance of the ProcessInstanceAttributes collection is provided by the [ProcessInstance](#) object.

**Namesp** IYOPRO.Api

**ace:**

**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)

**y:**

### Syntax

```
class ProcessInstanceAttributes : ObservableCollection<ProcessInstanceAttribute>
```

### Members

 <code>ProcessInstanceAttributes()</code>	Constructor to initialize a ProcessInstanceAttributes instance
 <code><a href="#">ProcessInstanceAttribute</a> Get(<a href="#">string</a> name)</code>	Gets a process instance attribute with the given name
 <code><a href="#">object</a> GetValue(<a href="#">string</a> name)</code>	Gets the value of a process instance attribute with the given name
 <code>void SetValue(<a href="#">string</a> name, <a href="#">object</a> value)</code>	Sets the value of a process instance attribute with the given name

## ProcessInstanceAttribute class

The ProcessInstanceAttribute object represents a process instance attribute.

**Namesp** IYOPRO.Api

**ace:**

**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)

**y:**

### Syntax

```
class ProcessInstanceAttribute()
```

### Members

 <code><a href="#">string</a> ToString()</code>	Overwritten method to provide a valuable string representation of the value
 <code><a href="#">string</a> Name</code>	Gets the name of the attribute
 <code><a href="#">object</a> Value</code>	Gets or sets the value of the attribute

## ProcessInstanceFile class

The ProcessInstanceFile object represents a file which is part of a process instance. This class provides two main operations:

1. Get the download url to provide the file content

## 2. Prepare an upload operation to store file content

**Namespace** IYOPRO.Api

**Access:**

**Assembly** IYOPRO.Api (in IYOPRO.Api.dll)

**Category:**

### Syntax

```
class ProcessInstanceFile()
```

### Members

 <code>ProcessInstanceFile(Task task, string internalname)</code>	Constructor to initialize a <code>GetDownloadUrl</code> operation for a given task. The Task contains the process instance for which the file should be provided. Internalname specifies the file to download
 <code>ProcessInstanceFile(ProcessInstance pi, string internalname)</code>	Constructor to initialize a <code>GetDownloadUrl</code> operation. Pi specifies the process instance for which the file should be provided. Internalname specifies the file to download
 <code>ProcessInstanceFile(string name, string path)</code>	Constructor for an upload operation. Name specifies the displayed name of the file, path the path where the file can be located
 <code>void GetDownloadUrl(object userdata = null)</code>	Provides the download url where the file content may be downloaded. Userdata will be provided in the <a href="#">GetDownloadUrlCompletedEventEventArgs</a>
 <code>string ToString()</code>	Overwritten method to provide a valuable string representation of the value
 <code>string Name</code>	Gets the name of the file
 <code>string InternalName</code>	Gets the internal name of the file. IYOPRO ensures, that the internal name is unique. Where as files with the same file names may be uploaded by different tasks of the same process instance the internal name is always different
 <code>GetDownloadUrlCompletedEvent</code> <code>GetDownloadUrlCompleted</code>	The completed event is triggered if <code>GetDownloadUrl</code> has finished

### Sample

Download sample

```
ProcessInstanceFile file = new ProcessInstanceFile(Task, internalname);
ProcessInstanceFile.GetDownloadUrlCompletedEvent handler = null;
handler = (s, args) =>
{
    file.GetDownloadUrlCompleted -= handler;
    string url = "https://www.iyopro.com/iyopro/" + args.Url;
    System.Diagnostics.Process.Start(url);
};
file.GetDownloadUrlCompleted += handler;
file.GetDownloadUrl();
```

Upload sample

```
// Add the Pdf to our process instance
ProcessInstanceFile pif=Task.AddFile("My Documentation", "c:\\temp\\File.pdf");
// Add the Pdf to our process attribute, that contains the file list
```

```
List<object> files = Task.Attributes.Get("Files") as List<object>;
files.Add(pif.InternalName);
// Finish the task
Task.Run(true);
```

## ProcessInstanceState enum

The ProcessInstanceState enumeration specifies the state of a [process instance](#). It is flagged, so multiple states may be specified.

**Namespace** YOPRO.Api

**ace:**

**Assembl** YOPRO.Api (in YOPRO.Api.dll)

**y:**

## Members

Created	Process instance has been created but has never been started
Running	Process instance is running
Finished	Process instance is finished
Error	Process instance has an error
Delayed	Process instance execution has been delayed
All	All above

## Tasks class

The Tasks object specifies a collection of loaded task objects. An instance of the Tasks collection is provided by the [Session](#) object.

**Namespace** YOPRO.Api

**ace:**

**Assembl** YOPRO.Api (in YOPRO.Api.dll)

**y:**

## Syntax

```
class Tasks()
```

## Members

 <b>void</b> Load(DateTime startDate, DateTime endDate, Dictionary<int, List<int>> activities, List<string> attributes, object userdata = null)	Retrieves tasks for the given user. Names is an optional filter for process instance names. StartDate and endDate are filters and specify the time range for the finishedDate of these tasks, which are finished. Activities filter specifies a list of Activity control ids (out of the diagrams) which are requested. Attributes specify a list of process instance variable names, which should be retrieved. Userdata will be provided in the <a href="#">TasksLoadedEventArgs</a>
 <b>event</b> TasksLoadedEvent TasksLoaded	The TasksLoaded event is triggered if the Load method has finished.

## Sample

```
// Load list of tasks
Session session = ...;
```

```

Tasks.TasksLoadedEvent handler = null;
handler = (s, args) =>
{
    session.Tasks.TasksLoaded -= handler;
    Data = args.Data;
    ...
};
session.Tasks.TasksLoaded += handler;
public Dictionary<int, List<int>> filter = // This is a filter. Here you specify, what
tasks should be retrieved
    new Dictionary<int, List<int>>() {
        { 11175, // This is the Diagram Id that holds the Task.
          new List<int>() { 72 } // This is a list of Task Id's inside the diagram.
        }
    };
// Skip all finished tasks and retrieve process variables Number and FormData
session.Tasks.Load(DateTime.UtcNow, DateTime.MaxValue, filter, new List<string>()
{ "Number", "FormData" });

```

## Task class

The Task object represents a workflow task which is triggered usually by a BPMN User Task Activity. because it will be created by the workflow the task class does not have a public constructor. To access available tasks use the [Tasks](#) collection class.

**Namespace** YOPRO.Api

**Assembly**

YOPRO.Api (in YOPRO.Api.dll)

**Category**

## Syntax

```
class Task()
```

## Members

 <b>ProcessInstanceFile</b> AddFile( <b>string</b> name, <b>string</b> path)	Add a local file to the process instance of the task. Keep in mind that you should also assign the internal filename to a process instance variable to be able to access the file later
 <b>void</b> RemoveFile( <b>string</b> internalname)	Removes a file from the process instance. Remember to remove the file also from the referencing process instance variable
 <b>APITaskPriority</b> PriorityFromString( <b>string</b> priority)	Gets the priority number to the given priority string
 <b>void</b> Run( <b>bool</b> finished, <b>object</b> userdata = <b>null</b> )	Continue the process instance. Finished specifies, if the task has been finished or not. Userdata will be provided in the resulting <a href="#">ProcessContinuedEventArgs</a>
 <b>void</b> Update( <b>object</b> userdata= <b>null</b> )	Updates the task. Userdata will be provided in the <a href="#">TaskUpdatedEventArgs</a>
 <b>void</b> LoadAttributes( <b>List&lt;string&gt;</b> attributes, <b>object</b> userdata= <b>null</b> )	Loads process instance attributes by their name and assigns them to the Attributes collection. Userdata will be provided in the <a href="#">LoadAttributesCompletedEventArgs</a>
 <b>Guid</b> GuId	Gets the unique identifier of the task
 <b>Guid</b> PiGuId	Gets the unique identifier of the process instance
 <b>string</b> PiName	Gets the name of the process instance
 <b>string</b> PiDetails	Gets the details of the process instance
 <b>int</b> WfId	Gets the Id of the describing Workflow

 <code>string</code> Name	Gets the name of the task
 <code>DateTime</code> CreatedAt	Gets the timestamp when the task has been created
 <code>DateTime</code> PlannedAt	Gets the timestamp when the task is planned
 <code>DateTime</code> FinishedAt	Gets the timestamp when the task is finished
 <code>string</code> FinishedBy	Gets the name who finished the task
 <code>ProcessInstanceAttributes</code> Attributes	Gets a collection of process instance attributes
 <code>APITaskPriority</code> Priority	Gets the priority of the task
 <code>event LoadAttributesCompletedEvent</code> <code>LoadAttributesCompleted</code>	The <code>LoadAttributesCompleted</code> event is triggered if the <code>LoadAttributes</code> method has finished
 <code>event ProcessContinuedEvent</code> <code>ProcessContinued</code>	The <code>ProcessContinued</code> event is triggered if the <code>Run</code> method has finished
 <code>event UpdatedEvent</code> Updated	The <code>LoginCompleted</code> event is triggered if the <code>Update</code> method has finished

## Syntax

```
public enum APITaskPriority
```

## Members

 <code>VeryHigh</code> = -6
 <code>High</code> = -5
 <code>Medium</code> = -4
 <code>Low</code> = -3
 <code>VeryLow</code> = -2
 <code>None</code> = -1

## Sample

```
// Add the Pdf to our process instance
ProcessInstanceFile pif=Task.AddFile("My Documentation", "c:\\temp\\File.pdf");
// Add the Pdf to our process attribute, that contains the file list
List<object> files = Task.Attributes.Get("Files") as List<object>;
files.Add(pif.InternalName);
// Finish the task
Task.Run(true);
```

## Roles class

The Roles object represents a collection of roles. An instance of the Roles collection is provided by the [ProcessInstance](#) object.

**Namesp** `IYOPRO.Api`

**ace:**

**Assembl** `IYOPRO.Api` (in `IYOPRO.Api.dll`)

**y:**

## Syntax

```
class Roles : ObservableCollection<Role>
```

## Members

 <code>public Role</code> GetRole( <code>string</code> role)	Gets the role by the given name
---	---------------------------------

 <code>public RoleUser AddUser(string role, string account)</code>	Adds a user to a given Role. If the role does not exist it will be created
---	--

## Role class

The Role object represents a role in a process instance.

**Namesp** IYOPRO.Api

**ace:**

**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)

**y:**

### Syntax

```
class Role(string rolename)
```

### Members

 <code>RoleUser AddUser(string account)</code>	Add a user to the role by specifying the account
 <code>void RemoveUser(string account)</code>	Remove a user from the role by specifying the account
 <code>string Name</code>	Gets the name of the role
 <code>RoleMembers Members</code>	Gets the members of the role

## RoleMember class

The RoleMember class is abstract and will be derived by the RoleUser and RoleEmail class.

**Namesp** IYOPRO.Api

**ace:**

**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)

**y:**

### Syntax

```
abstract class RoleMember
```

## RoleUser class

The RoleUser object represents a user as member of a role.

**Namesp** IYOPRO.Api

**ace:**

**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)

**y:**

### Syntax

```
class RoleUser : RoleMember
```

### Members

 <code>string Account</code>	The Account of the role user
---	------------------------------

## RoleEmail class

The RoleEmail object represents an e-mail as member of a role.

**Namesp** IYOPRO.Api

**ace:**

**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)

**y:**

### Syntax

```
class RoleEmail : RoleMember
```

### Members

 <code>string</code> Email	The e-mail of the role user
---	-----------------------------

## TailoredActivities class

The TailoredActivities object specifies a collection of tailored activities. An instance of the TailoredActivities collection is provided by the [ProcessInstance](#) object.

**Namesp** IYOPRO.Api

**ace:**

**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)

**y:**

### Syntax

```
class TailoredActivities : ObservableCollection<TailoredActivity>
```

### Members

 <code>TailoredActivity</code> Add( <code>int</code> diagramid, <code>int</code> ctrlid)	Add an activity represented by diagram / controlid to the list of tailored activities
---	---

## TailoredActivity class

The TailoredActivity object represents a tailored activity.

**Namesp** IYOPRO.Api

**ace:**

**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)

**y:**

### Syntax

```
class TailoredActivity
```

### Members

 <code>int</code> DiagramId	The id of the diagram the control resides in
 <code>int</code> ControlId	The id of the control

## ProjectTasks class

The ProjectTasks object represents a collection of project tasks.

**Namesp** IYOPRO.Api

**ace:**

**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)

**y:**

### Syntax

```
class ProjectTasks : ObservableCollection<ProjectTask>
```

### Members

 <b>void</b> Add( <b>ProjectTasks</b> tasks)	Adds a task to the collection
 <b>ProjectTasks</b> Flattenize()	Creates a flat list of all tasks and subtasks

## ProjectTask class

The ProjectTask object represents an activity within the project definition of the workflow.

**Namesp** IYOPRO.Api

**ace:**

**Assembl** IYOPRO.Api (in IYOPRO.Api.dll)

**y:**

### Syntax

```
class ProjectTask
```

### Members

 <b>bool</b> IsSubTask()	Returns true if the task is a subtask, false otherwise
 <b>string</b> Name	The name of the task
 <b>string</b> Description	The plain description of the task
 <b>int</b> Diagram	The diagram id of the task
 <b>int</b> CtrlId	The control id of the task
 <b>int</b> TaskId	The Id of the task in the PM System
 <b>bool</b> Tailorable	Indicates, if the task can be tailored or not
 <b>int</b> ChildDiagram	The Id of an associated child diagram or 0 if nothing is assigned
 <b>string</b> Predecessors	The PM predecessors of the task separated by semicolon, i.e. 1;2
 <b>string</b> ResourceNames	The PM ressource names of the task separated by semicolon i.e. Manager;Worker
 <b>DateTime?</b> ActualStart	The actual start timestamp of the activity
 <b>DateTime?</b> ActualFinish	The actual finish timestamp of the activity
 <b>int</b> PercentComplete	The finish indicator from 0..100
 <b>double</b> Duration	The duration of the activity in an undetermined unit



**ProjectTasks** SubTasks

A collection of subtasks. If not null, this activity is a collective task

## FAQ and troubleshooting

---

FAQ and troubleshooting

### Install Digital Pen

#### Installing the DP-201 Digital Pen Drivers

The Digital Pen drivers are required to connect your digital pen docking station to your computer. When installed the computer will send the written data stored in the pen to the appropriate IYOPRO workflow instance.

Download the driver [PenDownloadDriver-DP201-v2.7.5-32\\_64Bit.zip](#) and extract the content of the archive into a new folder. Then run Setup.exe and follow the instructions. There are no special settings needed so you may step through the dialogs with next until the installation process has finished.

Installation of SendAll

If you wish to send all data on the pen in one single transaction you may doubleclick on PR\_SRS\_SendALL\_1.reg. If you omit this, you always have to wipe over the USB-Symbol on your Pen card

### Installing the Batch Printer

The Batch Printer is required to print forms on your printer.

First download and install [Kayentis\\_Printing\\_Solution\\_1.3.4 \(SRS DYNAMICFORMS\).exe](#). This will also install Ghostscript on your computer which is required to print dynamic forms.

Then download and install [BatchPrinterService-V1.8.4.zip](#).

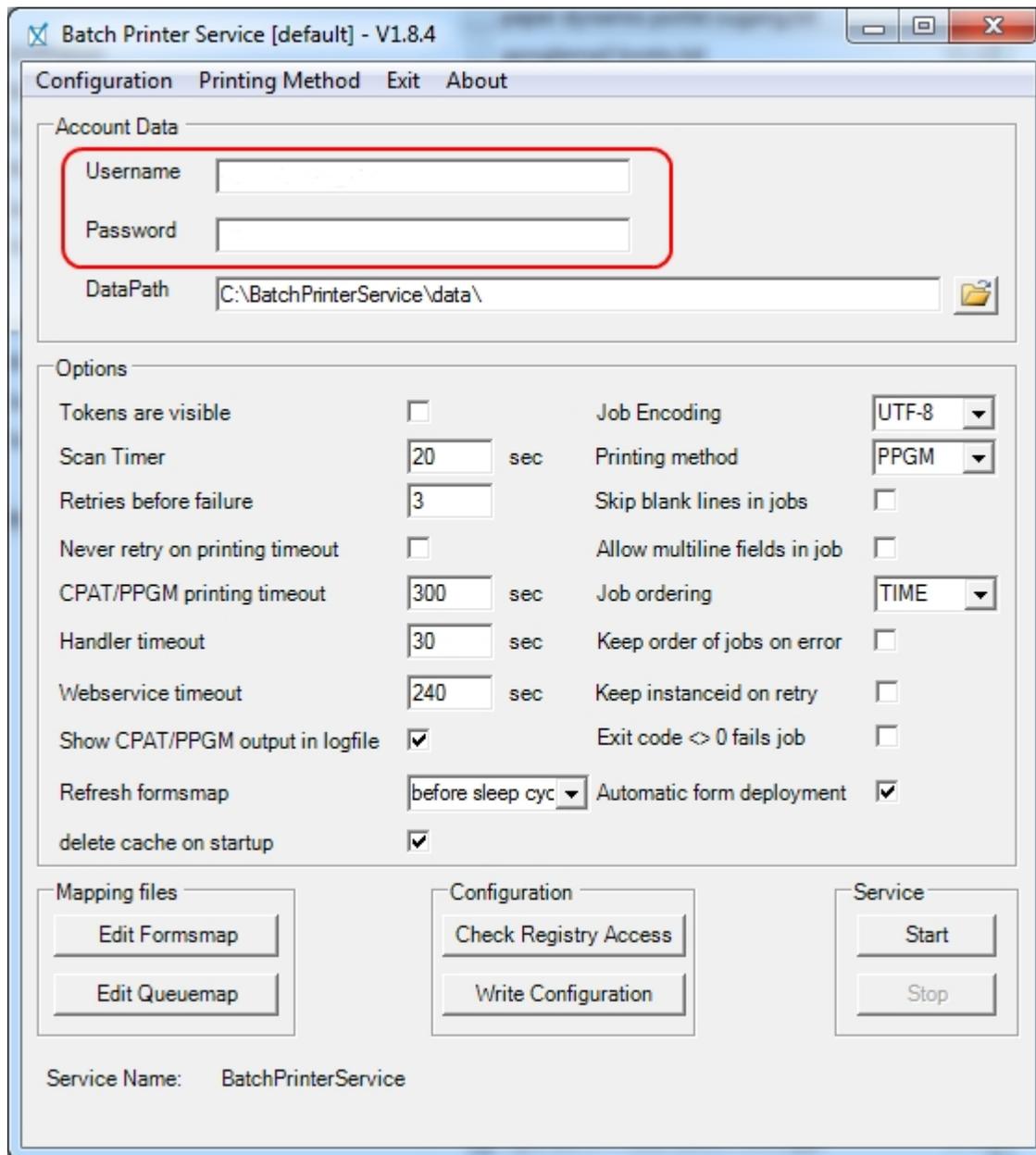


If you are asked where to install the Batch Printer Service we recommend to choose a directory without spaces in the filename to avoid further problems. There are several extensions available which require to reconfigure Batch Printer Service if there are spaces in the installation path.

We always install to c:\BatchPrinterService

After installation you need to configure the service. Use Start->Programs->SRS->Batch Printer Service->Batch Printer Service Config to start the configuration editor.

You need to input your Account data which has been deployed to you with your pen. Also adjust all settings as shown in the screenshot below.



In a Demo Environment choose Configuration->Service Endpoint and change the Endpoint to custom. Then enter  
<https://demo.paperdynamix.de>

To configure the PPGM settings choose Configuration->PPGM Settings... Adjust the Path to PPGM.exe. This is usually

C:\Program Files (x86)\Kayentis\PPGM\Command Line\ppgm\_cl.exe

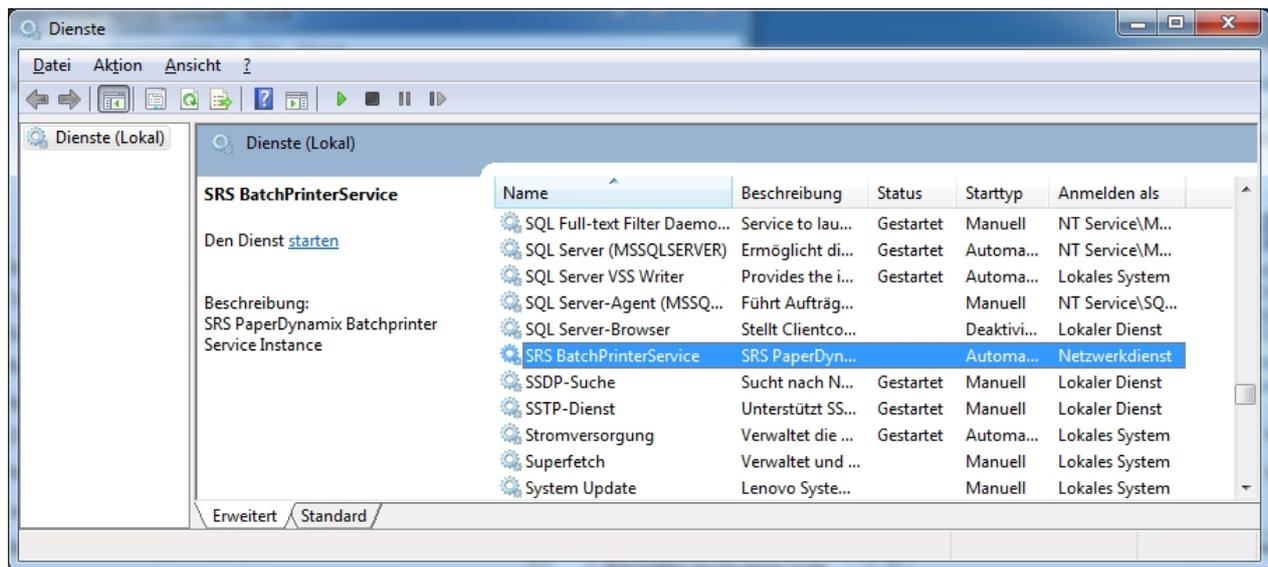
and click ok to apply these settings

Now you should be able to create your forms map without errors. To create the formsmap choose Configuration->Create Forms Mapping. To verify if the forms mapping has been created successfully click on "Edit Formsmap". You should see a list of deployed forms.

After that you need to create the Queue mapping to allow IYOPRO to print on your printers. Therefore click on Configuration->Create Queue Mapping. Again you may verify the results by clicking on "Edit Queuemap".

Next you need to configure the account, under which the service will operate. This must be an

administrative account of your computer. Click on Start and type services.msc and press enter. In the appearing service dialog locate "SRS BatchPrinterService"



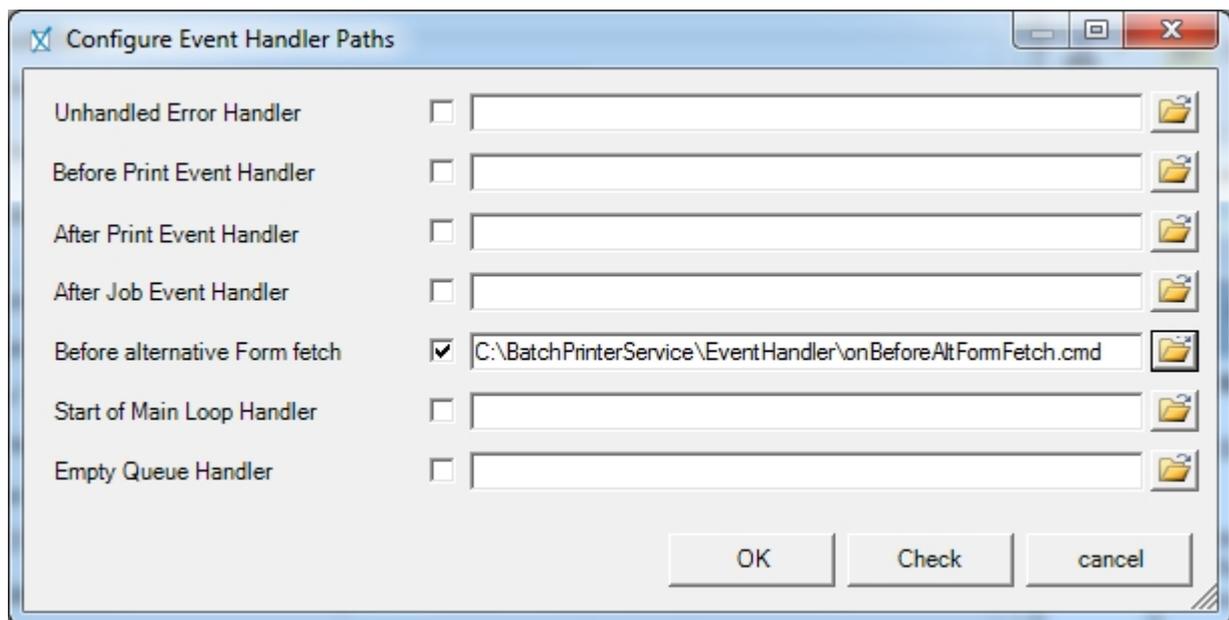
Right click on the service and choose Properties. On the LogOn Tab type in the credentials of an administrative account on your computer. Confirm the changes.

You may now be able to start the Service. To start the service go to the Batch Printer Configuration Dialog and hit the Start button.

## QR-Code support

To install the QR-Code support download the [qrcode-tools.zip](#) and extract the content into the tools subdirectory of your Batch Printer Service. If you have installed the service at c:\BatchPrinterService the content of the archive must be installed at c:\BatchPrinterService\tools. You may need to create this directory if it does'nt exist.

Download [qrcode-activate.zip](#) and extract the content to a temporary directory. Copy the file onBeforeAltFormFetch.cmd to the EventHandler subdirectory of your Batch printer Service. Copy the file modform.pdf to the data subdirectory of your Batch printer Service. Open the Batch Printer Service Configuration and choose Configuration->EventHandler Settings... Assign the "Before alternative Form fetch" event handler as shown below



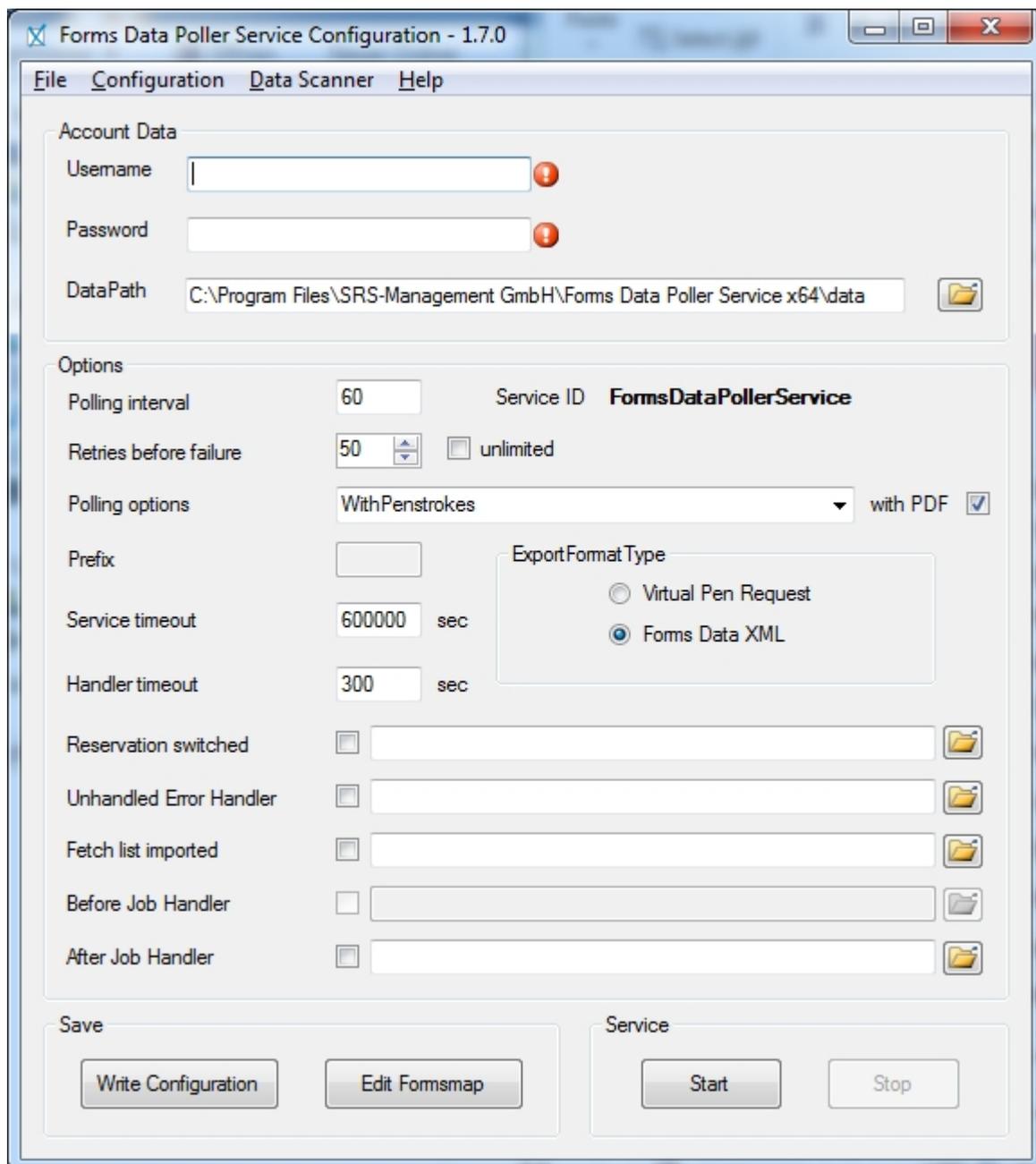
## Installing the Forms Data Poller

This is only required in special test scenarios and not recommended for productive use

Download and extract [FormsDataPollerService-1\\_7\\_0-Release-x64.zip](#) . Run Setup.exe and follow the instructions

After installation you need to configure the service. Use Start->Programs->SRS->Forms Data Poller->FormsDataPoller Config to start the configuration editor.

You need to input your Account data which has been deployed to you with your pen. Also adjust all settings as shown in the screenshot below.

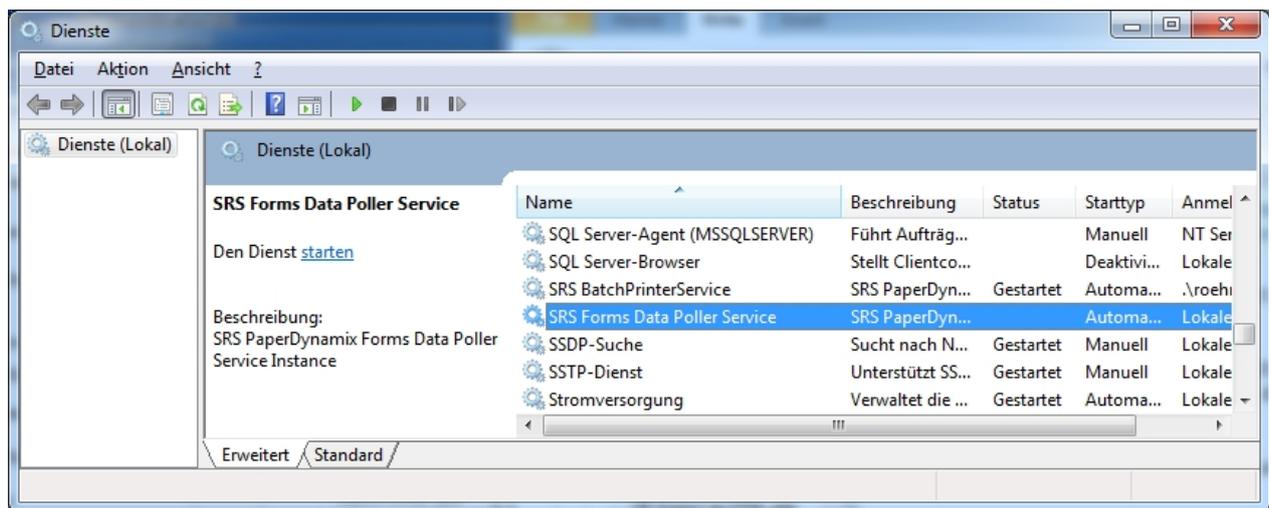


In a Demo Environment choose Configuration->Service Endpoint and change the Endpoint to custom. Then enter <https://demo.paperdynamix.de>

Now you should be able to create your forms map without errors. To create the formsmap choose Configuration->Create Forms Mapping. To verify if the forms mapping has been created successfully click on "Edit Formsmap". You should see a list of deployed forms.

To notify IYOPRO on the event of a new job add an "After Job Handler". This command script will be provided to you on request.

Next you need to configure the account, under which the service will operate. This must be an administrative account of your computer. Click on Start and type services.msc and press enter. In the appearing service dialog locate "SRS Forms Data Poller Service"



Right click on the service and choose Properties. On the LogOn Tab type in the credentials of an administrative account on your computer. Confirm the changes.

You may now be able to start the Service. To start the service go to the Batch Printer Configuration Dialog and hit the Start button.

## Legacy topics

This section contains information that is only still included for legacy reasons.

### Firefox security warning

Users of the Firefox browser receive the following Warning:



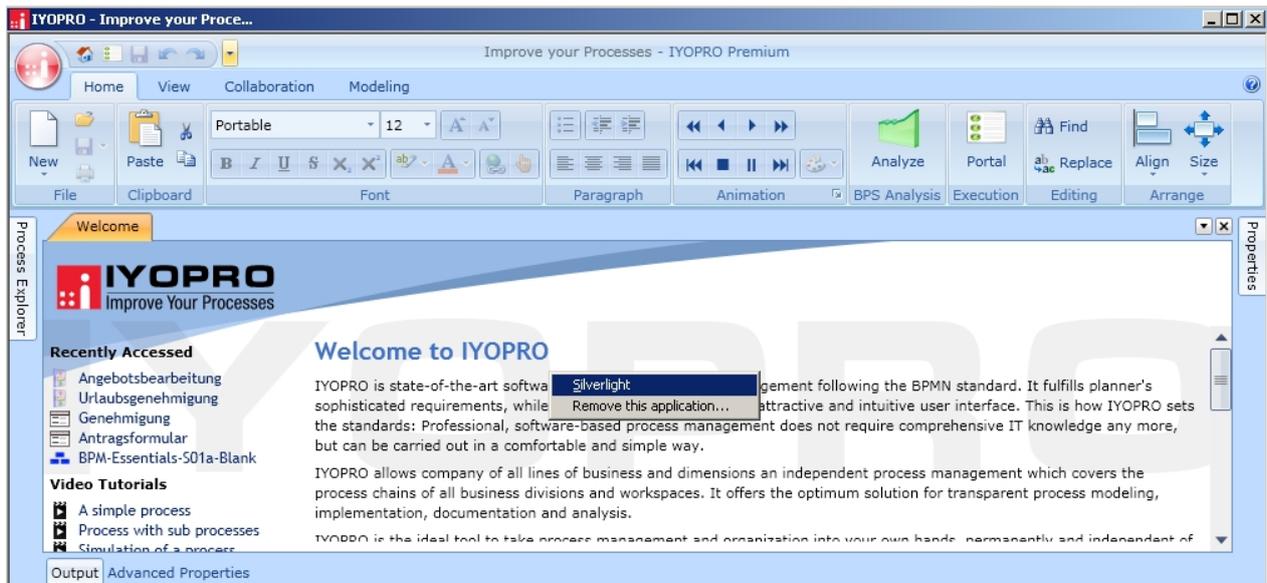
This is because the IYOPRO page needs a link to the Microsoft download page for users, which have not installed Silverlight.

The image which is generated by Microsoft is send by a http server from Microsoft where IYOPRO runs over a secure https connection.

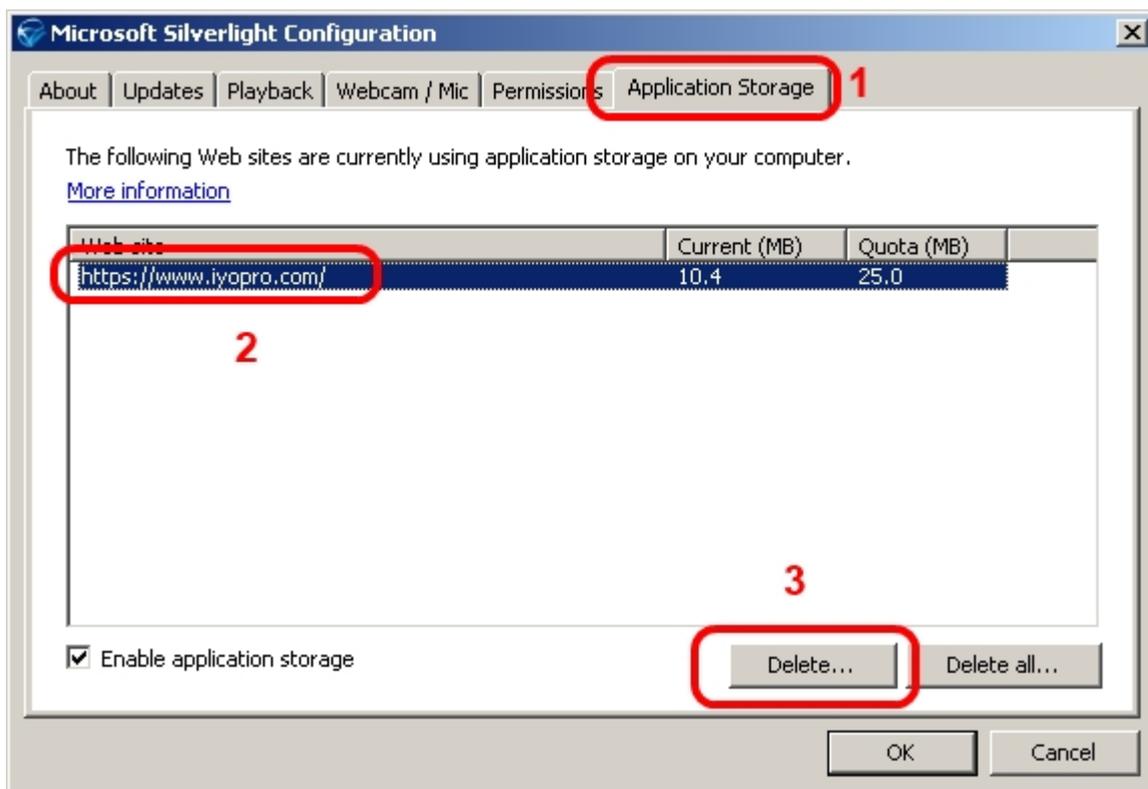
Because Firefox checks this even if the image is not used the above warning appears and may be safely ignored.

### Delete Silverlight Isolated Storage

To delete the Silverlight Isolated Storage open IYOPRO and press the right mouse button in an area where the Silverlight context menu appears. Choose Silverlight.



In the upcoming Silverlight configuration dialog navigate to the Application Storage Tab. Locate the IYOPRO entry which may be identified at the URL location <https://www.iyopro.com/>. Select the row and press the delete button. This will free the space which has been consumed by your local installation and by your user settings.



Close the dialog.

## Uninstall OOB Installation

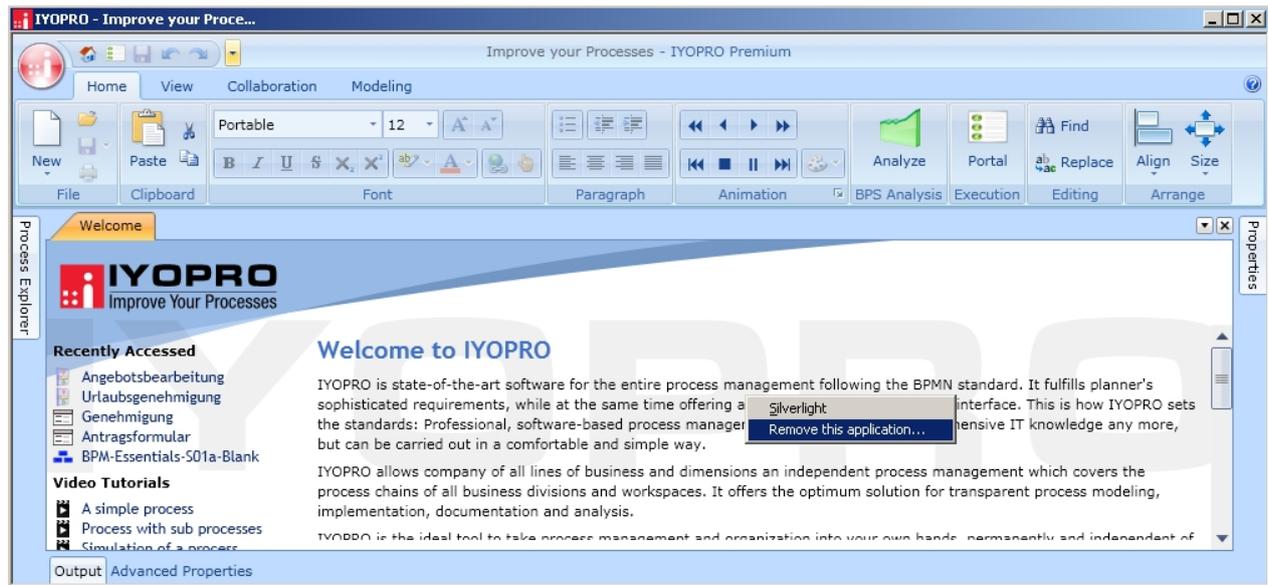


### Important notice:

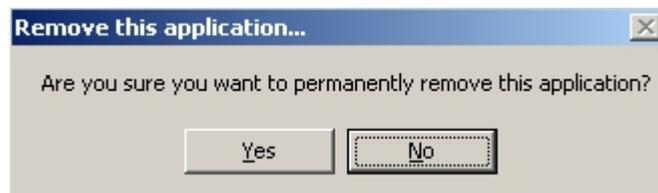
To uninstall IYOPRO completely from your computer you need to [delete the Silverlight Isolated Storage](#) as well as the application itself.

## Remove the IYOPRO OOB Application

To remove your IYOPRO OOB application start IYOPRO and press the right mouse button in an area where the Silverlight context menu appears. Choose "Remove this application..."

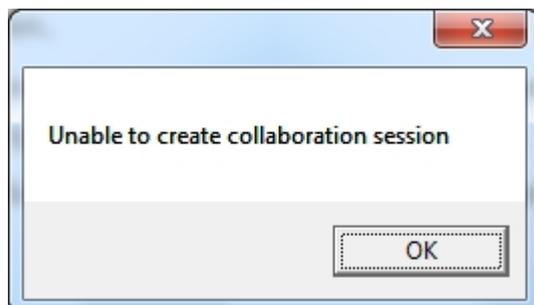


Confirm to permanently remove IYOPRO.

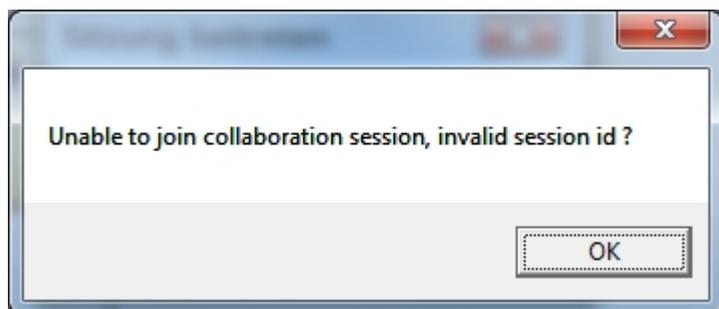


## Unable to create / join collaboration session

If you try to create a collaboration session IYOPRO may show the following error message:



If you try to join a collaboration session IYOPRO may show the following error message:



This errors indicate, that IYOPRO could not communicate with the collaboration server. This may be caused by a restrictive setting of your firewall.

## Configuring the Firewall

To be able to use services from Silverlight the following ports should be enabled for communication:

### **4502-4534**

In the case of the IYOPRO collaboration server you should at least ensure that the port **4502** has been opened.

In addition IYOPRO provides it's socket policy file on port **943** which must be open too.

For more information please refer [http://msdn.microsoft.com/en-us/library/cc645032\(v=vs.95\).aspx](http://msdn.microsoft.com/en-us/library/cc645032(v=vs.95).aspx)

# System management

---

## System management functions

To be able to perform any of the system management functions in an on premise installation of IYOPRO, one needs to be configured as system administrator.

### User management

The user management part of the system management is available via the users dialog of the teams. This dialog has a button **User Management** with a drop-down menu to make the individual system management functions available.

### Synchronization with a directory service

In an on premise installation of IYOPRO a synchronization with the existing directory service (usually Active Directory) with IYOPRO can be activated. This synchronization operates on 3 different levels:

1. Users
2. Team membership
3. User group membership

### User synchronization

With the user synchronization it is possible to maintain the users that should be able to work with IYOPRO as a user group within the directory.

This user group can have nested groups which will be evaluated as well.

As a result, users in the defined group are added as IYOPRO users, if they are not already present. On the other hand, IYOPRO user that are not members of the group are disabled.

In addition properties of the user in the directory are updated in IYOPRO to e.g. reflect changes of the name or the like.

Users who are part of the group, but are disabled in the directory, will be also disabled in IYOPRO.

### Team member synchronization

Specific teams in IYOPRO can be set up to have the users from the user synchronization added / removed automatically. This is helpful if one maintains one or more teams that should be accessible to all IYOPRO users.

### User group membership synchronization

The user groups in IYOPRO can be set up to be synchronized with the directory service as well. Based on a customized parent

entry, all user groups in this directory entry are available for the synchronization.

If one of the available user groups is selected to be the source of the group members in IYOPRO, the group members from the directory service are used in IYOPRO.

### Trigger synchronization

Click on **Trigger User Synchronization** in the user management drop down menu to request a run of the synchronization.

Please note that this entry is only available if the system is configured to synchronize information with the directory service.

## Reset a users security id

IYOPRO stores the security ID of a user synchronized with the directory service internally. Should the users security ID in the directory service change, IYOPRO can not match the user in the directory service with the IYOPRO user anymore.

To overcome this situation the security ID of an user can be cleared. The next synchronization with the directory service will match the user based on the account and not the ID. If a match is found the ID will be updated in IYOPRO.

## Reset users security id

Click on **Reset selected users security id** in the user management drop down menu to reset the security id of the selected users.

## Delete user

As part of the system management users in an on premise installation of IYOPRO can be delete. This will also delete the personal data of the users and **can not be undone**.

## Delete users

Click on **Delete selected users and their personal data from IYOPRO** in the user management drop down menu to delete the selected users from IYOPRO.

## Trigger license information update

In an on premise installation of IYOPRO the information about available licenses are periodically updated. In some cases it might not be desirable to wait for the next scheduled update, but to trigger a refresh right away.

Select the team used for the license management in the repository.

Click on **Software Keys** and than use the button **License management / Refresh license data**. This will initiate an update of the available license information.

## LDAP synchronization troubleshooting

In this section the possible issues when executing an automated synchronization with the directory service are described.

Also actions to mitigate the issues are given.

## Connection error during user synchronization

Connection to LDAP not possible (Error code 101).

To perform a synchronization of users between the directory service and IYOPRO a connection to the directory service is established by IYOPRO.

If this connection can not be established no synchronization of user can not be performed.

Typical reasons for connection errors are:

- Credential configured to access the directory service are incorrect. The used credentials are stored in the IYOPRO configuration
  - The domain name is configured with the configuration key [LDAP\\_DOMAINNAME](#)
  - The user to access the directory service is configured with the configuration key [LDAP\\_USERNAME](#).  
If no user name is configured, the user of the IYOPRO service is used.
  - The password to access the directory service is configured with the configuration key [LDAP\\_PASSWORD](#)
- Directory service is not reachable via the network
  - Please check the availability of the directory service

## Invalid directory entry

While processing the entries of the group containing the IYOPRO users an entry was found, that is not a user entry (Error Code 102).

Most likely the user group in the directory service contains some other resource records. Please verify that the group only contains

- user records
- group records, if recursing through the groups is activated by configuration option [LDAP\\_GROUP\\_RECURSE](#)

## Account already in use

An new user should be added to IYOPRO, but the account name is already used by another user (Error Code 103).

Most likely a user account was removed from the directory service and subsequently disabled in IYOPRO. Later a new user account was created in the directory service using the same account name. However, as this is a new account it was assigned an new security id by the directory service, the security id of the old (disabled) account in IYOPRO and the new account do not match.

To resolve this issue there are two options:

- Delete the existing IYOPRO user
  - Use the system management function **delete user** to remove the user and all data belonging to the user from IYOPRO
- Clear the existing IYOPRO user's security id
  - Use the system management function **reset user security id** to

Option 1 is a right choice, if the old user and the new user are different persons and the old user's data (e.g. Diagrams in My Files) can be deleted. After deletion of the IYOPRO user the synchronization will create an new IYOPRO account for the new user.

Option 2 is the right choice, if the old and the new user are the same person. With the reset of the security id the next synchronization run will detect, that the user has no security id assigned and will assign the new one. Next the user can log on and access his data.

### Account name change to account in use

Changing a users account name is not possible, as there is already another IYOPRO user with the target account name (Error code 104)

During the synchronization run a change in the account name of a user was detected. This user was identified by his security id. However the new account name is already in use in IYOPRO.

If this happens please verify that the accounts provided for the user synchronization by the directory service are unique and match the existing IYOPRO accounts.

### No enabled user

After processing all users, there are no enabled IYOPRO user left (Error code 107)

Once IYOPRO has processed all users from the directory service, there are no enabled user left in IYOPRO.

Probably there is a customizing issue with the user group of the directory service that is the source of the users or that group does not contain any users.

To resolve this issue, please check

- that the group in the directory service contains user, which are not disabled
- that the group name matches the configuration entry [LDAP\\_GROUPNAME](#)
- that the credentials user to connect to the directory service allow to read the group members (configuration entries [LDAP\\_USERNAME](#) and [LDAP\\_PASSWORD](#))

### Group not found

While trying to synchronize users the configured group was not found (Error code 109).

When trying to retrieve the users to be synchronized from the directory service, the configured group was not found by the directory service.

To mitigate this issue, please check:

- that the configured group name configured in [LDAP\\_GROUPNAME](#) matches the group name in the directory service
- that the configured group is part of the provided domain configured in [LDAP\\_DOMAINNAME](#)
- that the configured credentials (configuration entries [LDAP\\_USERNAME](#) and [LDAP\\_PASSWORD](#)) to access the directory service are valid and allow retrieval of the group members

### Connection error during group synchronization

The connection to the directory service could not be established, when trying to obtain the groups to synchronize (Error code 111).

To perform a synchronization of user group between the directory service and IYOPRO a connection to the directory service is established by IYOPRO.

If this connection can not be established no synchronization of user groups can not be performed.

Typical reasons for connection errors are:

- Credential configured to access the directory service are incorrect. The used credentials are stored in the IYOPRO configuration
  - The domain name is configured with the configuration key [LDAP\\_DOMAINNAME](#)
  - The user to access the directory service is configured with the configuration key [LDAP\\_USERNAME](#).  
If no user name is configured, the user of the IYOPRO service is used.
  - The password to access the directory service is configured with the configuration key [LDAP\\_PASSWORD](#)
- Directory service is not reachable via the network
  - Please check the availability of the directory service

### Group member load error

Then loading the list of groups to synchronized an exception occurred (Error code 112).

While trying to load the members of the organizational unit configured for the group synchronization, an error occurred.

The configuration [LDAP\\_GROUP\\_SYNC\\_OU](#) specifies which entries should be synchronized.

To resolve this issue, please check:

- That the provided organizational unit contains only user groups in the directory service
- That the user groups contained in the organizational unit are accessible with the configured credentials

### Loading groups for synchronization failed

Failed to load IYOPRO user groups to be synchronized (Error code 113).

When trying to obtain the list of IYOPRO user groups that should be synchronized with the directory service, an error occurred.

### Connection error during group member synchronization

Could not establish the connection to the directory service to synchronize group members (Error code 114).

To perform a synchronization of group members between the directory service and IYOPRO a connection to the directory service is established by IYOPRO.

If this connection can not be established no synchronization of user can not be performed.

Typical reasons for connection errors are:

- Credential configured to access the directory service are incorrect. The used credentials are stored in the IYOPRO configuration
  - The domain name is configured with the configuration key [LDAP\\_DOMAINNAME](#)
  - The user to access the directory service is configured with the configuration key

- [LDAP\\_USERNAME](#).  
If no user name is configured, the user of the IYOPRO service is used.
  - The password to access the directory service is configured with the configuration key [LDAP\\_PASSWORD](#)
- Directory service is not reachable via the network
  - Please check the availability of the directory service

### Group assignment not possible (no IYOPRO user)

Assignment of a user to a group not possible, as the user is not an IYOPRO user (Error code 115).

The reason for this issue is that a user is member of the group in the directory service. However this user is not a known IYOPRO user.

To resolve this issue make sure that all users specified as members in groups to be synchronized are also IYOPRO users. Check that the users are also part of the configured source for the user synchronization (configuration key [LDAP\\_GROUPNAME](#)).

### Group assignment not possible (user not in team)

Assignment of the user to a user group not possible, as the user is not in the team (Error code 116).

The user is not a member of the team which contains the group to be synchronized.

To resolve this issue make sure that all users specified as members in groups to be synchronized are team members in the team containing the IYOPRO group.

If the team is not part of the auto joined teams (configuration key [LDAP\\_AUTO\\_JOINED\\_TEAMS](#)), the user needs to be added to the team manually.

### Team user list empty

Failed to get the list of team users during auto join (Error code 118).

While trying to synchronize group members the user list of the team containing the user group could not be loaded.

To resolve this issue verify the team member and ensure that the list of team members is not empty.

### Configured team id is invalid

The team id configured for auto join is not a valid id (no int) (Error Code 120)

The configuration for the automatic joining of teams (auto join) contains an entry that can not be parsed as team id, as it's not of the data type integer.

To resolve the issue ensure that the configuration key [LDAP\\_AUTO\\_JOINED\\_TEAMS](#) contains a comma separated list of valid team id's.

### Configured team id not found

The team with the configured team id for auto join was not found (Error Code 121)

The configuration for the automatic joining of teams (auto join) contains an entry that is not a valid team id.

To resolve this issue verify the configuration key [LDAP\\_AUTO\\_JOINED\\_TEAMS](#) and ensure that all provided id's are valid team id's.

The team id can be found in the repository. Select the teams node and expand. When a team is selected, the team id is shown right below the team name in the details display.

### **Conversion of user information failed**

Conversion of the directory service object failed, when trying to determine the user status (enabled/disabled) (ErrorCode 122)

While trying to determine the status of an user via the directory service, the provided data could not be converted. IYOPRO will assume that the user is enabled in the directory service. If the user is currently disabled in IYOPRO, it will be enabled.

### **User enabled property is empty**

The directory service data has an empty UserAccountControl property (Error code 123).

While trying to determine the status of an user via the directory service, the provided data did contain the needed property UserAccountControl, but it does not contain any data. IYOPRO will assume that the user is enabled in the directory service. If the user is currently disabled in IYOPRO, it will be enabled.

### **User enabled property has wrong type**

The directory service data has an empty UserAccountControl property has the wrong data type (Error code 124).

While trying to determine the status of an user via the directory service, the provided data did contain the needed property UserAccountControl, but it has the wrong data type. IYOPRO will assume that the user is enabled in the directory service. If the user is currently disabled in IYOPRO, it will be enabled.

### **User enabled property is missing**

The directory service data has an empty UserAccountControl property is not present (Error code 125).

While trying to determine the status of an user via the directory service, the provided data did not contain the needed property UserAccountControl. IYOPRO will assume that the user is enabled in the directory service. If the user is currently disabled in IYOPRO, it will be enabled.

### **Synchronization aborted**

While performing the synchronization an error was detected and the synchronization process was aborted (Error Code 126).

To understand the root cause of the issue, please verify the ADSync thread entries in the backend log. If the configuration option LDAP\_ONERROR\_ABORT is set to false, detected errors will be skipped.

## On Premises Configuration

---

An on premises installation allows you to customize certain settings to suit your business needs.

### E-mails

An on premises installation allows you to customize e-mails that are not defined in a process model.

The following chapters describe which e-mails you can customize and how you can customize them.

### Delegation

When you [delegate](#) a task, the recipient of the task receives an e-mail. If no adjustments have been made, the default text will be sent.

You can either customize the default text or create a template.

### Standard Mail

You can customize the standard subject and body of the e-mail in the Backend.Config.xml of IYOPRO. To do this, simply exchange the values of the keys DELEGATE\_MAIL\_SUBJECT\_PREFIX and DELEGATE\_MAIL\_BODY\_ADDITION with the texts you require.

You may use the following tags within your text:

- USERNAME
- COMMENT

The tag {USERNAME} is replaced with the original user's name.

The tag {COMMENT} is replaced with the comment provided by the person who delegated the task. Use \n as new line indicator.

### Template Mail

With the template you are able to not only customize the text of your e-mail but also design your e-mail.

#### Create the template for the mail body

For IYOPRO to use the template, there has to be a file named "DelegationInfo" at the following location:

- ServerBackend\BackendStorage\Templates\System\EMail\Delegation

The file can be either a .html or a .txt file.

In case of a delegation the template will be processed and sent with the following tags:

- USERNAME = Original User Name
- SUBJECT = Org. Mail Subject
- MSGBODY = Org. Mail Body
- Context TaskUrl = Reference to Tasks
- COMMENT = Delegation comment

Except for Context TaskUrl, use \$ characters instead of braces around the tags: \$USERNAME\$

### Sample

This example shows a possible design of the e-mail body in html.

```

<table style="font-family:Tahoma, 'Times New Roman', serif;" width="100%">
  <tr bgcolor="#ffc000">
    <td style="padding-left:10px;"><b>Task was delegated by:</b></td>
    <td>${USERNAME}</td>
  </tr>
  <tr bgcolor="#ffc000">
    <td style="padding-left:10px;">Task:</td>
    <td><a href="{Context TaskUrl}" target="_blank">{Binding Token.Control.Name}</
a></td>
  </tr>
  <tr bgcolor="#ffc000">
    <td style="padding-left: 10px;">Process:</td>
    <td>{Binding ProcessInstance.Name} [{Binding ProcessInstance.Id}]</td>
  </tr>
  <tr bgcolor="#ffc000">
    <td style="padding-left: 10px;">Details:</td>
    <td>{Binding ProcessInstance.Details}</td>
  </tr>
  <tr bgcolor="#ffc000">
    <td style="padding-left: 10px;">Comment: </td>
    <td>${COMMENT}</td>
  </tr>
</table>
<br />

```

## Create the template for the mail subject

To use a custom mail subject, there has to be a .txt file named "DelegationInfo-Title" at the same location as the template for the mail body.

You may use the following tags:

- SUBJECT
- USERNAME
- COMMENT

Use \$ characters instead of braces around the tags.

### Sample

EN = IYOPRO task \$SUBJECT\$ was delegated by \$USERNAME\$

### Substitution

If you defined [substitutes](#) in IYOPRO, you can specify whether substitutes should receive an e-mail when a task has been forwarded to them.

To do this, set the value of SUBSTITUTE\_MAIL\_SEND to either True or False. You can find it in the Backend.Config.xml of IYOPRO.

If an e-mail is to be sent, the substitute receives the default text if no adjustments have been made. You can either customize the default text or create a template.

## Standard Mail

You can customize the standard subject and body of the e-mail in the Backend.Config.xml of IYOPRO. To do this, simply exchange the values of the keys SUBSTITUTE\_MAIL\_SUBJECT\_PREFIX and SUBSTITUTE\_MAIL\_BODY\_ADDITION with the texts you require.

You may use the following tags within your text:

- USERNAME

The tag {USERNAME} is replaced with the original user's name.

Use \n as new line indicator.

## Template Mail

With the template you are able to not only customize the text of your e-mail but also design your e-mail.

### Create the template for the mail body

For IYOPRO to use the template, there has to be a file named "SubstitutionInfo" at the following location:

- ServerBackend\BackendStorage\Templates\System\EMail\Substitution

The file can be either a .html or a .txt file.

In case of a substitution the template will be processed and sent with the following tags:

- USERNAME = Original User Name
- SUBJECT = Org. Mail Subject
- MSGBODY = Org. Mail Body
- Context TaskUrl = Reference to Tasks

Except for Context TaskUrl, use \$ characters instead of braces around the tags: \$USERNAME\$

### Sample

This example shows a possible design of the e-mail body in html.

```
<table style="font-family:Tahoma, 'Times New Roman', serif;" width="100%">
  <tr bgcolor="#ffc000">
    <td style="padding-left:10px;"><b>Task in substitution of:</b></td>
    <td>$USERNAME$</td>
  </tr>
  <tr bgcolor="#ffc000">
    <td style="padding-left:10px;">Task:</td>
    <td><a href="{Context TaskUrl}" target="_blank">{Binding Token.Control.Name}</
a></td>
  </tr>
  <tr bgcolor="#ffc000">
    <td style="padding-left: 10px;">Process:</td>
    <td>{Binding ProcessInstance.Name} [{Binding ProcessInstance.Id}]</td>
  </tr>
  <tr bgcolor="#ffc000">
    <td style="padding-left: 10px;">Details:</td>
    <td>{Binding ProcessInstance.Details}</td>
  </tr>
</table>
<br />
```

### Create the template for the mail subject

To use a custom mail subject, there has to be a .txt file named "SubstitutionInfo-Title" at the same location as the template for the mail body.

You may use the following tags:

- SUBJECT
- USERNAME

Use \$ characters instead of braces around the tags.

### Sample

EN = Substitution for IYOPRO task \$SUBJECT\$

## Workflow data access

The process instance state pages and user tasks provide the user with information about a workflow (instance).

In principle, a user who is the administrator of a team or of a workflow can view and manage all the data of that team or workflow.

But how much data of the process instance should a user have access to if he is not an administrator?

In the Backend.config.xml of your IYOPRO on premises installation, you may customize according to your decision.

To do this, simply set the value of the key `PI_ACCESS` and the values of the key `ADMIN_ONLY_PI_TABS`.

The **PI\_ACCESS** determines who is allowed to access workflow forms and process instance state pages. You have the following options:

- participants (default)
- creatorandparticipants (includes process instance creator)
- team
- world

The **ADMIN\_ONLY\_PI\_TABS** determines which tabs can only be accessed as an administrator.

You may choose the following:

- tasks
- roles
- diagram
- log
- attributes
- timeline
- statistic

According to your selection, for example, only a workflow participant has access to his edited tasks as well as the process instance pages. However, the workflow participant won't see the data of the selected tabs in the process instance state pages.

## Workflow version name

When you [deploy a productive version](#) of a workflow, you can either create a new version or select an already existing one. When creating a version, you have to choose a name for that version.

If you want to have a standard naming for the productive versions, you may specify a value for the key `WORKFLOW_VERSION_DEFAULT_NAME` in the Backend.config.xml.

## System-specific attributes

In the Backend.config.xml of IYOPRO, system-specific attributes can be stored in the form of key value pairs.

To do this, you specify values in form of key value pairs for the key `WORKFLOW_USER_CONFIG`.

The values can then be retrieved through the configuration value.

To retrieve these custom attributes you use the [Application](#) object.

## Default Customer Settings

Use the [customer settings](#) to make company-wide defaults.

To create and activate the default customer settings, log into IYOPRO with your system user.

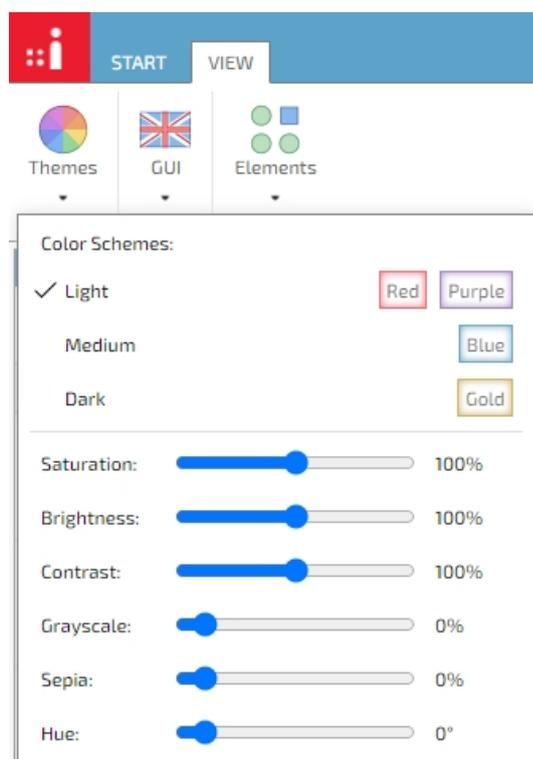
Create your customer settings and save them with the name "Default" in the following directory: MyFiles/ Templates/ CustomerSettings

If the directory doesn't already exist, create the necessary folders according to specified order.

## Color Schemes

You can change the colors of the web client's surface according to the options available. However, you can also customize these options.

The following image shows you a colorful example:



As you can see in the example, you have both the option to add other themes than the standard Light and Dark and to add color options.

In the web client 2.0 folder you can find a themes folder that contains the files that define the available themes. **Attention:** the files with the default themes will be overwritten during an update. So if customer-specific settings are desired, make a copy of the file and make the corresponding adjustments there.

In the config.json of the IYOPRO web client, you can then define the themes that should be available in your company.

### Addition

The possible customizations include not only the different colors but also the font and the option to add a logo (displayed on the bottom left).

### Warning

The content of the files must be syntactically correct, otherwise the config file cannot load.

## Use internal portal

You may specify whether the portal button of your IYOPRO on premises installation should open the internal portal or the web portal.

By default, the web portal opens. If you want to set the internal portal as the default instead, simply set the value of the key `USE_APP_INTERNAL_PORTAL_DEFAULT` to `True` in the `Backend.config.xml` of your IYOPRO.

User have the option to change which portal opens for them in their [preferences](#).

## System Performance Logging

There are extended system performance logging options. You can adjust these in the `Backend.Config`.

The entries are as follows:

```
<add key="SYSTEM_LOG_PYENGINE_PROVISION" value="NONE"/>
<add key="SYSTEM_LOG_PYEXPRESSION_EXECUTION" value="NONE"/>
<add key="SYSTEM_LOG_LOADWORKFLOWFORMDATA" value="NONE"/>
<add key="SYSTEM_LOG_GENERALCALL" value="DURATION"/>
```

You may choose from this list of available values:

NONE	None (default)
DURATION	Always with duration
BLOCK	Always as block
BLOCKDURATION	Always as block with duration
EXCEEDED:nn	Only if threshold exceeded

## Check the service

With the provided functionality to monitor the system, you have the opportunity to check whether your service is running normally. This information can be evaluated automatically.

All you have to do is call up the following URL:

`<IYOPRO URL>/WebService-IYOPRO.asmx/ServiceCheck`

If you would like further information to be evaluated in addition to the standard information, you can specify the corresponding values of the `SERVICE_CHECK_EXTRAINFO` key in the `Backend.Config`.