KISSsoft 03/2014 – Instruction 093

Define rules

KISSsoft AG
Rosengartenstrasse 4
8608 Bubikon
Switzerland

Tel: +41 55 254 20 50
Fax: +41 55 254 20 51
info@KISSsoft.AG
www.KISSsoft.AG

Contents

1 Rules

1.1 Introduction / manual text

Rules are used to ensure that user specific guidelines for the validity ranges of parameters are applied and adhered to. This typically concerns the maximum and minimum limits of input values, calculated values and the relationships between these values i.e. length-width relationships, length-diameter relationships or even the relationship between the module and the center distance. If a rule is violated, (e.g. a rule was defined that profile shift should not be negative but the user enters a negative profile shift), then, KISSsoft will issue a message.

These rules are defined by being stored in a <module>.rls file, where <module> stands for the KISSsoft description of the calculation module, e.g. Z012 for cylindrical gear pairs.

These rules are subdivided into those that must be fulfilled before the calculation is performed and those that must be checked afterwards. If a rule is infringed, the appropriate messages can be displayed. In the case of rules that must be checked before the calculation, variables can also be set to constant or calculated values.

The following statements are possible:

- precalc: marks the beginning of the rules that are checked before a calculation is performed
- postcalc: marks the beginning of the rules that are checked after a calculation
- assert(<condition>): The <condition> is ensured. In this case, the <condition> usually represents a comparison, where both the right-hand and left-hand side of the comparison can also be calculated.
- action msg <message>: If the <condition> of the previous assert has not been fulfilled, the <message> is output. Here the <message> can include variables, in the same way as report templates.
- action set <assignment>: If the <condition> of the previous assert has not been fulfilled, the <assignment> is performed. The assigned value can be a constant or be calculated from variables, in the same way as for the report templates.

An assignment only makes sense in the precalc section because changing the contents of variables after the calculation merely leads to inconsistent results and has no other effects.

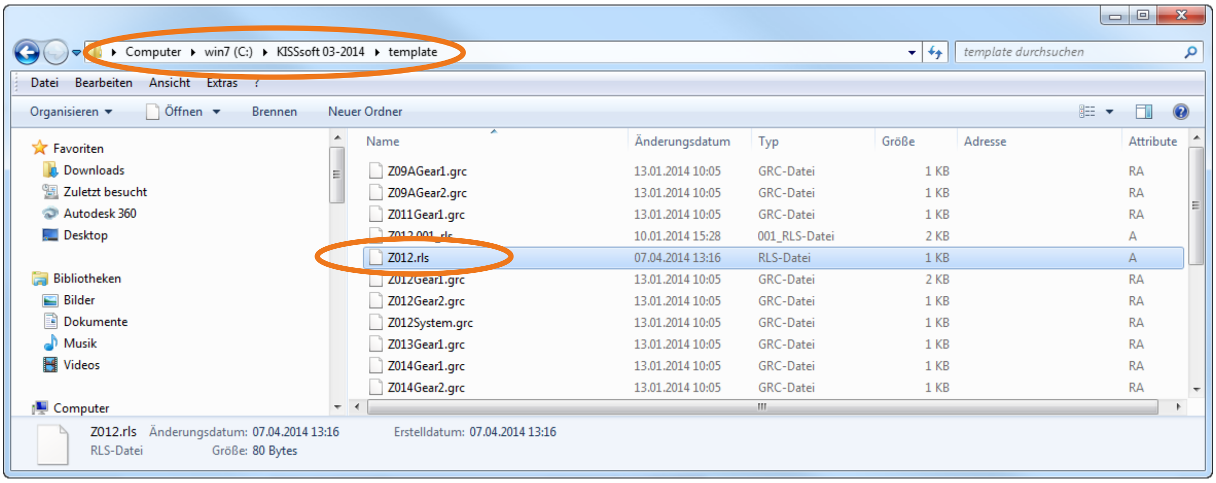
1.2 Location and type of rules file

A rules file is a text file with the name <module>.rls where <module> is the name of the calculation module for which it is used, e.g. Z12 for gear pair calculation. So, a file Z012.rls is a rules file for the gear pair calculation.
Save the rules files in the folder …\KISSsoft 03-2014\template

1.3 Example 1

In this example, it is checked before the calculation that the number of teeth of the pinion (gear 1) is larger than 10.

First, we generate a rules file for the Z12 calculation (gear pair calculation) in the respective folder:



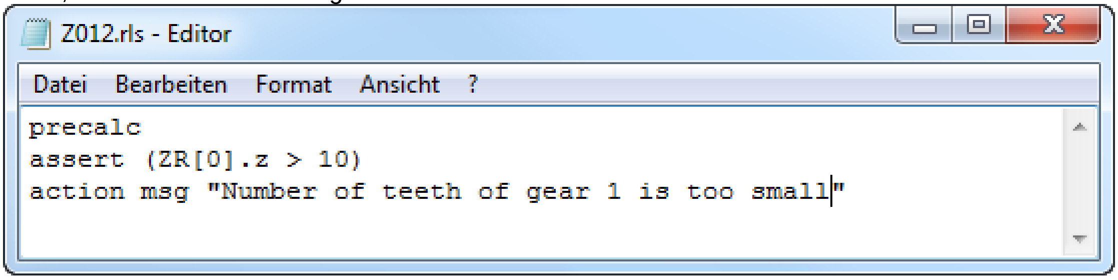
Computer ▸ win7 (C:) ▸ KISSsoft 03-2014 ▸ template
template durchsuchen
Datei Bearbeiten Ansicht Extras ?
Organisieren ▾ Öffnen ▾ Brennen Neuer Ordner
Favoriten
Downloads
Zuletzt besucht
Autodesk 360
Desktop
Bibliotheken
Bilder
Dokumente
Musik
Videos
Computer

Name	Änderungsdatum	Typ	Größe	Adresse	Attribute
Z09AGear1.grc	13.01.2014 10:05	GRC-Datei	1 KB		RA
Z09AGear2.grc	13.01.2014 10:05	GRC-Datei	1 KB		RA
Z011Gear1.grc	13.01.2014 10:05	GRC-Datei	1 KB		RA
Z012.001_.rls	10.01.2014 15:28	001_RLS-Datei	2 KB		A
Z012.rls	07.04.2014 13:16	RLS-Datei	1 KB		A
Z012Gear1.grc	13.01.2014 10:05	GRC-Datei	2 KB		RA
Z012Gear2.grc	13.01.2014 10:05	GRC-Datei	1 KB		RA
Z012System.grc	13.01.2014 10:05	GRC-Datei	1 KB		RA
Z013Gear1.grc	13.01.2014 10:05	GRC-Datei	1 KB		RA
Z014Gear1.grc	13.01.2014 10:05	GRC-Datei	1 KB		RA
Z014Gear2.grc	13.01.2014 10:05	GRC-Datei	1 KB		RA

Z012.rls Änderungsdatum: 07.04.2014 13:16 Erstelldatum: 07.04.2014 13:16
RLS-Datei Größe: 80 Bytes

Figure 1. Creating a rules file in the respective folder

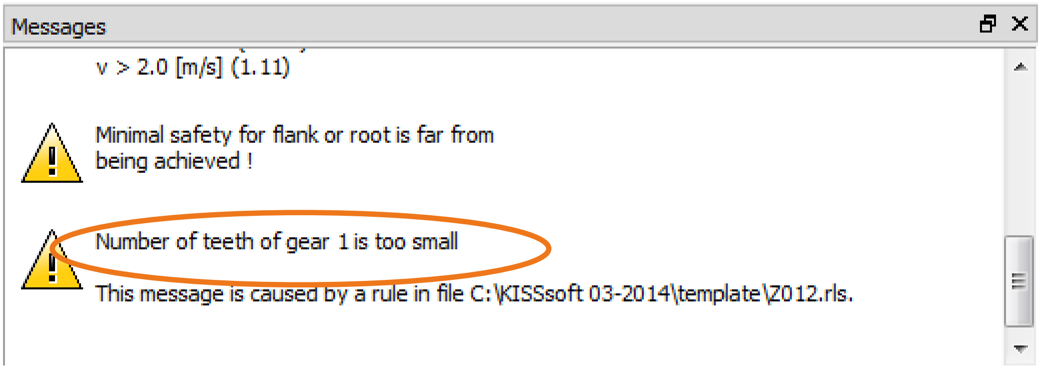
Then, we define the rule using the above listed commands:



```
precalc
assert (ZR[0].z > 10)
action msg "Number of teeth of gear 1 is too small"
```

Figure 2. Rule definition that number of teeth of pinion should be more than 10

When you run the calculation where the gear no. 1 has less than 10 teeth, the warning message in the window “Messages” appears:



v > 2.0 [m/s] (1.11)

Minimal safety for flank or root is far from being achieved !

Number of teeth of gear 1 is too small

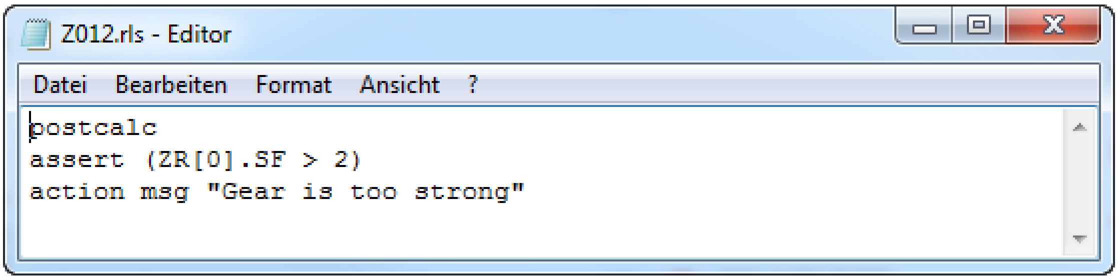
This message is caused by a rule in file C:\KISSsoft 03-2014\template\Z012.rls.

Figure 3. Warning message shown.

1.4 Example 2

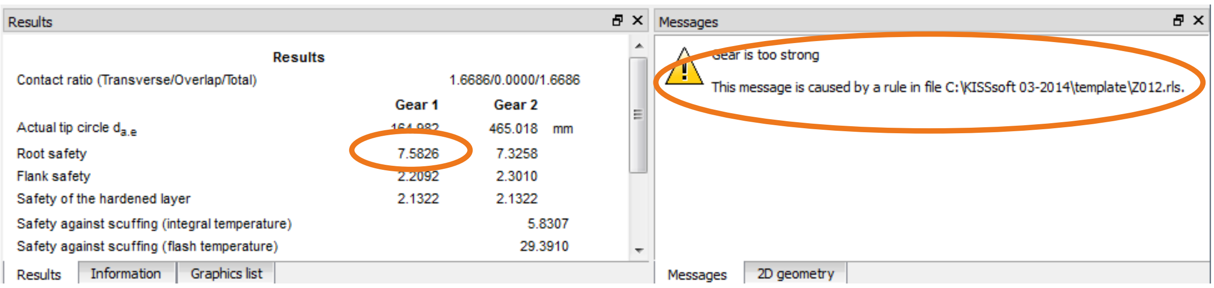
In the below example, a message is shown if after the calculation, the safety factor are higher than e.g. 2:

The difference to the above example is that the rules are checked after the calculation with respect to the result:



```
postcalc
assert (ZR[0].SF > 2)
action msg "Gear is too strong"
```

Figure 4. rule that checks after the calculation is executed whether the root safety factor of gear 1 is more than 2



```
Results
Contact ratio (Transverse/Overlap/Total)              1.6686/0.0000/1.6686
                                            Gear 1      Gear 2
Actual tip circle d_a.e                     164.982     465.018   mm
Root safety                                 7.5826      7.3258
Flank safety                                2.2092      2.3010
Safety of the hardened layer                2.1322      2.1322
Safety against scuffing (integral temperature)             5.8307
Safety against scuffing (flash temperature)                29.3910
```

```
Gear is too strong
This message is caused by a rule in file C:\KISSsoft 03-2014\template\Z012.rls.
```

Figure 5. Gear calculation where safety factor SF1>2, and corresponding message

1.5 Example 3

Here is an example file for a helical gear calculation where some conditions are checked before the calculation and some after the calculation:

```
precalc
assert (ZR[0].x.nul < 1)
action msg "Profile shift for Gear 1 too big, is {ZR[0].x.nul}, maximum 1. It has been reset to 1."
action set ZR[0].x.nul = 1
assert (ZR[1].x.nul < 1)
action msg "Profile shift for Gear 2 too big, is {ZR[1].x.nul}, maximum 1. It has been reset to 1."
action set ZR[1].x.nul = 1
postcalc
assert ((ZP[0].a/ZS.Geo.mn) < 200)
action msg "Center distance too big for module a={ZP[0].a}, mn={ZS.Geo.mn}, a/mn={ZP[0].a/ZS.Geo.mn})."
```

Explanations:
The "precalc" statement opens the section of the rules that must be executed before the calculation.
The first "assert" statement checks whether the nominal profile shift of gear 1 is less than 1.0.
If this "assert" is not fulfilled, the "action msg" statement outputs the message that the profile shift is too big, displays the current value and tells the user that the profile shift has been set to 1.0.
The "action set" then sets the profile shift to 1.0.
The second "assert" statement checks the same values for gear 2.
The "postcalc" statement signifies the end of the set of rules to be executed before the calculation and opens the section of the rules to be checked after the calculation. The example shows a definition of an "assert" statement. This checks the ratio from the center distance to the module. If the rule is infringed, the "action msg" statement triggers a message. However, there is no point in converting one of these two values after the calculation and this is why the "action set" statement is not present here.
The "General – Results and Reports – Report templates – Formatting – Calculation variables" section details which operators and functions can be used in the formulae.
The file containing the rules is stored in the template directory (TEMPLATEDIR, usually the "template" subdirectory, see the "Directory structure" section in the "Setting Up KISSsoft" section). As the template directory can also be project-specific, you can also define project-specific rules.

1.6 Example 4

A company wishes to have specific values for the outer diameter tolerances, depending in the module. A rules file Z012.rls with this context fulfills the requirements:

```
precalc
assert (ZS.Geo.mn >= 0.9)
action set ZR[0].Ada.E = 0
action set ZR[0].Ada.i = -0.05
action set ZR[1].Ada.E = 0
action set ZR[1].Ada.i = -0.05
action set ZR[2].Ada.E = 0
action set ZR[2].Ada.i = -0.05
action set ZR[3].Ada.E = 0
action set ZR[4].Ada.i = -0.05

assert (ZS.Geo.mn < 0.9 || ZS.Geo.mn >=3)
action set ZR[0].Ada.E = 0
action set ZR[0].Ada.i = -0.1
action set ZR[1].Ada.E = 0
action set ZR[1].Ada.i = -0.1
action set ZR[2].Ada.E = 0
action set ZR[2].Ada.i = -0.1
action set ZR[3].Ada.E = 0
action set ZR[3].Ada.i = -0.1

assert (ZS.Geo.mn < 3 || ZS.Geo.mn >= 7)
action set ZR[0].Ada.E = 0
action set ZR[0].Ada.i = -0.2
action set ZR[1].Ada.E = 0
action set ZR[1].Ada.i = -0.2
action set ZR[2].Ada.E = 0
action set ZR[2].Ada.i = -0.2
action set ZR[3].Ada.E = 0
action set ZR[3].Ada.i = -0.2

assert (ZS.Geo.mn < 7)
action set ZR[0].Ada.E = -0.1
action set ZR[0].Ada.i = -0.3
action set ZR[1].Ada.E = -0.1
action set ZR[1].Ada.i = -0.3
action set ZR[2].Ada.E = -0.1
action set ZR[2].Ada.i = -0.3
action set ZR[3].Ada.E = -0.1
action set ZR[3].Ada.i = -0.3
```

Remarks:

1) The statement „assert (ZS.Geo.mn>=0.9)“ leads the block for the case, that the module is smaller than 0.9. Since the assert statement is intended to assert a certain condition, in this case the logic has to be inverted.
2) The tolerances are set for a total of four gears. This is to ensure compatibility with other gear modules of KISSsoft, e.g. planetary drives or three gears. If you only want to apply these rules to the gear pair module, you can delete the lines with ZR[2] and ZR[3].