

PROPOS XML import specification - PROPOS inbound interface

Version history

Version	Date	Author	Remarks
1.33	2019/01/05	ing. R. Peters	New field added: WaitTimeAfterCellMin.
1.32	2018/10/29	ing. R. Peters	New fields added: ProdStartQty and MinLeadtimeMin.
1.31	2018/10/23	ing. R. Peters	Reference to varchar(250).
1.30	2018/09/20	ing. R. Peters	StockShortageText to varchar(max)
1.29	2018/07/05	ing. R. Peters	POFactoryDescr to varchar(100).
1.28	2018/01/23	ing. R. Peters	AuthorWarning added to PO.
1.27	2017/11/30	ing. R. Peters	Better explanation of ParallelResourceID.
1.26	2016/09/29	ing. R. Peters	Title and introduction change. Field sizes added.
1.25	2016/07/29	ing. R. Peters	General updates, clarifications.
1.24	2016/04/04	ing. R. Peters	Attachments added to structure.
1.23	2016/03/07	ing. R. Peters	WindowsUsername added to structure.
1.22	2016/02/17	ing. R. Peters	XML structure change.
1.21	2016/02/12	ing. R. Peters	Small adjustment XML checker.
1.20	2016/02/11	ing. R. Peters	PlanningNr added. Small datetime format changes. XML checker beta added.
1.19	2015/11/07	ing. R. Peters	Extra cooling time remarks.
1.18	2015/11/03	ing. R. Peters	ParallelResourceID = Int.
1.17	2015/10/24	ing. R. Peters	Cooling time implemented.
1.16	2015/10/19	ing. R. Peters	ParallelResourceID implemented.
1.15	2013/06/25	ing. R. Peters	Small adjustment XML example page 3.
1.14	2013/03/12	ing. R. Peters	POEndDT time added.
1.13	2013/03/05	ing. R. Peters	UserSort1, UserSort2 added for POLines.
1.12	2012/11/02	ing. R. Peters	ERPLineID added for PO steps.
1.11	2012/10/16	ing. R. Peters	Employee Registration module specs added.
1.10	2012/10/10	ing. R. Peters	Update functionality synchronisation.
1.09	2012/09/12	ing. R. Peters	Synchronisation fields changed.
1.08	2012/06/02	ing. R. Peters	Stock shortage functionality added.
1.07	2012/05/20	ing. R. Peters	General information adjusted.
1.06	2012/04/23	ing. R. Peters	General information added.
1.05	2012/04/19	ing. R. Peters	Two fields added.
1.04	2012/04/12	ing. R. Peters	Synchronization fields added.
1.03	2012/04/02	ing. R. Peters	RemoveAndRebuild field added.
1.02	March 2012	ing. R. Peters	Updated field descriptions.
1.01	March 2012	ing. R. Peters	Extension of field descriptions.
1.0	February 2012	ing. R. Peters	1.0 version

Disclaimer

The information in this document can be subject to change. Always ensure that you are using the latest version. Please contact us for information. No rights can be derived from this document.

Document description

This document describes the PROPOS inbound interface.

PROPOS imports production orders with their production order lines from an XML file. This document describes what the XML file that PROPOS imports should look like, what its layout should be, what the data types per field are etc.

For the PROPOS module **Employee Registration**, additional information is needed in XML. You will find the XML layout for this module in this document after the description for the basic PROPOS module.

General information about connecting to PROPOS

PROPOS imports production orders (PO's) and production order lines from an XML file. As long as PO's are active, they must appear in the XML file. When a PO disappears from the XML file, PROPOS will mark the PO as closed and PROPOS will archive the PO.

The location and the name of the XML file can be specified in PROPOS.

Import / export synchronisation

PROPOS imports periodically, the frequency in minutes can be set in PROPOS. Only if an 'XML ready' file is present, PROPOS will import. After import, PROPOS will delete the XML ready file. The exporting function should not export if the XML ready file still exists. When export finished, the exporting function should (re)create the XML ready file. This synchronisation system guarantees that PROPOS does not try to import an XML file that has not been completed by the exporting function and that the exporting function does not export while PROPOS is still reading the XML file.

The file name of the XML ready file can be specified in PROPOS. The XML ready file can be a 0 length text file (e.g. 'ready.txt').

We strongly advise that you send your XML files to PROPOS software for testing before you start using them in practice.

New: PROPOS XML checker Beta version

At <http://xmlcheck.propos-software.nl> you can test your xml file yourself.

For this checker, keep in mind that it is case sensitive for field names. So for example you will need to <CellName> in your xml file and not <Cellname> because the latter will generate an error.

The xml checker is meant for the first phase of xml development. If the xml checker indicates that your xml file is correct, we strongly advise to test it also on a PROPOS test database before putting it live. The xml checker will do some syntax checks and semantic checks but there is no guarantee that your xml file will be 100% correct. We can help you do the final checks. Please contact us for that.

XML layout (PROPOS basic module)

General layout:

```
<?xml version="1.0" encoding="UTF-8"?>
<dataroot>
  <T_imp_PO>
    [PO 1 data]
  </T_imp_PO>
  <T_imp_POLine>
    [PO 1 line 1 data]
  </T_imp_POLine>
  <T_imp_POLine>
    [PO 1 line n data]
  </T_imp_POLine>
  <T_imp_PO>
    [PO n data]
  </T_imp_PO>
  <T_imp_POLine>
    [PO n line 1 data]
  </T_imp_POLine>
  <T_imp_POLine>
    [PO n line n data]
  </T_imp_POLine>
</dataroot>
```

The next page shows how the XML file should actually look like.

General information about creating the XML file

Please send PO first and close PO with </PO>. Then send the POLine, then send the POLineSteps (if applicable). You can also first put all PO's in the xml file, then all POLines, then all POLineSteps.

Layout details (PROPOS basic module):

```
<?xml version="1.0" encoding="UTF-8"?>
<dataroot>
  <T_imp_PO>
    <PONr>[data]</PONr>
    <PlanningNr>[data]</PlanningNr>
    <ParentPONr>[data]</ParentPONr>
    <SalesOrderNr>[data]</SalesOrderNr>
    <StatusCode>[data]</StatusCode>
    <ItemCode>[data]</ItemCode>
    <ItemGroupCode>[data]</ItemGroupCode>
    <POEndDT>[data]</POEndDT>
    <ProdQty>[data]</ProdQty>
    <ProdUnit>[data]</ProdUnit>
    <CustomerID>[data]</CustomerID>
    <CustomerName>[data]</CustomerName>
    <Reference>[data]</Reference>
    <CreatedDT>[data]</CreatedDT>
    <Creator>[data]</Creator>
    <ERPBlocked>[data]</ERPBlocked>
    <RemoveAndRebuild>[data]</RemoveAndRebuild>
    <SyncWithPO>[data]</SyncWithPO>
    <SyncWithVO>[data]</SyncWithVO>
    <SyncWithSubs>[data]</SyncWithSubs>
    <SyncERPLineID>[data]</SyncERPLineID>
    <SyncWithCellName>[data]</SyncWithCellName>
    <SyncWithERPLineID>[data]</SyncWithERPLineID>
    <POFactoryDescr>[data]</POFactoryDescr>
    <PORemarks>[data]</PORemarks>
    <WindowsUsername>[data]</WindowsUsername>
    <AuthorWarning>[data]</ AuthorWarning>
    <AuthorWarningText>[data]</ AuthorWarningText>
    <ProdStartQty>[data]</ProdStartQty>
  </T_imp_PO>
  <T_imp_POLine>
    <PONr>[data]</PONr>
    <PlanningNr>[data]</PlanningNr>
    <Sequence>[data]</Sequence>
    <CellName>[data]</CellName>
    <Operation>[data]</Operation>
    <PlanStartDT>[data]</PlanStartDT>
    <PlanEndDT>[data]</PlanEndDT>
    <ActualStartDT>[data]</ActualStartDT>
    <ActualEndDT>[data]</ActualEndDT>
    <SetupTimeMin>[data]</SetupTimeMin>
    <TactTimeMin>[data]</TactTimeMin>
    <CoolingTimeMin>>[data]<CoolingTimeMin>
    <BlockPlanStartDT>[data]<BlockPlanStartDT>
    <ERPLineID>[data]</ERPLineID>
    <OperationRemarks>[data]</OperationRemarks>
    <StockShortage>[data]</StockShortage>
    <StockShortageText>[data]</StockShortageText>
    <CombiID>[data]</CombiID>
    <UserSort1>[data]</UserSort1>
```

```
<UserSort2>[data]</UserSort2>  
<Attachments>[data]</Attachments>  
<MinLeadtimeMin>[data]</MinLeadtimeMin>  
<WaitTimeAfterCellMin>[data]</WaitTimeAfterCellMin>  
</T_imp_POLine>  
</dataroot>
```

Production order fields

Field	Unique	Description	Data type	(Max) length	Required	Format / values / remarks
PONr	Yes	Production order (PO) number	Varchar	50 char	Yes	
PlanningNr	Yes	Main planning number	Int	32 bits	Yes	Normally 1
ParentPONr	No	Parent PO number	Varchar	50 char	No	A valid PONr
SalesOrderNr	No	Sales order number	Varchar	50 char	No	
StatusCode	No	Status of the PO	Varchar	1 char	Yes	Always use 'A'
ItemCode	No	Item / article code	Varchar	50 char	No	
ItemGroupCode	No	Item / article group code	Varchar	50 char	No	
POEndDT	No	Date +time when the PO needs to be finished	Datetime	16 char	Yes	yyyy/mm/dd hh:nn
ProdQty	No	Quantity to be produced	Int	32 bits	Yes	
ProdUnit	No	Unit of measure	Varchar	50 char	Yes	E.g. 'pcs'
CustomerID	No	Customer ID in the ERP system	Varchar	50 char	No	
CustomerName	No	Name of the customer	Varchar	50 char	No	
Reference	No	PO reference	Varchar	250 char	No	
CreatedDT	No	Creation date of the PO in ERP	Datetime	16 char	Yes	yyyy/mm/dd [hh:nn]
Creator	No	Name of the creator of the PO in ERP	Varchar	50 char	No	
ERPBlocked	No	PO is blocked in ERP	Int	32 bits	Yes	-1, 0
RemoveAndRebuild	No	PO is completely removed from PROPOS en imported again	Int	32 bits	No	-1, 0
POFactoryDescr	No	Description of the PO as showed in the factory screens.	Varchar	100 char	Yes	
PORemarks	No	Remarks for the PO	Text		No	
SyncERPLineID	No	Sync my ERP line ID	Varchar	50 char	No	
SyncWithPO	No	Sync with this PO	Varchar	50 char	No	[PONr]
SyncWithVO	No	Sync with all PO's of this VO	Varchar	50 char	No	-1, [SalesOrderNr]
SyncWithSubs	No	Sync with my subs	Int	32 bits	No	-1, 0
SyncWithCellName	No	Sync with cellname in foreign PO('s)	Varchar	50 char	No	
SyncWithERPLineID	No	Sync with ERP line ID in foreign PO('s)	Varchar	50 char	No	
WindowsUsername	No	Used for PO simulator. Usually leave blank.	Varchar	50 char	No	
AuthorWarning	No	If True (-1) an exclamation mark will be shown in the Authorization screen.	Int	32 bits	No	-1, 0
AuthorWarningText	No	The text that is shown when user clicks the exclamation mark.	Varchar	250 char	No	
ProdStartQty	No	If module GPA is used in PROPOS, this field can contain the actual start quantity. See below.	Int	32 bits	No	

Primary key: (PONr, PlanningNr)

Further explanation and remarks on production order fields

Field	Remarks
PONr	PONr, PlanningNr form the primary key.
PlanningNr	
ParentPONr	Every production order with this field filled in will be considered a suborder. If the PO has no parent PO then this value should be Null.
StatusCode	A = Active C = Closed (production order will be archived) Any other value than 'A' or 'C' will be treated as 'A'.
POEndDT	The date+time that the PO needs to be finished. Specific time for shipment etc. has to be added by ERP or XML exporting function. With sub PO's if no date is available, use parent PO end date. Example: '2016/02/01 15:00' for February 1 st 2016 at 15:00 hours.
ProdQty	E.g. the number of items to be produced in this PO
ProdUnit	Unit of measure, e.g. 'pieces' or 'm2', 'litres', etc.
CreatedDT	Date and optional time when PO was created in ERP. Examples: '2016/02/01 15:00' or '2016/03/16'.
ERPBlocked	-1 = yes, the PO is blocked in the ERP 0 = no, the PO is not blocked in the ERP The PO can automatically be blocked in PROPOS if it is blocked in the ERP. Any other value than -1 or 0 will be treated as 0.
RemoveAndRebuild	-1 = yes, the PO will be removed completely from PROPOS (also if it has been archived it will be removed from archive) and it will be imported again. 0 = no Any other value than -1 will be treated as 0.
* SyncWithPO	Synchronise the current PO with another PO. SyncWithPO must contain a valid PO number that is known to PROPOS.
* SyncWithVO	Synchronise the current PO with all other PO's having the same SalesOrderNr. This field contains -1 for the same SalesOrderNr as the current PO or an explicit SalesOrderNr.
* SyncWithSubs	Synchronise the current PO with all of it's sub PO's. (-1 = true, 0 or Null = false)
* SyncERPLineID	Synchronise the ERPLineID of the current PO with the specified PO's. If not specified, the first or last ERPLineID will be synchronised, depending on PROPOS settings. ERPLineID in specified PO is looked for at POLine level and not at POLineStep level.
* SyncWithCellName	Synchronise the current PO with the first occurrence of this cell in the specified PO's.
* SyncWithERPLineID	Synchronise the current PO with the first occurrence of this ERPLineID in the specified PO's.
ProdStartQty	If module GPA is enabled and PROPOS does not manage the counting of parts after the first (few) operation(s), this field can contain the actual number of parts produced in those first operations. The field ProdQty should be filled with the initial (requested) quantity. Example: customers asks for 100 parts. This goes into ProdQty. PROPOS does not count lasered parts. Start quantity is 105 parts because risk of scrap. Then the field ProdStartQty should be 105.

* please contact PROPOS software for more information about synchronising PO's.

Production order line fields

Field	Unique	Description	Data type	(Max) Length	Required	Format / value / remarks
PONr	Yes	Production order (PO) number	Varchar	50 char	Yes	
PlanningNr	Yes	Main planning number	Int	32 bits		
Sequence	Yes	Sequence of the production step for a specific production order	Int	32 bits	Yes	
CellName	No	Name of cell or workstation	Varchar	50 char	Yes	
Operation	No	Name of a specific operation	Varchar	50 char	No	
PlanStartDT	No	<i>Reserved for future use</i>	Datetime	16 char	No	yyyy/mm/d d hh:nn
PlanEndDT	No	<i>Reserved for future use</i>	Datetime	16 char	No	yyyy/mm/d d hh:nn
ActualStartDT	No	Actual start date and time	Datetime	16 char	No	yyyy/mm/d d hh:nn
ActualEndDT	No	Actual end date and time	Datetime	16 char	No	yyyy/mm/d d hh:nn
SetupTimeMin	No	Setup time of the production step	Int	32 bits	Yes	minutes
TactTimeMin	No	Tact time of the production step	Int	32 bits	Yes	minutes
CoolingTimeMin	No	Cooling time / drying time	Int	32 bits	No	minutes
BlockPlanStartDT	No	<i>Reserved for future use</i>	Int	32 bits	No	
ERPLineID	No	ERP unique ID	Varchar	50 char	Yes	
OperationRemarks	No	Remarks (e.g. instructions) for this production step	Text		No	
StockShortage	No	Stock shortage for this step	Int	32 bits	No	-1, 0
StockShortageText	No	Description of stock shortage	Text		No	
CombiID	No	<i>Reserved for future use</i>	Varchar	50 char	No	
UserSort1	No	Custom cell sort field 1	Varchar	50 char	No	
UserSort2	No	Custom cell sort field 2	Varchar	50 char	No	
Attachments	No	Attached files separated by CrLf	Text		No	
MinLeadtimeMin	No	Minimum lead time in minutes.	Int	32 bits	No	
WaitTimeAfterCellMin	No	Wait time after cell before next cell starts in working minutes.	Int	32 bits	No	

Primary key: (PONr, PlanningNr, Sequence)

Further explanation and remarks on production order line fields

Field	Remarks
PONr	PONr, PlanningNr, Sequence form the primary key.
PlanningNr	The planning number, same as for the PO.
Sequence	Sequence of the production step, starting with 1.
CellName	Name of the cell, workstation or operation as defined in PROPOS and in the ERP, e.g. 'DRILLING', 'SAWING' or 'BLUE', 'YELLOW'.
Operation	Name of a specific operation within a cell. Can be used to subdivide the cell into different operations.
SetupTimeMin	The precalculated setup time of the production step in minutes.
TactTimeMin	The precalculated total tact time of the production step in minutes. E.g. if 1 product has a tact time of 10 seconds within a cell and there are 100 products to be processed in that cell, the value should be $10 \text{ sec} * 100 \text{ products} / 60 \text{ sec per minute} = 16.67 = 17 \text{ minutes}$.
CoolingTimeMin	The cooling time can be used for cooling down, warming up, drying, etc. This time is independent from working hours. If POLineSteps are not used, cooling time will be planned after the tact time, before the PO goes to the next cell.
ERPLineID	The ERP internal production order line ID. This must be a unique ID or a unique ID within the PO. This value is used for main PO / sub production order (sub PO) control (synchronization). If not available it can be equal to the value of the Sequence field.
StockShortage	There is a stock shortage for this step. -1 = yes, 0 = no.
StockShortageText	A description of the material that is missing / insufficient.
UserSort1	Custom color code / raw material code / material thickness / ...
UserSort2	Custom color code / raw material code / material thickness / ...
Attachements	Files + paths with pictures / Excel sheets / PDF files separated by CrLf (0Dh & 0Ah).
MinLeadtimeMin	Minimum lead time in minutes (white space + black space) for this PO in this cell. Can be used to reserve enough lead time e.g. for engineering, purchase, etc. Number of minutes in working minutes per day. So 4 weeks lead time at 8 working hours a day would become $4 * 5 * 8 * 60 = 9600 \text{ minutes}$.
WaitTimeAfterCellMin	The number of working minutes wait time after the cell. This actually creates a 'gap' in the planning between two cells. Where lead time for CoolingTimeMin is calculated using 24 hours per day and 7 days per week, the lead time for WaitTimeAfterCellMin is calculated based on working days (Mon-Fri) and on the number of working hours per day for that cell.

Extended XML layout for the PROPOS Employee Registration (MWR) module

The PROPOS Employee Registration module will allow you to define extra steps within cells. These extra steps are for employee registration purposes only, not for planning purposes.

The extra information can be inserted in the XML file, one level deeper than the POLine level.

XML layout (PROPOS employee registration module)

```
<?xml version="1.0" encoding="UTF-8"?>
<dataroot>
  <T_imp_PO>
    [PO 1 data]
  </T_imp_PO>
  <T_imp_POLine>
    [PO 1 line 1 data]
  </T_imp_POLine>
    <T_imp_POLineSteps>
      [PO 1 line 1 step 1 data]
    </T_imp_POLineSteps>
    <T_imp_POLineSteps>
      [PO 1 line 1 step n data]
    </T_imp_POLineSteps>
  <T_imp_POLine>
    [PO 1 line n data]
  </T_imp_POLine>
    <T_imp_POLineSteps>
      [PO 1 line n step 1 data]
    </T_imp_POLineSteps>
    <T_imp_POLineSteps>
      [PO 1 line n step n data]
    </T_imp_POLineSteps>
</dataroot>
```

Layout details (PROPOS Employee Registration module):

```
<?xml version="1.0" encoding="UTF-8"?>
<dataroot>
  <T_imp_PO>
    <PONr>[data]</PONr>
    .....
  </T_imp_PO>
  <T_imp_POLine>
    <PONr>[data]</PONr>
    .....
  </T_imp_POLine>
  <T_imp_POLineSteps>
    <PONr>[data]</PONr>
    <PlanningNr>[data]</PlanningNr>
    <Sequence>[data]</ Sequence >
    <Step>[data]</Step>
    <Operation>[data]</Operation>
    <SetupTimeMin>[data]</ SetupTimeMin >
    <TactTimeMin>[data]</TactTimeMin>
    <CoolingTimeMin>[data]</ CoolingTimeMin >
    <ERPLineID>[data]</ERPLineID>
    <ParallelResourceID>[data]<ParallelResourceID>
  </T_imp_POLineSteps>
</dataroot>
```

Production order line step fields

Field	Unique	Description	Data type	(Max) length	Required	Remarks
PONr	Yes	Production order (PO) number	Varchar	50 char	Yes	
PlanningNr	Yes	The planning number	Int	32 bits		
Sequence	Yes	Sequence, same as POLine Sequence	Int	32 bits	Yes	
Step	Yes	The number of the step	Int	32 bits	Yes	
Operation	No	The operation description of the step	Varchar	50 char	Yes	
SetupTimeMin	No	The Setup time of the production step	Int	32 bits	Yes	minutes
TactTimeMin	No	Tact time of the production step	Int	32 bits	Yes	minutes
CoolingTimeMin	No	Cooling down / warming up / drying.	Int	32 bits	No	
PlanStartDT	No	<i>Reserved for future use</i>	Datetime	16 char	No	yyyy/mm/dd hh:nn
PlanEndDT	No	<i>Reserved for future use</i>	Datetime	16 char	No	yyyy/mm/dd hh:nn
ActualStartDT	No	<i>Reserved for future use</i>	Datetime	16 char	No	yyyy/mm/dd hh:nn
ActualEndDT	No	<i>Reserved for future use</i>	Datetime	16 char	No	yyyy/mm/dd hh:nn
ERPLineID	No	ID for this step in ERP. Please make sure this is a unique value within the POLine.	Varchar	50 char	Yes	
ParallelResourceID	No	ID for parallel resources	Int	32 bits	No	

Primary key: (PONr, PlanningNr, Sequence, Step)

Further explanation and remarks on production order line step fields

Field	Remarks																																																		
PONr	PONr, Sequence, Step form the primary key.																																																		
Sequence	PONr, Sequence, Step form the primary key. Same as POLine Sequence.																																																		
Step	PONr, Sequence, Step form the primary key. Step of the operation in this sequence.																																																		
Operation	Description of the step, e.g. 'DRILLING', 'SAWING'.																																																		
SetupTimeMin	The precalculated setup time of this step in minutes.																																																		
TactTimeMin	The precalculated total tact time of this step in minutes. E.g. if 1 product has a tact time of 10 seconds and there are 100 products to be processed in this step, the value should be $10 \text{ sec} * 100 \text{ products} / 60 \text{ sec per minute} = 16.67 = 17 \text{ minutes}$.																																																		
ERPLineID	The ERP internal line ID for the step. This can be a unique ID or a unique ID within a PO (line). If not available it can be equal to the value of the Step field.																																																		
ParallelResourceID	<p>Steps with the same ParallelResourceID represent steps/resources that are being executed parallel within this POLine. For this field you must use an integer e.g. '10', '20' etc. If ParallelResourceID is left empty, the step will be considered as serial. Example. For a certain POLine, you are going to do some drilling first. After that, you need 2 machines at the same time for 8 hours and 1 man for 4 hours to keep the machines running. You need half an hour of setup time for that. After that, you need a man to clean up the shop floor for half an hour. We will send to PROPOS:</p> <table border="1"> <thead> <tr> <th>POLineStep</th> <th>Operation</th> <th>Setup (hrs)</th> <th>Takt (hrs)</th> <th>ParallelResourceID</th> </tr> </thead> <tbody> <tr><td>1</td><td>Drilling</td><td>0.25</td><td>2</td><td>10</td></tr> <tr><td>2</td><td>Man</td><td>0.25</td><td>2</td><td>10</td></tr> <tr><td>3</td><td>Machine1</td><td>0.5</td><td>0</td><td>20</td></tr> <tr><td>4</td><td>Machine2</td><td>0.5</td><td>0</td><td>20</td></tr> <tr><td>5</td><td>Man</td><td>0.5</td><td>0</td><td>20</td></tr> <tr><td>6</td><td>Machine1</td><td>0</td><td>8</td><td>30</td></tr> <tr><td>7</td><td>Machine2</td><td>0</td><td>8</td><td>30</td></tr> <tr><td>8</td><td>Man</td><td>0</td><td>4</td><td>30</td></tr> <tr><td>9</td><td>Man</td><td>0</td><td>0.5</td><td></td></tr> </tbody> </table> <p>For line 9, we can leave ParallelResourceID blank. For PROPOS this means not parallel. We can also fill in '40' (or any integer other than 30).</p> <p>The lead time for this set of operations is 2.25 for lines 1-2 plus .5 for lines 3-5 plus 8 for lines 6-8 plus .5 for line 9 = 11.25 hours.</p> <p>One level higher - at POLine level - you have to fill in this calculated time of 11.25 hours. You can set setup time to 0 and use 11.25 as takt time or you can fill in 0.75 for setup time and 10.5 for takt time.</p> <p>Remark: If there is a ParallelResourceID '10' in step 1 and '20' in step 2 and then '10' again in step 3, steps 1 and 3 will not be seen as parallel resources. PROPOS will create a new parallel group each time ParallelResourceID changes or is left empty.</p>	POLineStep	Operation	Setup (hrs)	Takt (hrs)	ParallelResourceID	1	Drilling	0.25	2	10	2	Man	0.25	2	10	3	Machine1	0.5	0	20	4	Machine2	0.5	0	20	5	Man	0.5	0	20	6	Machine1	0	8	30	7	Machine2	0	8	30	8	Man	0	4	30	9	Man	0	0.5	
POLineStep	Operation	Setup (hrs)	Takt (hrs)	ParallelResourceID																																															
1	Drilling	0.25	2	10																																															
2	Man	0.25	2	10																																															
3	Machine1	0.5	0	20																																															
4	Machine2	0.5	0	20																																															
5	Man	0.5	0	20																																															
6	Machine1	0	8	30																																															
7	Machine2	0	8	30																																															
8	Man	0	4	30																																															
9	Man	0	0.5																																																
CoolingTimeMin	This time is independent from working hours in planning. At POLine level, enter the total amount of cooling time of the POLineSteps. Try to keep cooling time the last POLineStep. If during cooling you want the resource to be occupied, fill TactTimeMin with the same number as the CoolingTimeMin. But in that case make sure not to add the TactTimeMin to the TactTimeMin at POLine level because that will result in double lead time at that level (TactTimeMin+ CoolingTimeMin).																																																		