

# **Machine industry task**

# Contents

**Machine industry task.....3**

# Machine industry task



DITA 1.2 introduces a task specialization for the machine industry. This task is specialized from the new general task that has become available in DITA 1.2 and that relaxes the definition of the strict task model in DITA 1.1. The general tasks allows more options in the procedure header as well as in procedure steps. Compared to the general task, the machine industry task specialization introduces preliminary and closing requirements with more semantics.



## Note:



In this article, I am using a method to illustrate DITA language specifications in block diagrams. An article explaining the meanings of graphical elements and colors in that method is available as [<pointer to article>](#).

## Purpose



In previous releases of the DITA standard, there was little support for the specific type of documentation required in the machine industry. The most important aspect that made DITA of little practical value for machine documentation was the impossibility to include a safety notice prior to the command in a procedure step. Warning personnel of imminent danger after the command is already given is considered to be an unsafe method and is discouraged in the industry.



Also, many machine manuals list categorized requirements in the header of each task, indicating required personnel, equipment, supplies etc. In many publications, the categories are indicated by icons or standard headings. In the `prereq` element of the previous DITA release, there was no differentiation for various types of preliminary requirements.



## General, strict and machinery tasks

In DITA 1.2, the original task from DITA 1.0 and 1.1 is replaced with a relaxed general task model, from which the earlier strict task is then created via constraints. This offers two task models. The machine industry task was specialized from the general task and inherits the relaxed definition of elements contained in the general task.



The main relaxations relate to the ordering of `prereq` and `context` and add a `section` element to that first part of the `taskbody`. To allow any order in these elements, any number of choices can be included. Also, the number of `example` and `postreq` elements is not restricted in the general task model.

The most important change for the machine industry is in the `step` element. This now allows a `note` or `hazardstatement` before the `cmd`. This change in the `step` element is discussed further below.

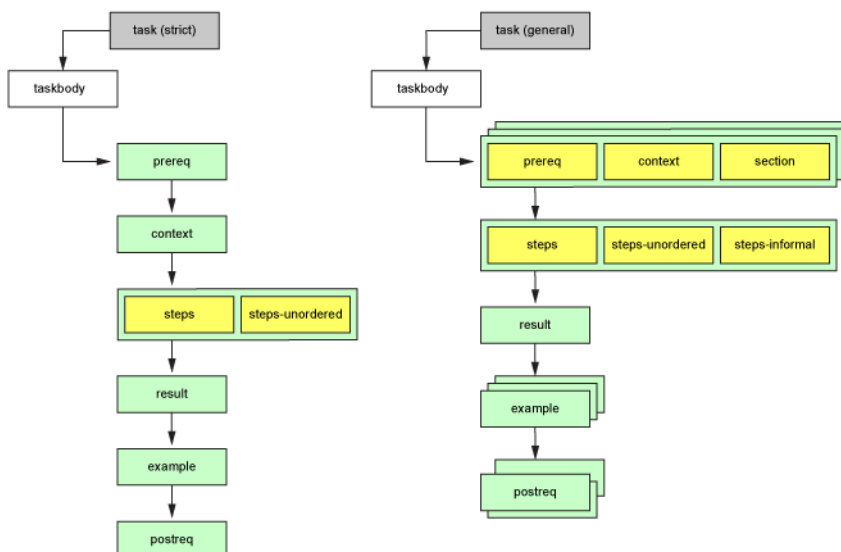


Figure 1: Block diagram comparing the general and strict task elements

But even though the general task model of DITA 1.2 does make many things easier for authors in the machinery industry, a number of important elements were still missing. In the machine industry `taskbody` specialization, two elements from the general `taskbody` are therefore replaced with elements that allow a more clearly defined internal structure.

The `prereq` element is replaced by `prelreqs`. This new element has a much more defined structure and envelopes all preliminary requirements. Even though it is possible to include multiple `prelreqs` in a machinery task, this should not be necessary: it is merely a side effect of allowing `prelreqs`, `context` and `section` elements in any order.

At the end of the machinery `taskbody`, the collection of zero or more `postreq` elements from the general task is replaced by one single `closereqs` element. In fact, even though the `prelreqs` element may occur more than once in a single machinery task, this is not considered good practice. The reason `prelreqs` can be replicated is simply that there may be a `prelreqs`, `context` and `section` element in any order.

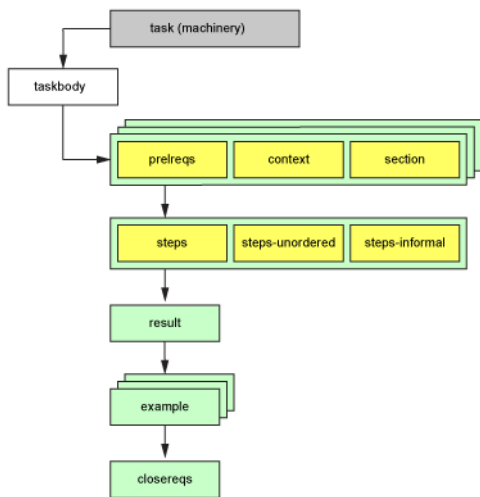


Figure 2: Block diagram of the machinery task

### Preliminary requirements

Many machine manuals include more or less standardized indications of required personnel, equipment, tools and supplies at the start of each procedure. The new `prelreqs` element in DITA 1.2 is designed to meet these requirements.

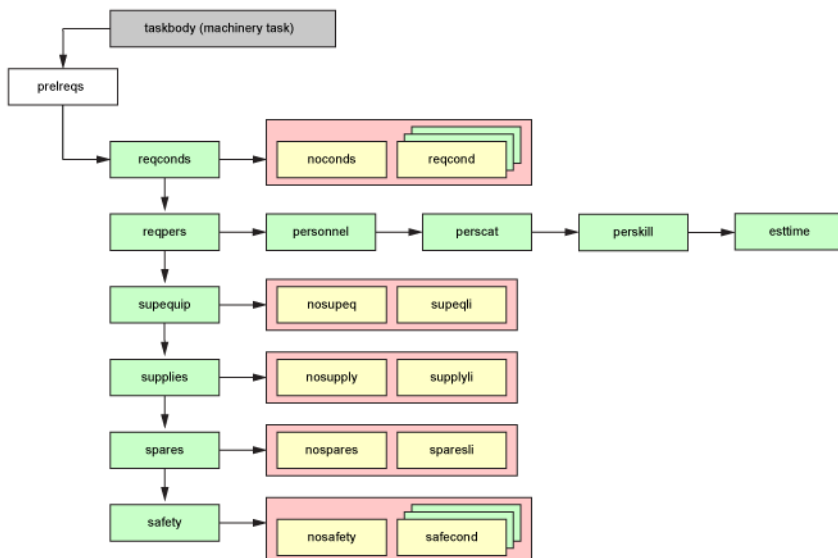


Figure 3: Block diagram of the prelreqs element

Each optional child except the `reqpers` contains either an empty element (such as `noconds` or `nospares`) or a list of conditions, support equipment etc. The `reqpers` element is used to indicate the number of people required to do the job (given in the personnel element), followed by optional indicators for their categories and skill levels, as well as the estimated time required to do the job.

In the output, the various categories inside the `prelreqs` may be transformed into icons or standard headings, depending on each individual company's style guide. The location of each element in the output will be defined by that style guide, rather than by the fixed order in which they appear in the DITA source.

The availability of empty elements for each of the preliminary requirements categories allows inclusion of that information in the authoring process, even if you do not want to have those elements appear in the final output. It is also possible to transform empty elements into standard sentences that indicate to the personnel that no specific equipment or supplies are required to perform the work. After all, the absence of specific items (such as supplies in the above example) may be informative, too.

### Example: Preliminary requirements for a maintenance task

```
prelreqs>
  <reqconds>
    <reqcond>Remove the covers.</reqcond>
  </reqconds>
  <reqpers>
    <personnel>2</personnel>
    <perscat>Electrical engineer</perscat>
    <esttime>15 minutes</esttime>
  </reqpers>
  <supequip>
    <supeqli>
      <supqui>Multi-meter</supqui>
    </supeqli>
  </supequip>
  <supplies>
    <nosupply/>
  </supplies>
</prelreqs>
```

The above section may be transformed into a section at the top of a procedure that looks like shown in this example output:

## Checking the batteries



1 Electrical engineer



15 minutes



None



Multi-meter

### Preparations:

Remove the covers.

### Procedure:

1. Connect the black probe of the multi-meter to the machine frame.

Figure 4: Possible styling of `prelreqs` elements in a PDF




### Note:



The above image is a mock-up. It merely shows the potential of having specific DITA elements automatically transformed into cues for the reader. The code required to transform the DITA content into a specific PDF or other

format should be developed by experts and based on the particular company's style guide. Default rendering by the DITA Open Toolkit will not yield the above results.



### Closing requirements


 At the end of a procedure, some further work may be required (such as test procedures, safety-related finishing work, reporting etc.). These can be placed inside the `closereqs` element at the `end` of the `taskbody` in a machinery task. This single `closereqs` element replaces the `postreq` elements in the general task.

The only (optional) element inside the `closereq` is `reqconds`, which either contains an empty `noconds` element or a list of one or more `reqcond` elements. Instead of the `reqcond` element, a `reqcondp` can be used to point to an external publication that specifies conditions or regulations.

 In the output, the `closereqs` should show up as an unordered list with a standard heading that identifies the list items as work that must be done after the procedure has finished (such as cleaning up, reporting, testing, adjusting, etc.). Normally, the steps listed in the `closereqs` are not part of the procedure but procedures in themselves, that can be referenced from the `reqcond` elements. 

### Safer steps

 One of the important limitations of the task model in DITA 1.1 was the inability to include important warnings prior to the command that they apply to. In the machine industry, it is considered much safer to give the warning first, as it might be too late to warn the reader once the command is read and executed. Often, procedures are executed step by step and reading ahead is not common practice, especially in emergency situations.  Therefore, listing the relevant safety warnings immediately prior to a command is considered good practice. In DITA 1.2, there are several options for this.





 First of all, the `steps` and `steps-unordered` elements in DITA 1.2 have received an optional opening element, `stepsection`, in which content can be placed that precedes the first step in the procedure. This `stepsection` can be used to show content (such as text, an image, notes, hazard statements) that must precede the first step in the procedure.



#### Note:

This location, at the beginning of a procedure, is different from the location in the safety conditions prior to the procedure - listed in the `prelreqs` element. The procedure will normally start with a standard heading or icon identifying the start of the procedure. The `stepsection` element is shown after that heading. This increases the chance that the content will actually be read.

## Checking the batteries

	1 Electrical engineer		15 minutes
	None		Multi-meter

### Preparations:

Remove the covers.

### Safety precautions:

The machine must be switched off.

<safecond>

<stepsection>


### Procedure:

- Make sure the power switch is in the OFF position and place your personal padlock to prevent the machine from being switched on.
1. Connect the black probe of the multi-meter to the machine frame.

**Figure 5: Safety notes in a safety and a stepsection element**

Apart from the notes at the start of a procedure, there might be notes that must be included in a particular step before the actual command is given. This has become possible in the new definition of the `step` in DITA 1.2. The `step` now starts with optional `note` and/or `hazardstatement` elements.

<step>

 **WARNING !**  
Keep your fingers away from the left side of the splicer bracket. The tape holder swings out of the splicer fast.

**4. Set the tape holder release pin**  
Slowly turn the tape holder release pin up [4] against the retaining lever [3] until the tape bracket [1] swings out of the splicer. Fasten the nut [6] in this position.

**Figure 6: Using a hazardstatement in a procedure step**

In this example, the warning will be read before step 4 is executed. This is enforced by placing the `hazardstatement` before the `cmd`. Otherwise, the service engineer might not have any fingers left to keep away from the dangerous machine part.

The `hazardstatement` can also be placed after the `cmd`. This may be useful when warning against something that should be avoided at the end of a step (such as discarding a part that will be useful later). Also, informative notes may be better placed after the command.



### Note:

The `hazardstatement` is new in DITA 1.2 and of great importance for the machinery task. It is described in a separate feature article.

## Conclusion

With the relaxation of the basic task model in DITA 1.2 and subsequent specialization of the `taskbody` for machinery tasks, DITA has become a viable alternative for documentation in the machinery domain. It enables documentation departments in the machinery domain to follow the modular design practices that have been in place in the CAD engineering departments since decades.

