

LoCo Release Notes

Release 1.42

info@scale.eu www.scale.eu

SCALE 

IT-Solutions for CAE

| | | |
|----------|--|-----------|
| 1 | Neue Features | 1 |
| 1.1 | LoCo benötigt zum Starten eine gültige Lizenzdatei | 1 |
| 1.2 | Hinweis bei Ablauf von Berechtigungen | 1 |
| 1.3 | Enforceable attributes | 2 |
| 1.4 | Konfigurationsmöglichkeit für Liste erlaubter Tags | 4 |
| 1.5 | Konfiguration von userspezifischen Parametern, die global im LoCo zur Verfügung stehen | 4 |
| 1.6 | Mehrstufiger Modellaufbau (Multi Stage Assembly) | 5 |
| 1.7 | Vergleichen von Parametergruppen | 6 |
| 1.8 | Studies sind nun Teil der Pool Version | 7 |
| 1.9 | Komponenten duplizieren | 12 |
| 1.10 | Bestehende Komponente bei Import ersetzen | 13 |
| 1.11 | Löschen bzw. Deaktivieren von Usern | 13 |
| 1.12 | Reports und Bilder als sticky markieren | 14 |
| 1.13 | CSV-Export der transponierten Darstellung von Parametertabellen | 14 |
| 1.14 | Suche nach nicht sichtbaren Teilen vom History-Kommentar im History-Graph | 15 |
| 1.15 | Neuer Kontextmenü-Eintrag <i>Jump back to last used version</i> | 15 |
| 1.16 | Auswahl der Namensschema kontextabhängig | 15 |
| 1.17 | <i>Jump to Version</i> auf einem Include zeigt die Pool-Version an | 16 |
| 2 | Weitere Verbesserungen | 17 |
| 3 | Bugfixes | 19 |

Neue Features

1.1 LoCo benötigt zum Starten eine gültige Lizenzdatei

Ab Version 1.42 benötigt LoCo eine gültige Lizenzdatei zum Starten.

#2116

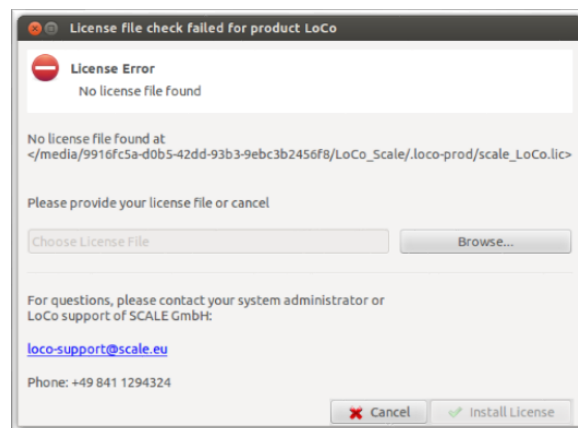


Abb. 1.1: Überprüfen der Lizenzdatei

Eine Lizenzdatei wird nur für LoCo-Versionen benötigt, welche für EDL von AUDI zur Verfügung gestellt werden. Die Lizenz wird bei jedem Start überprüft. Dabei sucht LoCo in folgenden Verzeichnissen nach einer Lizenzdatei:

- Linux: `$HOME/.loco-prod`
- Windows: `%APPDATA%\loco-prod`

Alternativ kann der Pfad zur Lizenzdatei auch als Wert für die Umgebungsvariable `SCALE_LOCO_LICENSE` gesetzt werden. Das Überprüfen der Lizenz kann nicht mehr übersprungen werden. Bei Problemen können sich Nutzer an loco-support@scale.eu wenden.

Weiter Informationen zur Lizenz sind im Dokument `LoCo-Client-Lizenzierung.pdf` zu finden.

1.2 Hinweis bei Ablauf von Berechtigungen

31 Tage vor Ablauf von Pool-Berechtigungen wird ein Hinweis-Dialog beim Start der Applikation angezeigt. Die Meldung enthält alle ausgecheckten Pools, die vom Berechtigungsablauf betroffen sind. Je nach Berechtigungsvergabe werden zusätzliche Informationen dargestellt:

#2350

- Berechtigung wurde über Gruppenmitgliedschaften vergeben:
 - group** Gruppe, in welcher der angemeldete Nutzer Mitglied ist und dessen Mitgliedschaft ablaufen wird
 - grantor** Urheber der Berechtigung, d.h. jene Person, welche die Mitgliedschaft in der Gruppe gesetzt hat und bzgl. Verlängerung als Kontaktperson in Frage kommt
 - expire** Zeitpunkt, an welchem die Mitgliedschaft auslaufen wird
- Berechtigung wurde direkt dem Nutzer erteilt:
 - pool** Pool sowie Art der Berechtigung (lesend/schreibend)
 - grantor** Urheber der Berechtigung, d.h. jene Person, welche die Berechtigung gesetzt hat und bzgl. Verlängerung als Kontaktperson in Frage kommt.
 - expire** Zeitpunkt, an welchem die Berechtigungen auslaufen wird.

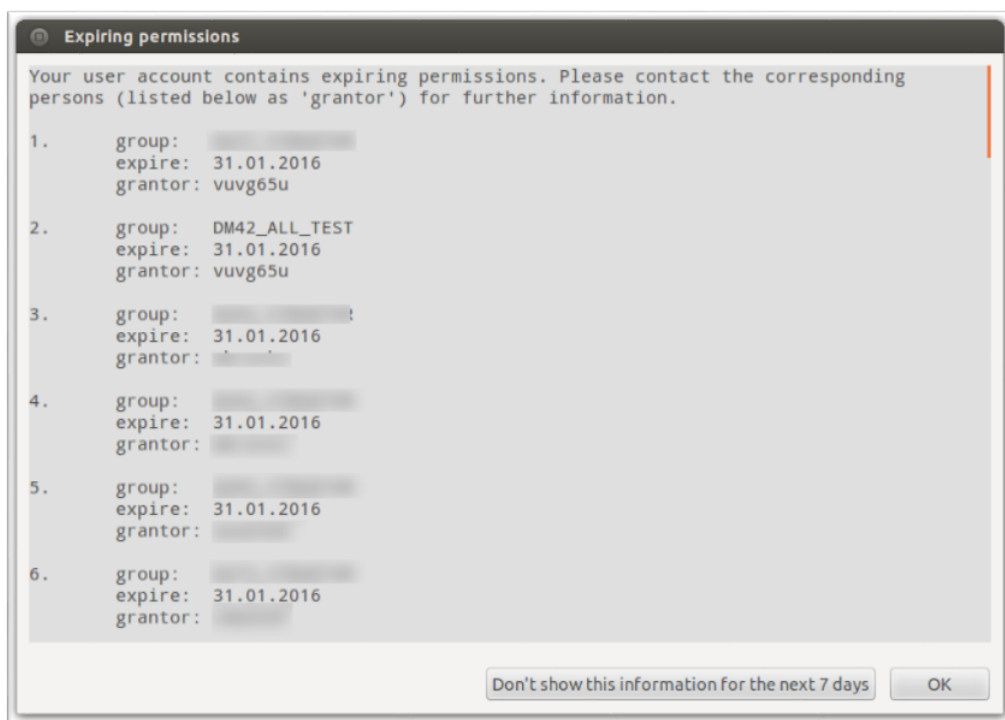


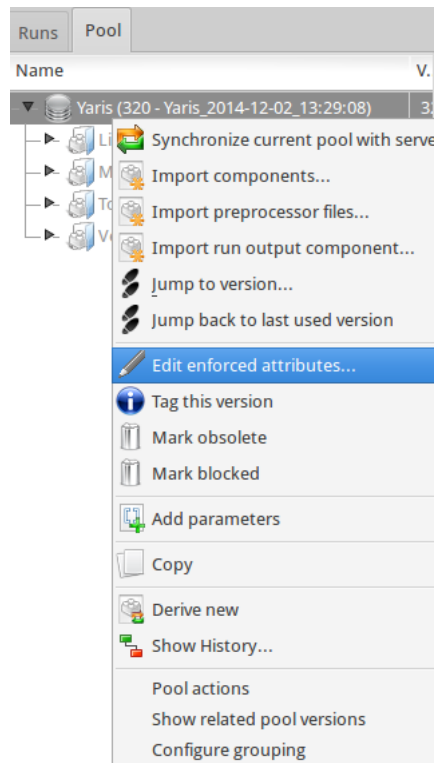
Abb. 1.2: Warnung bei anstehendem Ablauf von Berechtigungen

1.3 Enforceable attributes

Um zu gewährleisten, dass in einer Pool-Version nur Modelle mit bestimmten Attributen aufgebaut werden, steht ab Version 1.42 die Möglichkeit zur Verfügung, die validen Attribut-Belegungen direkt an der Pool-Version zu definieren. Dafür kann über das Kontextmenü einer Pool-Version die Option *Edit enforced attributes* aufgerufen werden. Im dadurch angezeigten Dialog können die Attribut-Belegungen definiert werden, die für alle Runs dieser Pool-Version übernommen werden sollen.

#2407
#2747

Bemerkung: In dieser Version haben die definierten Attribut-Belegungen noch keine Auswirkungen auf die Runs der Pool-Version. Diese werden erst in der kommenden LoCo-Version angewendet.



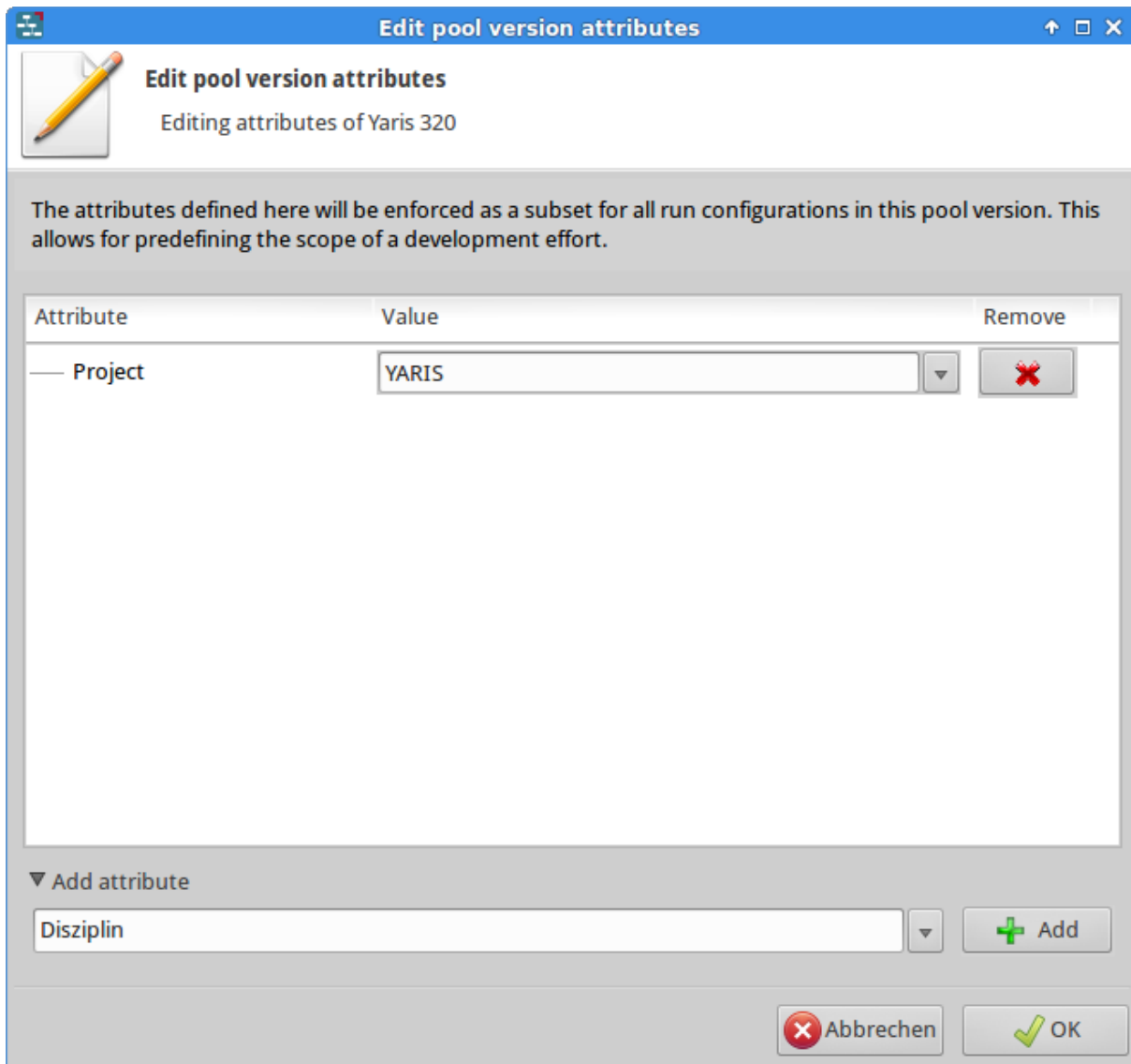


Abb. 1.3: Definition der Attribut-Belegungen, die für alle Runs einer Pool-Version verwendet werden sollen

1.4 Konfigurationsmöglichkeit für Liste erlaubter Tags

KeyUser können eine Liste von erlaubten Tags verwalten. Diese gilt ähnlich wie Attribute und *Name Schema* global für alle Pools. Die Konfiguration wird im XML Format vorgenommen. Der Editor befindet sich unter *View > Window > Administration > Tag Editor*.

#2411

Bemerkung: Bisher wurde nur der Editor für die Tags umgesetzt. Die hier erstellte Liste von Tags wird im LoCo noch nicht verwendet. Dies wird erst in kommenden Releases umgesetzt.

1.5 Konfiguration von userspezifischen Parametern, die global im LoCo zur Verfügung stehen

#1940

Listing 1.1: Konfiguration im XML Format

```
1 <tag-configuration>
2   <tag pattern="reference" regex="false">
3     <color value="#ffee00" />
4   </tag>
5   <tag pattern="user .*" regex="true">
6     <!-- Erlaubt die Vergabe beliebiger Tags, die mit "user " anfangen -->
7   </tag>
8 </tag-configuration>
```

Es können userspezifische Parameter über eine Datei `site_parameters.txt` definiert werden. Diese Datei muss dabei im Verzeichnis mit den Einstellungen vom LoCo liegen.

- Linux: `$HOME/.loco-prod`
- Windows: `%APPDATA%\loco-prod`

Darüber hinaus ist es auch möglich, die Datei für userspezifische Parameter über die Umgebungsvariable `LOCO_SITE_PARAMETERS_FILE` bereitzustellen. Die Parameter in der Datei müssen dabei in der Form: `parameter_name=parameter_value` angegeben werden.

Beispiel: `dyna_license_file='/etc/dyna_license.lic'`

Sollte ein Parameter bereits in LoCo selbst in einer Parametertabelle definiert worden sein, so wird vorrangig der Wert aus der Definition im LoCo verwendet.

Wurde ein Parameter sowohl in der Datei `site_parameters.txt` definiert als auch in einer Datei, welche über die Umgebungsvariable `LOCO_SITE_PARAMETERS_FILE` angezogen worden ist, so wird vorrangig der Wert verwendet, welcher in der Datei `site_parameters.txt` definiert wurde.

1.6 Mehrstufiger Modellaufbau (Multi Stage Assembly)

Bislang bestand der Modellaufbau in LoCo (Assembly) immer genau aus einem Deckaufbau. In diesem wurde die gewünschte Menge an zu verwendenden Komponenten (Includes, Submodelle, Skripte) ermittelt und über ein Template in das gewünschte Solver-Format gebracht.

#138

Jede der dabei verwendeten Komponenten referenziert hierbei Massendaten, welche zum Zeitpunkt des Assemblies bereits vorliegen, da sie explizit durch die Anwender in LoCo bereitgestellt/importiert worden sind.

Der neue *Multi Stage Assembly*-Mechanismus erlaubt nun, dass die Massendaten einer Komponente nicht a priori feststehen, sondern durch eine weitere referenzierte *Run Configuration* generiert werden. Das heißt, der Inhalt einer Komponente entsteht erst durch die Assemblierung einer weiteren *Run Configuration*. Infolgedessen ermöglicht das *Multi Stage Assembly* hierarchische Assemblies, welche sich durch eine Baumstruktur abbilden lassen.

Enthält also beispielsweise eine *Run Configuration*, welche für das Assembly eines Fahrzeugs für eine Strukturcrashsimulation vorgesehen ist, als Komponente ein Rad, das durch eine *Run Configuration* generiert wird, die sich um die parametrische Erzeugung des Rades kümmert, so wird bei der Assemblierung des Fahrzeugmodells zunächst das Assembly für das Rad gestartet und erst dann das eigentliche Fahrzeugmodell für den Crash erstellt. Dabei kann der Modellaufbau des Rades Simulationen enthalten, die beispielsweise den Aufblasvorgang im Rad sowie die Abplattung durch Belastung mit dem zu erwartenden Fahrzeuggewicht abbilden.

Der mehrstufige Modellaufbau kann dabei in beliebig vielen Stufen erfolgen. Denkbar wäre beispielsweise:

- * Fahrzeug
 - * Abplattung vom Rad
 - * Aufblasvorgang vom Rad
 - * Parametrische Generierung des Rades
 - * Einsitzprozess
 - * Dummypositionierung
 - * Sitzpositionierung
 - * Gurtpositionierung
 - * Cockpitzusammenbau
 - * Airbagfaltung

Das Ergebnis der *Sub Assemblies* wird fortan persistiert und synchronisiert, sodass anderen Nutzern entsprechende Ergebnisse der Komponenten, also die entstandenen Massendaten, automatisch zur Verfügung gestellt werden, ohne dass diese selbst nochmals die *Sub Assemblies* starten müssen.

Der gesamte Vorgang geschieht dabei aus Sicht des Nutzers transparent. Im Normalfall wird der Nutzer demnach nicht merken, wie viele *Sub Assemblies* durch das Assembly vom Fahrzeugmodell angestoßen wurden.

Die erforderliche Konfiguration der Fahrzeug- und *Sub Assembly*-Pools erfolgt durch die KeyUser und kann den Projekten entsprechend den Anforderungen zur Verfügung gestellt werden.

1.7 Vergleichen von Parametergruppen

Es ist jetzt möglich, die Unterschiede zwischen zwei Versionen einer Parametertabelle zu visualisieren. Dafür werden zunächst im History-Graph zwei verschiedene Versionen einer Parametertabelle mit *[STRG]* + Linksklick ausgewählt und das Kontextmenü mit einem Rechtsklick auf einer der ausgewählten Versionen geöffnet. Die Vergleichsansicht kann hier über den Eintrag *Show diff > Parameter Compare & Merge* oder über den Eintrag *Compare versions* geöffnet werden (Abb. 1.4).

#1365

In der Vergleichsansicht werden die Werte, die zwischen beiden Versionen unterschiedlich sind, in der Farbe der jeweiligen Version hervorgehoben (Abb. 1.5). Die Farbzusweisungen können der Legende über der Parametertabelle entnommen werden. Falls beide Versionen der Tabelle einen gemeinsamen Vorgänger besitzen, versucht LoCo eventuell auftretende Konflikte automatisch über einen 3-Wege-Merge aufzulösen.

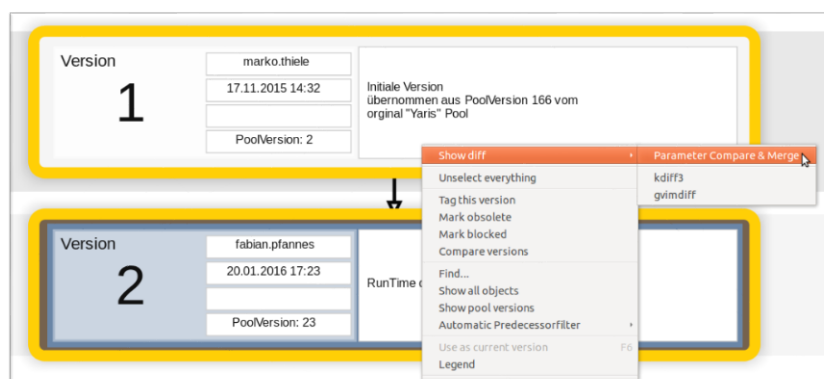
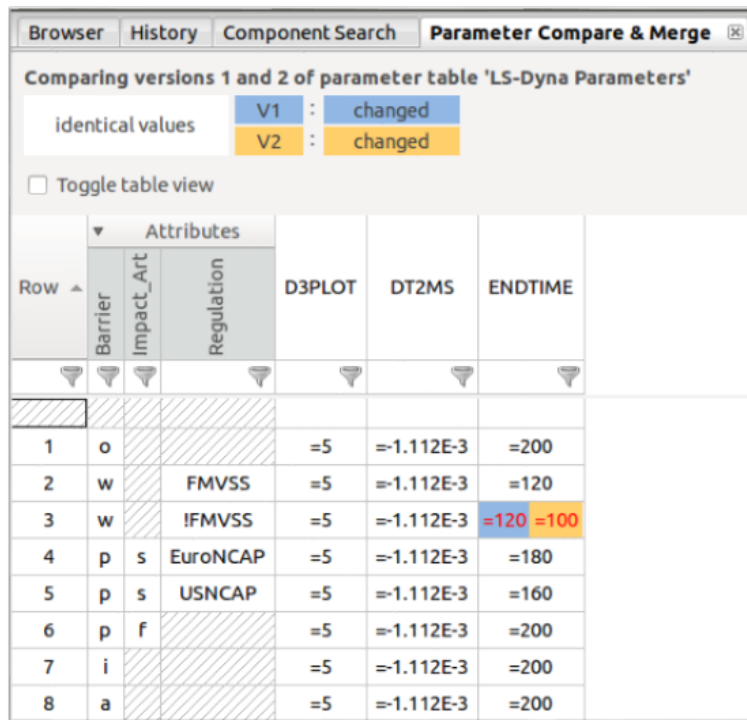


Abb. 1.4: Auswahl von zwei Versionen einer Parametertabelle im History-Graph



| Row | Barrier | Impact_Art | Regulation | D3PLOT | DT2MS | ENDTIME |
|-----|---------|------------|------------|--------|------------|-----------|
| 1 | o | | | =5 | =-1.112E-3 | =200 |
| 2 | w | | FMVSS | =5 | =-1.112E-3 | =120 |
| 3 | w | | !FMVSS | =5 | =-1.112E-3 | =120 =100 |
| 4 | p | s | EuroNCAP | =5 | =-1.112E-3 | =180 |
| 5 | p | s | USNCAP | =5 | =-1.112E-3 | =160 |
| 6 | p | f | | =5 | =-1.112E-3 | =200 |
| 7 | i | | | =5 | =-1.112E-3 | =200 |
| 8 | a | | | =5 | =-1.112E-3 | =200 |

 Abb. 1.5: Parameter *compare and merge*

1.8 Studies sind nun Teil der Pool Version

Studies werden ab sofort in den Pool-Versionen versioniert. Wenn eine Studie in ihrem Umfang (Anzahl der RunDescription), ihrem Namen oder in ihrer Beschreibung geändert wird und diese public ist, wird eine neue Studie erzeugt.

Studies sind also Teil der Pool-Version. Hinzufügen / Editieren und Löschen einer Study führt somit automatisch zu einer neuen Pool-Version.

Bemerkung: Aktuell existierende Studies werden nicht konvertiert, sondern gelöscht.

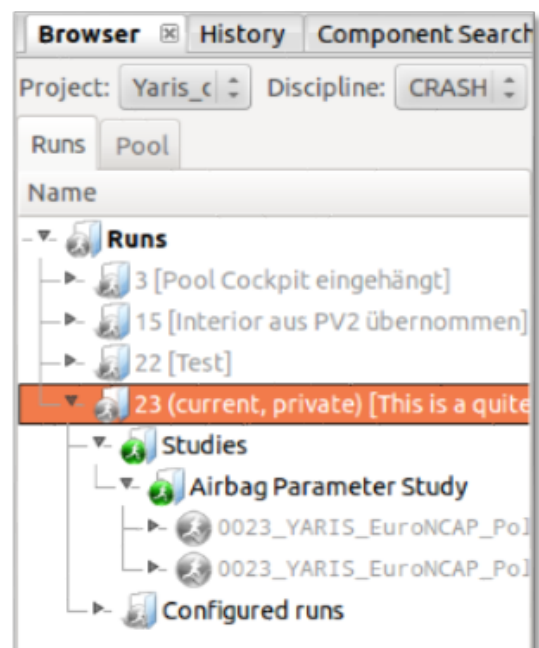

 #2217
 #1890

Abb. 1.6: Pool-Version

1.8.1 Studies

Studies bieten die Möglichkeit, mehrere RunConfigs zusammenzufassen und diese als Grundlage für ein Optimierungsprojekt zu nutzen. An den RunConfigs wird definiert, welche Parameter für die Optimierung verwendet werden und was der gültige Bereich für die Parameter ist.

Hinweis: Im LoCo werden nur die für die Optimierung notwendigen Parameter bereitgestellt. Alle anderen die Optimierung betreffenden Einstellungen müssen in der Optimierungssoftware selbst vorgenommen werden.

1.8.2 Study erstellen

- 1 Im Browser muss ein Knoten ausgewählt, das Kontextmenü geöffnet und auf *Create study* geklickt werden.

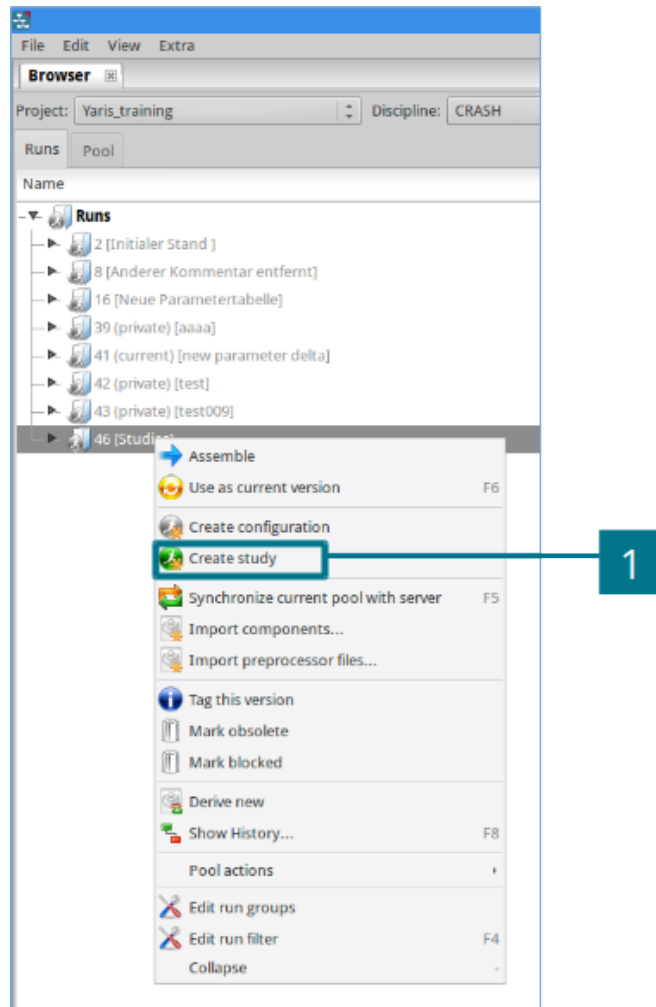


Abb. 1.7: Neue Study erstellen

Eine Study kann immer nur im Kontext einer Pool-Version angelegt werden. Die Study kann aber nachträglich weiter modifiziert werden (siehe [Modifizieren einer Study](#)).

- 1 Es muss ein Name für die Study eingegeben werden.
- 2 Es muss eine Beschreibung für die Study eingegeben werden.
- 3 Hier kann der Pool ausgewählt werden.
- 4 Hier kann die Pool-Version ausgewählt werden, für die die Study angelegt werden soll.

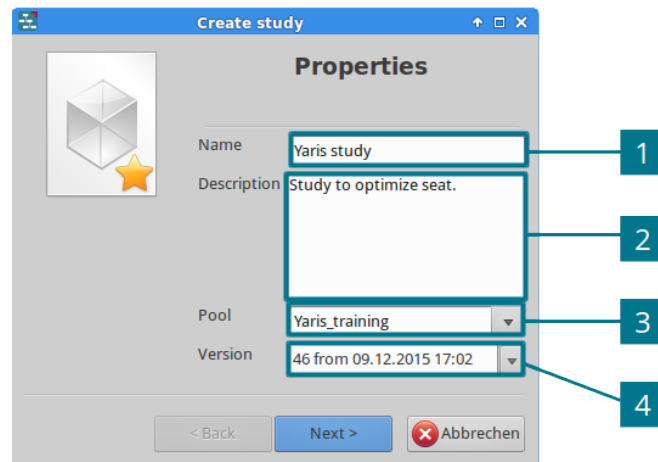


Abb. 1.8: Study-Eigenschaften

Der Study müssen nun noch die RunConfigurations zugewiesen werden. Zur Auswahl stehen dabei alle RunConfigurations der Pool-Version, die im vorherigen Dialog ausgewählt wurde.

- 1 Wählen Sie die RunConfigurations für die Study aus.

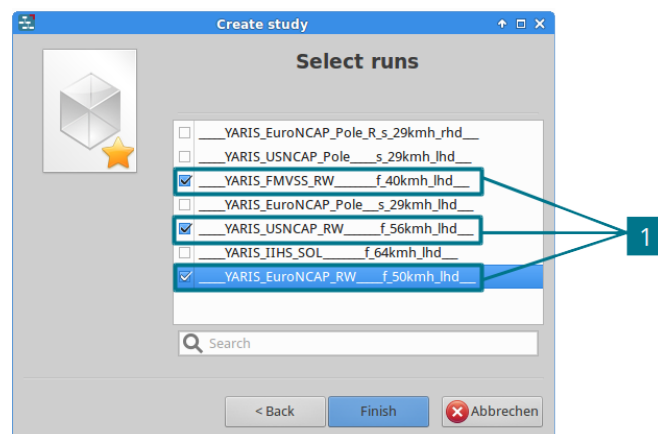


Abb. 1.9: RunConfigurations für Study auswählen

1.8.3 Design Parameter

Design Parameter stellen die für den Optimierer modifizierbaren Teile des Berechnungsmodells dar. *Common Design Parameter* sind Parameter, die global für alle gewählten RunConfigurations verwendbar sind. Ansonsten gelten *Design Parameter* immer nur für die jeweilige RunConfiguration.

Um die *Design Parameter* global als *common Design Parameter* für die Study zur Verfügung zu stellen, müssen diese in der jeweiligen RunConfiguration als *common* markiert werden (siehe Abb. 1.11).

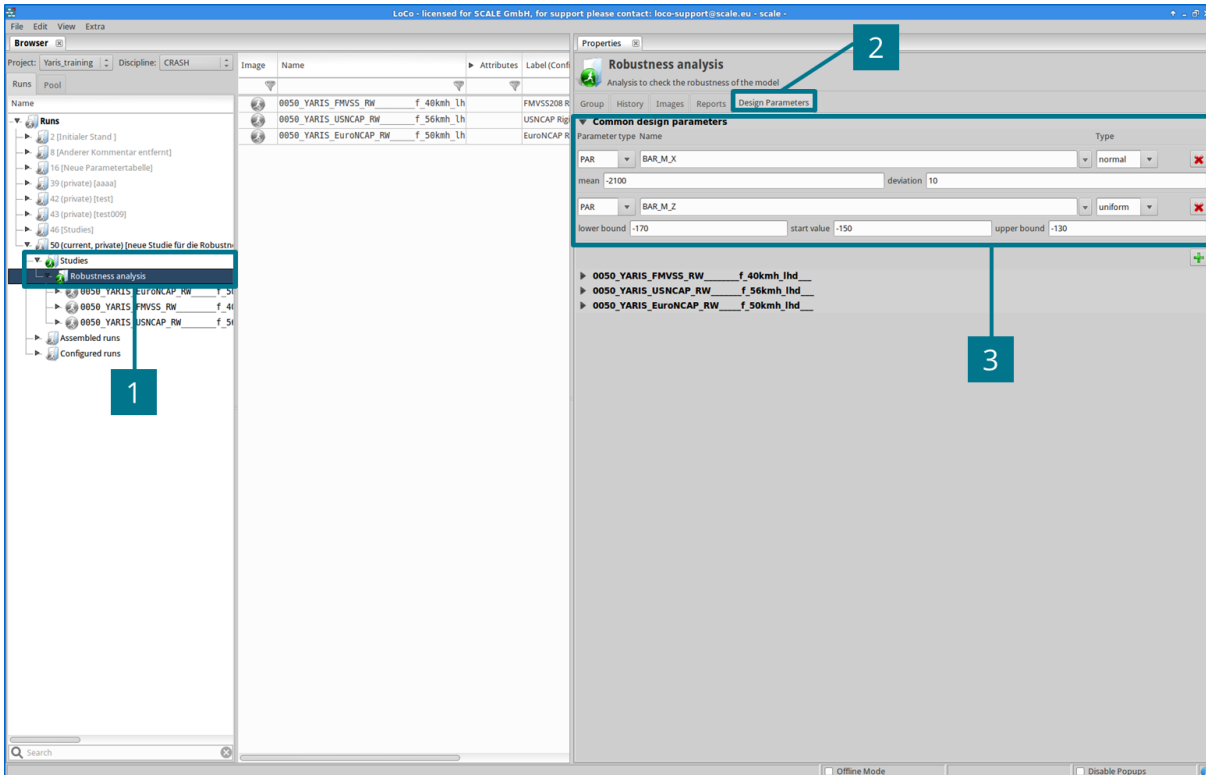


Abb. 1.10: Run Design Parameter

- 1 Studies können im Browser unter *Studies* ausgewählt werden.
- 2 Unter *Properties* muss auf den Reiter *Design Parameters* geklickt werden.

Ein *Design Parameter* kann entweder ein LoCo-Parameter oder ein Element aus dem Berechnungsdeck darstellen. LoCo-Parameter werden über *PAR* angesprochene Elemente aus dem Berechnungsdeck über *BOM*.

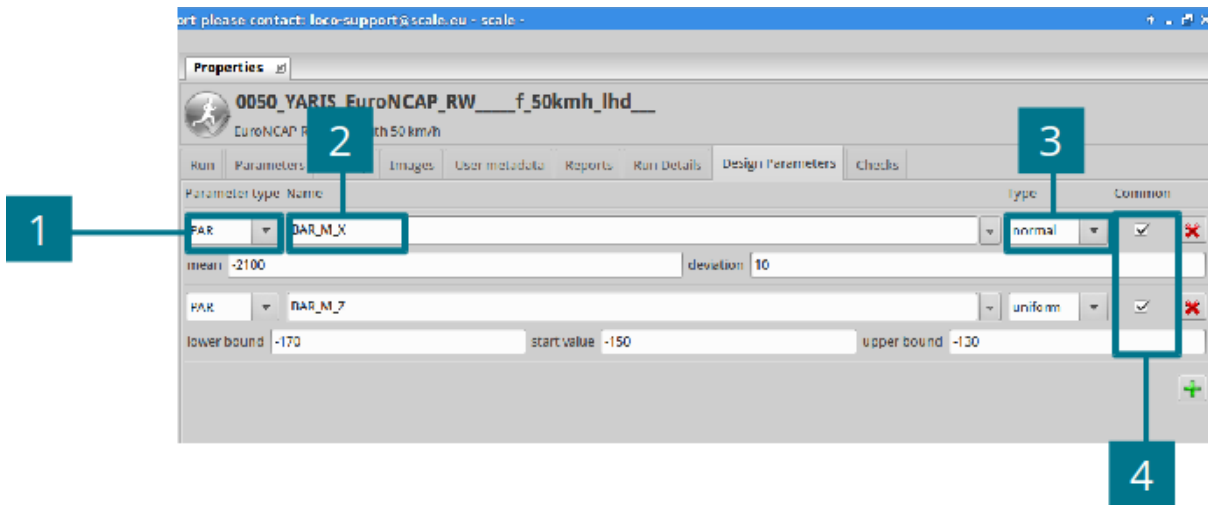


Abb. 1.11: Design Parameter für RunConfiguration

- 1 Quelle des *Design Parameters*: PAR oder BOM
- 2 Name des *Design Parameters*
- 3 Typ des *Design Parameters*
- 4 Definition, ob der *Design Parameter* ein *common Design Parameters* ist

Wenn alle RunConfigurations innerhalb einer Study den selben Parameter besitzen und das Häkchen bei *common* gesetzt ist, lassen sich diese Parameter für alle RunConfigurations der Study auf einmal einstellen (siehe Abb. 1.10).

1.8.4 Modifizieren einer Study

- 1 Im Browser ist unter *Studies* eine Study auszuwählen. Mit der rechten Maustaste ist auf die Study und dann auf *Manage study* zu klicken. Über das nachfolgende Menü lässt sich die Study bearbeiten, duplizieren, löschen und ihre Historie anzeigen.

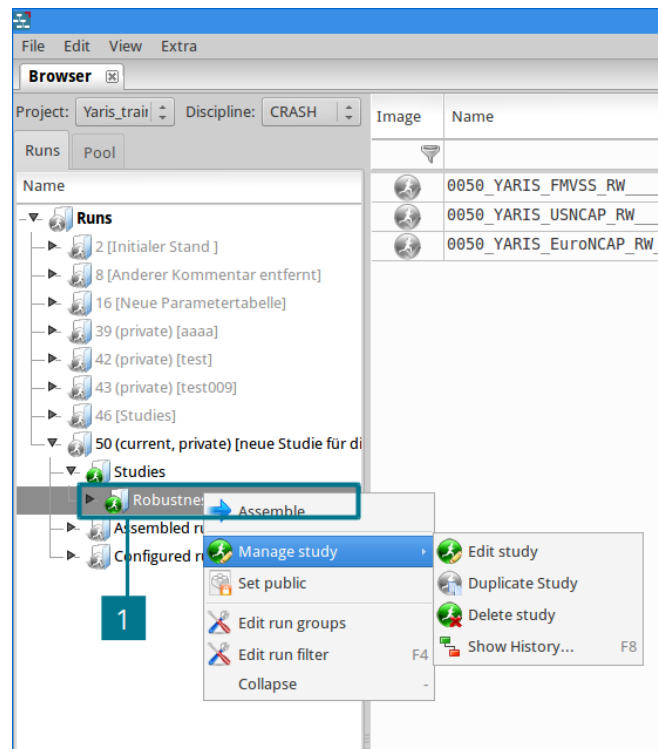
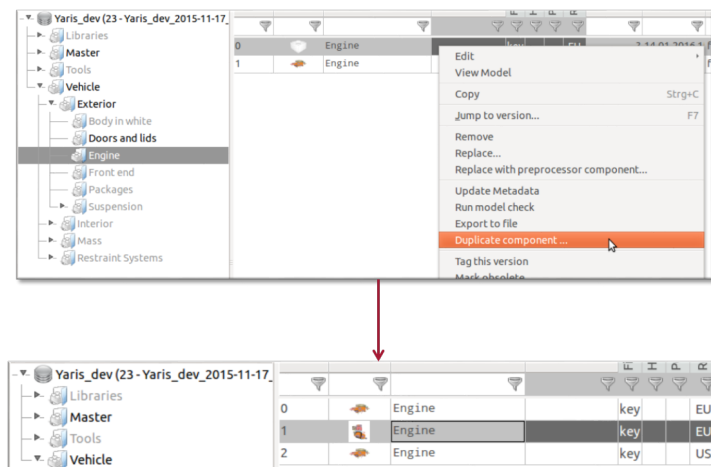


Abb. 1.12: Study bearbeiten

1.9 Komponenten duplizieren

Komponenten können dupliziert werden, um Variationen von bestehenden Includes zu erstellen. Auf diesem Weg können auf einfache Weise im LoCo neue Komponenten angelegt bzw. aus bestehenden Komponenten abgeleitet werden, ohne diese aus dem Filesystem importieren zu müssen.



#1990

Abb. 1.13: Duplizierte Komponenten

1.10 Bestehende Komponente bei Import ersetzen

Beim Importieren einer Komponente wird dem Anwender ab Version 1.42 die Option geboten, eine bestehende Komponente zu ersetzen. Bislang konnte diese nur hinzugefügt werden. Dies ist immer dann möglich, wenn LoCo die Komponente an Hand der im Dateinamen angegebenen Teil-UUID im LoCo findet: #1993

engine_____0023_acbfce22.key

Per default wird angenommen, dass die Komponente ersetzt werden soll.

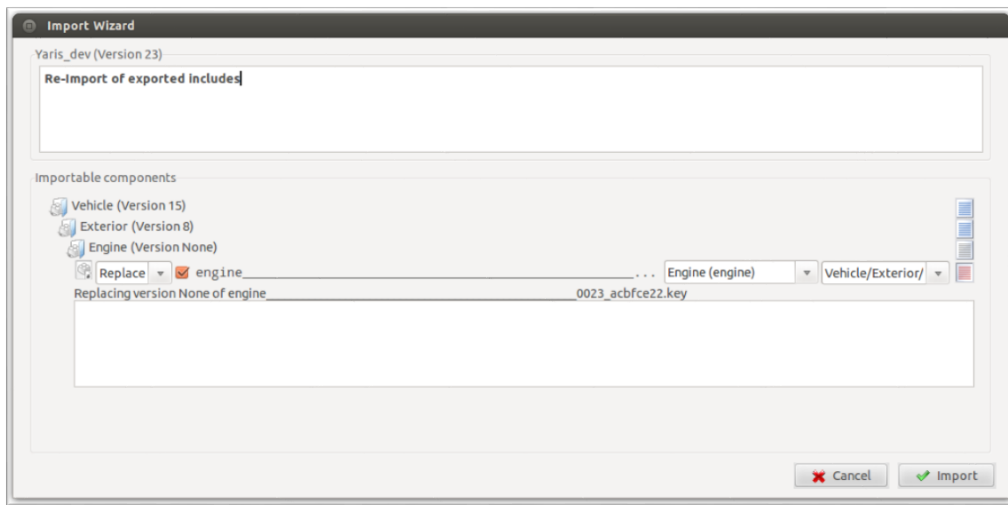


Abb. 1.14: Import Wizard

1.11 Löschen bzw. Deaktivieren von Usern

Nutzer können im Account-Manger deaktiviert werden. Unter *User Information* gibt es die CheckBox *deactivate*. Diese wird erst freigeschaltet, sobald der Nutzer nicht mehr den Status *Enable* hat. Um Nutzer zu deaktivieren, müssen die entsprechenden Rechte vorhanden sein. Um deaktivierte Nutzer anzuzeigen, gibt es neben dem *Reload*-Button die *Show deactivated user*-Checkbox. Diese ist nur sichtbar, wenn die Rechte zum Deaktivieren der Nutzer vorliegen. #171

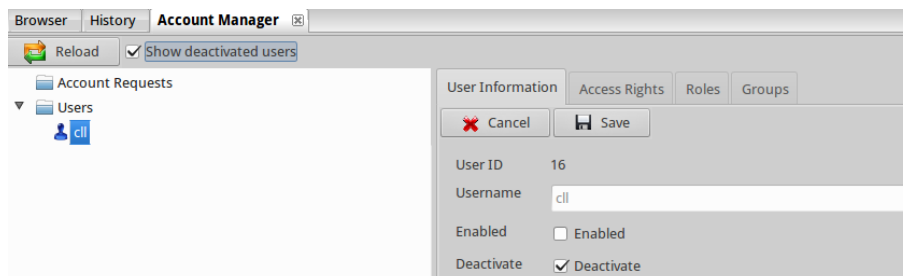


Abb. 1.15: Nutzer deaktivieren

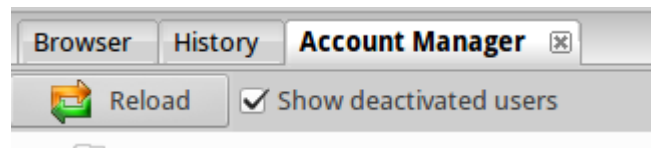


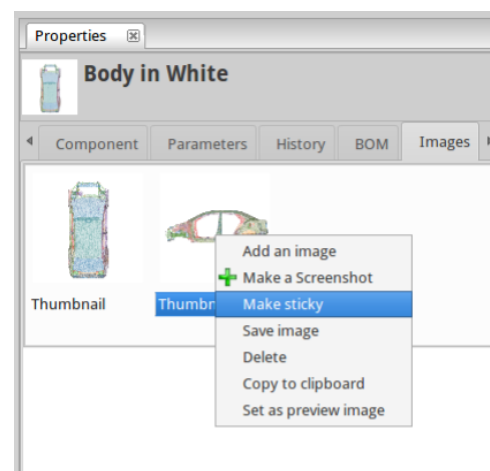
Abb. 1.16: Deaktivierte Nutzer anzeigen

1.12 Reports und Bilder als sticky markieren

Im LoCo ist es möglich, an Komponenten, Komponenten-Gruppen oder Pool-Versionen ergänzende Dokumente anzuhängen.

Dies können Bilder sein, welche bei den Properties im Reiter *Images* abgelegt werden oder beliebige andere Dateien, welche im Reiter *Reports* als Anhänge zu dem Objekt abgelegt werden können.

Ab Version 1.42 kann der Anwender Bilder oder Dateien in den Reitern *Images* oder *Reports* über die Aktion *Make sticky* als permanent markieren. Dadurch werden diese Anhänge in die Nachfolger der entsprechenden Objekte übernommen sobald neue Versionen von diesen abgeleitet werden.



#1887

Abb. 1.17: Reports für Nachfolger übernehmen

1.13 CSV-Export der transponierten Darstellung von Parametertabellen

Parametertabellen können auch in ihrer transponierten Form exportiert werden. Dafür kann der Kontextmenü-Eintrag *Export as csv* verwendet. Allerdings ist **kein** Reimport über *Import from csv* möglich. Ebenso werden keine Änderungen beim Öffnen der transponierten Parametertabelle über das *Edit with*-Kontextmenü übernommen. Dem Nutzer wird in diesem Fall eine Warnung angezeigt, dass die Änderungen nicht reimportiert werden können.

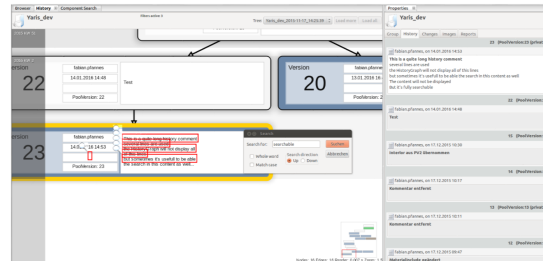
#1738

| 1 | Name | Parameter description | Attribute:Fahrzeugzone | Attribute:Geschwindigkeit | Parameter value |
|---|-------------|-----------------------|------------------------|---------------------------|-----------------|
| 2 | Parameter 1 | Description 1 | h | 20 | Value 1 |
| 3 | Parameter 1 | Description 1 | h | 30 | Value 2 |
| 4 | Parameter 2 | Description 2 | h | 20 | Value 3 |
| 5 | Parameter 2 | Description 2 | h | 30 | Value 4 |

Abb. 1.18: CSV-Repräsentation einer transponierten Parametertabelle

1.14 Suche nach nicht sichtbaren Teilen vom History-Kommentar im History-Graph

Bei einer Suche im History-Graph werden ab sofort die vollständigen Kommentare (History Comments) aller sichtbaren Objekte durchsucht. Dabei wird auch nicht angezeigter Text von sehr langen Kommentaren durchsucht. Bislang wurden nur jene Bestandteile der Kommentare durchsucht, die im History-Graph sichtbar waren.

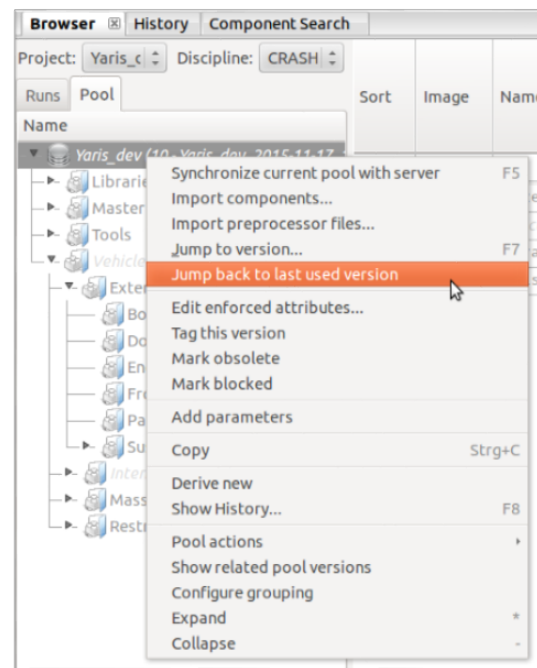


#2369

Abb. 1.19: Suche im History-Graph

1.15 Neuer Kontextmenü-Eintrag *Jump back to last used version*

Auf Pool-Versionen gibt es einen neuen Kontextmenü-Eintrag *Jump back to last used version*. Wird dieser ausgewählt, so wird die aktuelle Version gegen die zuletzt verwendete ausgetauscht. Auf diese Weise kann man schnell zwischen zwei verschiedenen Pool-Versionen wechseln.



#1984

 Abb. 1.20: *Jump back to last used version*

1.16 Auswahl der Namensschema kontextabhängig

Ob ein Namensschema für die Anzeige verfügbar ist, wird durch den Kontext definiert. Im Beispiel ist das Namensschema für die Anzeige verfügbar, wenn die aktuelle Disziplin *CRASH* ist und in der Pool-Konfiguration im `context_filter` unter `namescheme` `Region: ["EU"]` definiert ist. Wenn in der `nameschemes configuration` kein `<context>`-Tag gesetzt ist, wird das Namensschema nicht gefiltert.

#1965

Listing 1.2: Konfiguration des Pools

```

1 <pool-config name="Yaris" uuid="64c99bf8-8124-4c2b-966a" is_subordinate="0">
2   <description>Toyota Yaris</description>
3   <checks-location>Tools/Checks</checks-location>
4   <context_filter>
5     <namespace>Region: ["EU"]</namespace>
6   </context_filter>
7 </pool-config>
  
```

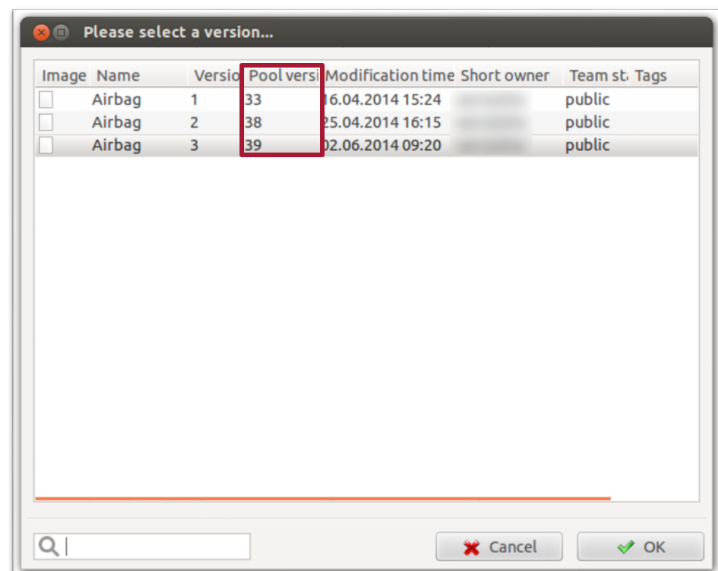
Listing 1.3: Konfiguration eines Namensschemas

```

1 <namespace id="CRASH" usage="run">
2   <context>Region: ["EU"]</context>
3   <attribute-value length="5" aname="Project"/>
4   <fixed value="_"/>
5   <scenario group="Load Case" length="15"/>
6   <fixed value="_"/>
7   <attribute-value length="1" aname="Impact_Art"/>
8   <attribute-value length="2" aname="Velocity"/>
9   <fixed value="_"/>
10  <repository-version length="4"/>
11  <attribute-value maxlength="0" aname="Disziplin"/>
12 </namespace>
  
```

1.17 *Jump to Version* auf einem Include zeigt die Pool-Version an

Im *Jump to Version*-Dialog auf Includes wird dem Nutzer zusätzlich die Pool-Version des Includes angezeigt. Dadurch ist schnell erkennbar, in welcher Pool-Version die jeweilige Version des Includes hinzugekommen ist.



#2422

Abb. 1.21: Versionsdialog

Weitere Verbesserungen

- Die *Job status*-Anzeige besitzt zwei neue Buttons für die Aktualisierung bzw. Bereinigung der Einträge
 - *Clear finished jobs* wird verwendet, um beendete Jobs aus der Liste zu entfernen.
 - *Update status for remote jobs* wird verwendet, um den Job status manuell zu aktualisieren. Das automatische Aktualisierungsintervall ist auf 5 Minuten eingestellt.

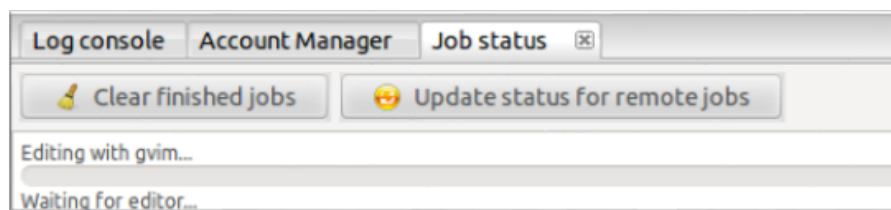


Abb. 2.1: Job status
[#1976]

- In der *Job status*-Anzeige werden Einträge auf die verfügbare Breite gekürzt [#1991]
- Der Vergleich von Komponenten (diff) ist auch für Preprocessor-Dateien möglich [#2131]
- Model-Checker Skripte können Bilder unter *Properties > Reports* an eine Komponente anfügen [#2332]
- *Projectmanager* können neue Trees in Pool-Versionen erstellen und deren Namen editieren [#2569]
- Sind mehrere Komponenten im Grid markiert (z.B. *STRG* + Klick), können sie ab sofort mit einem weiteren Klick wieder deselektiert werden
- Wird ein privater Pool mit *Derive new* abgeleitet, so werden die *HistoryComments* kopiert [#2092]
- Aus dem *History-Graph* können Parametertabellen ebenfalls als CSV exportiert und editiert werden [#2345]
- Lokale und globale Versionsnummern sind ab sofort für die Verwendung im *name scheme* verfügbar. Lokale Versionsnummern entstehen ausschließlich bei Offline-Nutzung von LoCo. [#2617]:
 - `<item-global-version length="3"/>`
 - `<item-local-version length="3"/>`
 - `<repository-global-version length="4"/>`
 - `<repository-local-version length="4"/>`

- Verbesserungen beim automatischen Download (prefetching) von Massendaten mit dem Ziel, zukünftig weniger nicht benötigte Massendaten automatisch zu beziehen
 - *Prefetch BLOBs during synchronizing* - Option ist per default deaktiviert. Damit werden keine Komponenten-Daten mehr *auf Verdacht* heruntergeladen [#2314]
 - Downloadlimit für den initialen Checkout ist per default auf Null gesetzt. Damit werden keine Komponenten-Daten mehr *auf Verdacht* heruntergeladen.
 - Beim Wechsel der Pool-Version werden nur die zur aktuellen Pool-Version gehörenden Daten automatisch heruntergeladen [#2455]
- Mit der Kommandozeilenoption `--enable-context-filtering` kann aktiviert werden, dass in der GUI nur Attribute und Komponenten-Typen angezeigt werden, die für die jeweils gesetzte Disziplin relevant sind. (Damit dies funktioniert, ist eine entsprechende Konfiguration durch die KeyUser erforderlich.) [#2314]
- Der Parameter `POOL_HISTORYCOMMENT_BODY` wurde eingeführt. Dieser enthält den History-Comment ohne die erste Zeile (diese ist verfügbar in `POOL_HISTORYCOMMENT_HEADER`) [#1994]
- Beim Filtern in Parametertabellen bleibt die *Description*-Spalte jetzt immer sichtbar [#1686]
- Die Benutzernamen von LoCo-Nutzern können geändert werden [#967]
- LoCo wird zusätzlich für die Linuxderivate Ubuntu 14.04 sowie OpenSuse 13.1 bereitgestellt [#2519], [#2392]

Bugfixes

- MultiRunConfigs:
 - Als Diskriminator wird nicht – gesetzt, solange eine RunConfiguration editiert wird [#2622]
 - Kinder-RunConfigurations werden auf Basis der entsprechenden Pool-Version und nicht auf Grundlage der aktuellen bestimmt [#2163]
- Beim Aufruf von *Edit* auf Komponenten: Gelegentliches *Hängen* von Editoren im Job status tritt nun nicht mehr auf [#1528]
- Veränderungen an Snippets gehen nicht mehr verloren, wenn zeitgleich ein Sync durchgeführt wird [#2571]
- Das Tabellenlayout von Parametertabellen wird ohne Verzögerung gespeichert. LoCo merkt sich ein- und ausgeblendete Spalten. [#2644]
- Includes werden beim Import nicht mehr als *raw header include* importiert [#2354]
- LoCo konnte einfrieren, wenn der History-Tab (in der Properties-View) zu schmal war [#2593]
- Es ist wieder möglich, Komponenten direkt in das Pool-Wurzelverzeichnis zu importieren [#2751]
- Wird eine einzelne Datei mit ANSA editiert, so werden neue Entities nicht mehr verworfen. Die komplette ANSA-DB wird als Include exportiert. [#2583]
- Ein Lpack-Import überschreibt die Pool-Konfiguration nicht mehr [#2533]
- Beim Checkout eines Pools auftretender ECONNRESET Fehler wurde behoben [#2531]
- Im RunStudy-Tree ist eine Filterung nach *Username* möglich [#2299]
- Ein Fehler beim Hochladen von Blobs konnte behoben werden (*Can not upload remote BLOB*) [#2192]
- Die Eigenschaften von im RunStudy-Tree selektierten Parametertabellen sind nun in der Properties-View sichtbar [#2132]
- Das Ergebnis einer *Compare & Merges*-Aktion wird als *current version* gesetzt [#2035]
- Tree Namen werden nach einer Änderung korrekt zum Server synchronisiert [#2013]
- Beim *Mount* eines neuen Pools sind alle Trees im Versionsdialog verfügbar [#1941]
- *Show history* auf Komponentengruppen funktioniert wieder im RunStudy-Tree [#2544]
- Probleme mit SOAP Remote Submit über WebSeal wurden behoben [#2158]
- Probleme mit Remote Submit bei externen Anwendern wurden behoben [#2048]
- Copy and Paste bei Komponenten mit zwei Association Row's funktioniert [#2020]

- Sortierung der Versionen im *Jump to Version*-Dialog erfolgt wieder numerisch (100 nicht mehr vor der 20) [#2418]