

LoCo Release Notes

Release 1.67

info@scale.eu www.scale.eu

SCALE 

IT-Solutions for CAE

1	Neue Funktionen	1
1.1	Widget Presets	1
1.2	Prozess-Feedback	5
1.3	Staged Tree	6
1.4	Kontext Filter	8
2	Verbesserungen	10
3	Bugfixes	12

Neue Funktionen

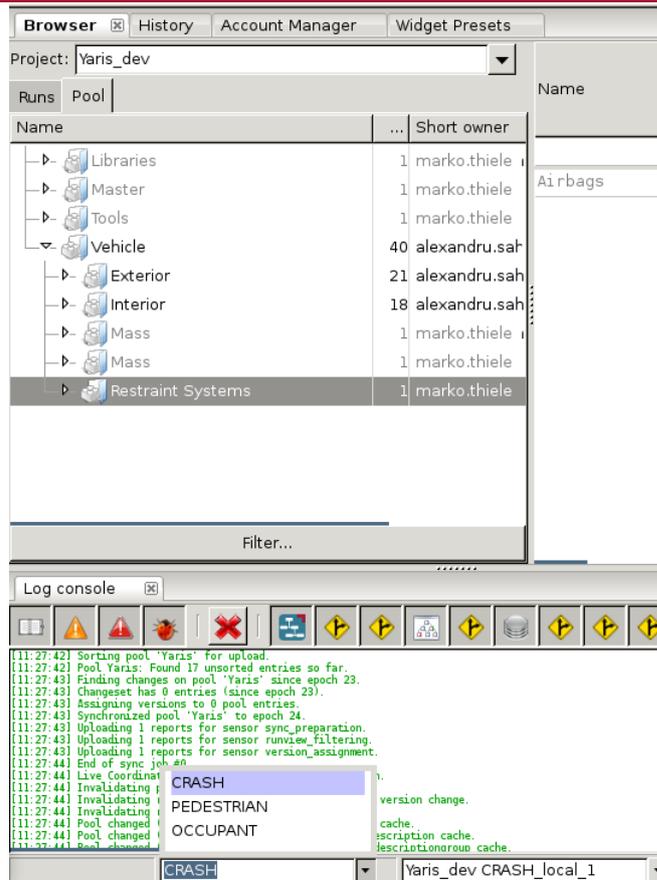
1.1 Widget Presets

Dieses Feature vereinfacht die Verwaltung und das Speichern der GUI-Einstellungen (WidgetPresets). Die Schwerpunkte sind:

- Verwaltung der GUI-Einstellungen an einer Stelle im LoCo
- Speichern der Presets auf dem Server
- Disziplin- und projektabhängige Definitionen von Standard-Presets durch die KeyUsers

Hinweis: Dieses Feature befindet sich noch im Beta-Stadium. Es muss mit der Kommandozeilenoption `--enable-new-widget-presets` aktiviert werden.

Die Auswahlliste der Disziplin wird nicht mehr im oberen Bereich neben der Projekt-Auswahlliste, sondern in der unteren Menüleiste angezeigt.



The screenshot shows the LoCo interface with the following components:

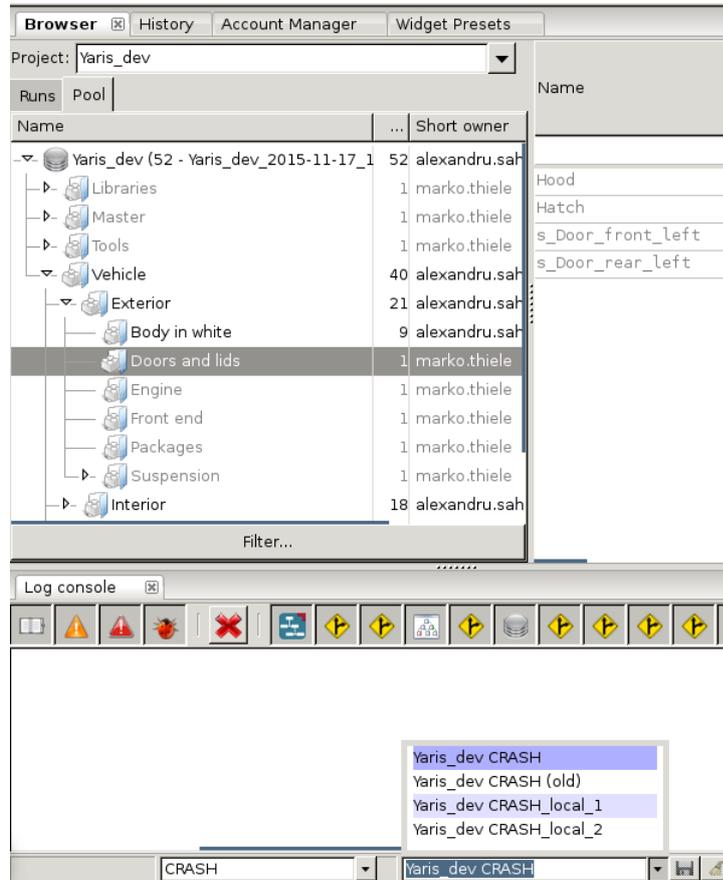
- Browser:** A file browser window showing a project named 'Yaris_dev'. The 'Pool' tab is active, displaying a tree view of folders and files. The 'Vehicle' folder is expanded, showing sub-folders like 'Exterior', 'Interior', 'Mass', and 'Restraint Systems'. A table below the tree lists items with their counts and owners.
- Log console:** A window showing system logs. A context menu is open over the log text, listing 'CRASH', 'PEDESTRIAN', and 'OCCUPANT'. The 'CRASH' option is selected, and a dropdown menu shows 'Yaris_dev CRASH_local_1'.

Name	Count	Short owner
Libraries	1	marko.thiele
Master	1	marko.thiele
Tools	1	marko.thiele
Vehicle	40	alexandru.sah
Exterior	21	alexandru.sah
Interior	18	alexandru.sah
Mass	1	marko.thiele
Mass	1	marko.thiele
Restraint Systems	1	marko.thiele

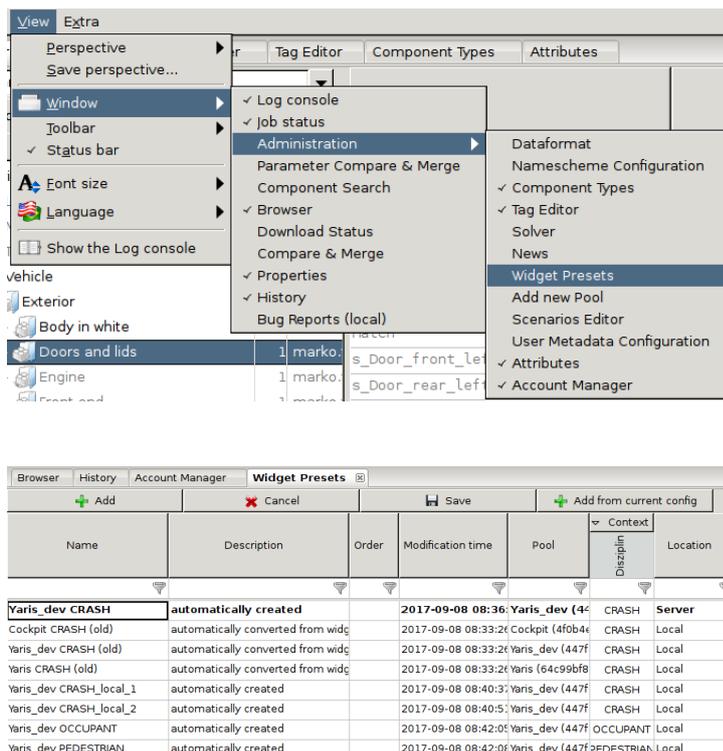
```

[11:27:42] Sorting pool 'Yaris' for upload.
[11:27:42] Pool Yaris: Found 17 unsorted entries so far.
[11:27:43] Finding changes on pool 'Yaris' since epoch 23.
[11:27:43] Changeset has 0 entries (since epoch 23).
[11:27:43] Assigning versions to 0 pool entries.
[11:27:43] Synchronized pool 'Yaris' to epoch 24.
[11:27:43] Uploading 1 reports for sensor sync_preparation.
[11:27:43] Uploading 1 reports for sensor runview_filtering.
[11:27:43] Uploading 1 reports for sensor version_assignment.
[11:27:44] End of sync job.
[11:27:44] Live Coordinates:
[11:27:44] Invalidating j...
[11:27:44] Invalidating j...
[11:27:44] Invalidating j...
[11:27:44] Pool changed
[11:27:44] Pool changed
[11:27:44] Pool changed
    
```

Daneben wird eine Liste der verfügbaren WidgetPreset angezeigt.



Über *View > Window > Administration > WidgetPresets* kann der WidgetPresets-Editor geöffnet werden:



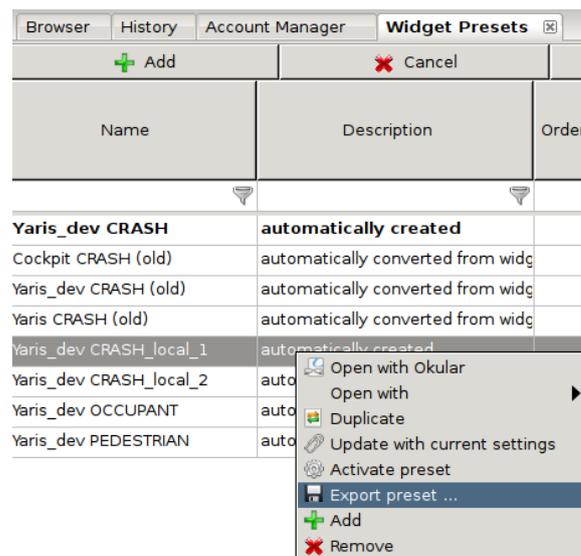
Hier werden alle vorhandenen WidgetPresets gelistet. Es kann zwischen zwei WidgetPreset-Typen unterschieden werden: *Server* und *Local*, die auf dem Server bzw. in der lokalen LoCo-Instanz gespeichert sind. Auf dem Server dürfen nur User mit der Rolle *KeyUser* WidgetPresets speichern und editieren. Alle anderen LoCo-Anwender können ein auf dem Server gespeichertes WidgetPreset verwenden oder eigene lokale Versionen davon ableiten. Der Anwender kann die GUI-Elemente (angezeigte Spalten, Filter, ...) nach seinen Anforderungen einstellen. Bei Änderungen und Auswahl des "Server"-WidgetPreset wird automatisch ein neues lokales WidgetPreset erstellt, das den gleichen Namen wie das *Server*-Preset trägt, aber mit einem * vor dem Namen gekennzeichnet wird.

Wird ein WidgetPreset *Yaris_dev CRASH* vom Server geladen und eine Änderung an einem Filter vorgenommen, erzeugt LoCo automatisch eine Version *Yaris_dev CRASH*.

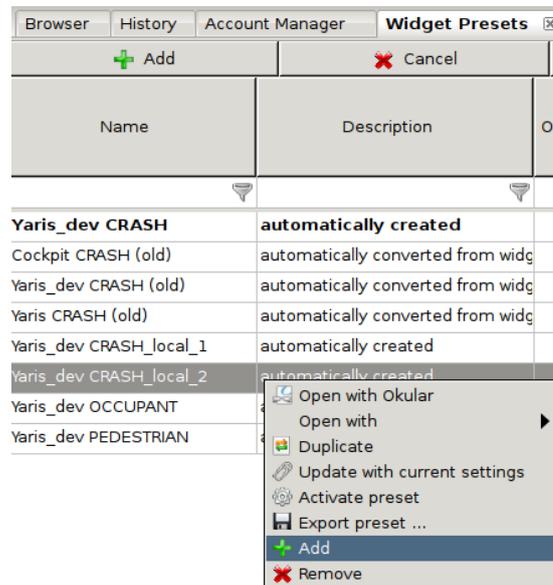


Der *Save*-Button wird in diesem Fall aktiviert und bietet die Möglichkeit, das neu erzeugte WidgetPreset als *Local*, d.h. lokale Kopie zu speichern. Wenn ein lokales WidgetPreset gewählt ist und GUI-Einstellungen verändert wurden, wird automatisch das gewählte WidgetPreset überschrieben.

Darüber hinaus gibt es die Möglichkeit, ein WidgetPreset als eine .xml-Datei zu speichern und in eine andere LoCo-Instanz als WidgetPreset zu importieren.



Anwendern mehrerer LoCo-Instanzen, die keine Berechtigung zur Speicherung von WidgetPresets auf dem Server haben, ist es möglich, die WidgetPresets instanzübergreifend zu nutzen. Das in einer Instanz generierte und auf Festplatte exportierte WidgetPreset, kann in einer anderen LoCo-Instanz über Rechtsklick im WidgetPresets-Editor und *Add* auf einem früher erzeugten lokalen WidgetPreset importiert werden.



1.2 Prozess-Feedback

Mit dem neuen Prozess-Feedback-Feature wird der Fortschritt eines Jobs in der *Jobs*-View wesentlich detaillierter angezeigt.

Hinweis: Dieses Feature befindet sich im Beta-Stadium. Es muss mit der LoCo-Startoption `--enable-unstable-features new_job_view` beim Start aktiviert werden.

Hinweis: Das Job-Status-Reiter kann unter *View > Window > Jobs* aktiviert werden.

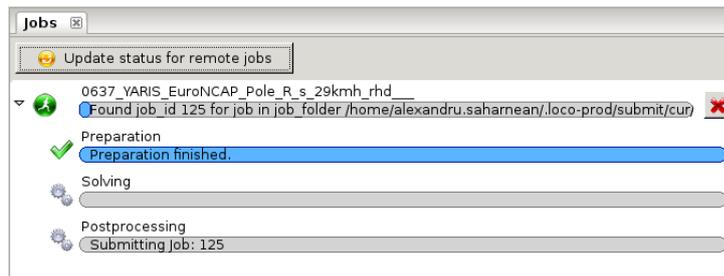
Die Vorteile des Prozess-Feedbacks sind:

- Alle zu einem Job gehörenden *Kind-Jobs* werden als Unterknoten des Jobs angezeigt.
- Sowohl der Hauptjob als auch die Kind-Jobs werden mit eigenen Fortschrittsbalken und kurzen Nachrichten dargestellt.
- Sowohl der Hauptjob als auch die Kind-Jobs werden mit kleinen Icons dargestellt, die den Job-Status (läuft/Error/Warning/Normal) anzeigen

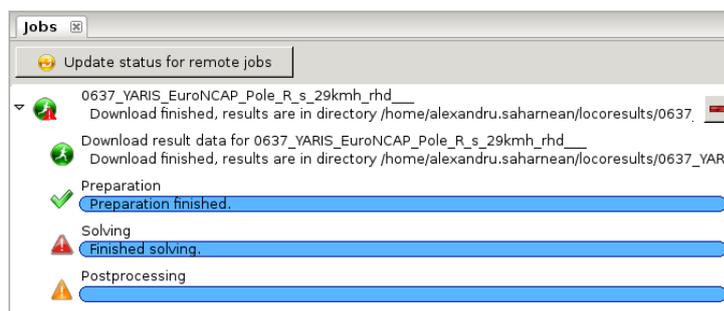
Zum Beispiel kann ein *RemoteSubmit-Job* in drei *Kind-Jobs* eingeteilt werden:

- Vorbereitung (Preparation)
- FEM-Rechnung (Solving)
- Postprozessing

In diesem Fall werden die drei Kind-Jobs als untergeordnete Knoten des Hauptjobs angezeigt:



Jeder Knoten besitzt also seinen eigenen Fortschrittsbalken, Nachricht- und Status- bzw. Erfolg-Icon.



Der Fortschrittsbalken des Hauptjobs ist abhängig vom Fortschritt der Kind-Jobs. Welchen Anteil am Fortschritt jeder Kindjob am Hauptjob hat, ist in den Skripten definierbar.

Hinweis: Für RemoteSubmit-Jobs benötigt LoCo diverse Informationen (Gewichtung, Fortschritt, ...), um die Knotenstruktur usw. eines Kind-Jobs anzuzeigen. Die Skripte/Clusterskripte müssen diese Information in einer bestimmten Datei in einem bestimmten Format ablegen. Für weitere Informationen steht der LoCo-Support zur Verfügung.

1.3 Staged Tree

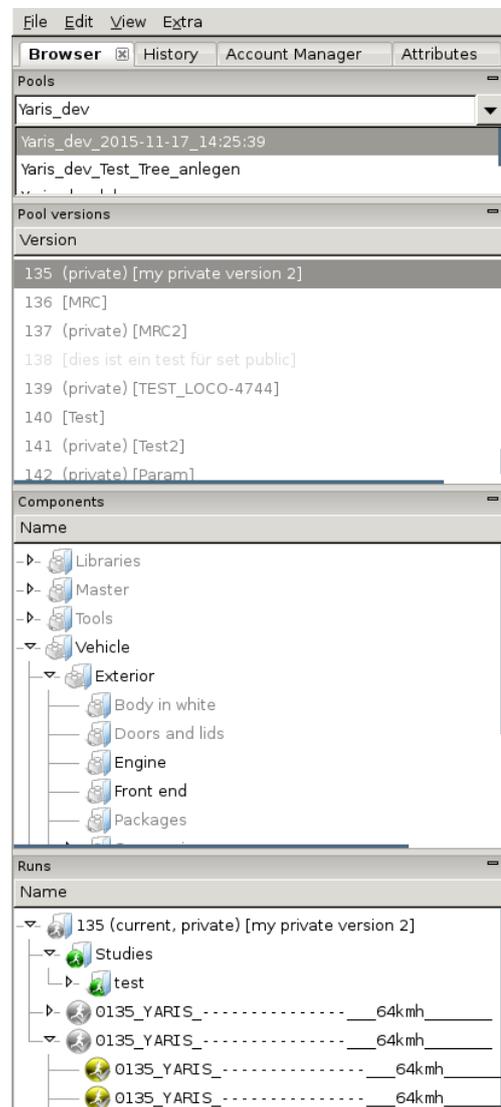
Das Staged Tree Feature bietet ein alternatives Bedienkonzept für LoCo an. In Zukunft wird dieses Bedienkonzept die aktuelle Navigation über PoolTree und RunStudyTree ersetzen. Ziel ist es, eine intuitivere und schnellere Navigation zu ermöglichen. Dazu wird die Navigation auf mehrere Ebenen (Stages) aufgeteilt. Der Anwender kann in jeder Ebene Selektionen vornehmen, welche, soweit dies möglich ist, auch dann erhalten bleiben, wenn er in anderen Ebenen etwas anderes selektiert. So bleibt beispielsweise die Selektion auf den Runs erhalten, wenn der Anwender die PoolVersion wechselt, vorausgesetzt, in der neu gewählten PoolVersion existiert der gleiche Run (bzw. eine passende RunConfig). Auf diese Weise kann der Anwender wesentlich schneller zwischen unterschiedlichen PoolVersionen wechseln, ohne den Fokus auf die aktuell relevanten Runs zu verlieren.

Hinweis: Dieses Feature befindet sich noch im Beta-Stadium. Es muss mit der Kommandozeilenoption `--tree-browser-type=staged` aktiviert werden.

Nach Aktivierung werden im "Browser" View nicht mehr die zwei Reiter *Pool* und *Runs* angezeigt. Stattdessen werden im Stage Tree dem Anwender von oben nach unten folgende Selektionen angeboten:

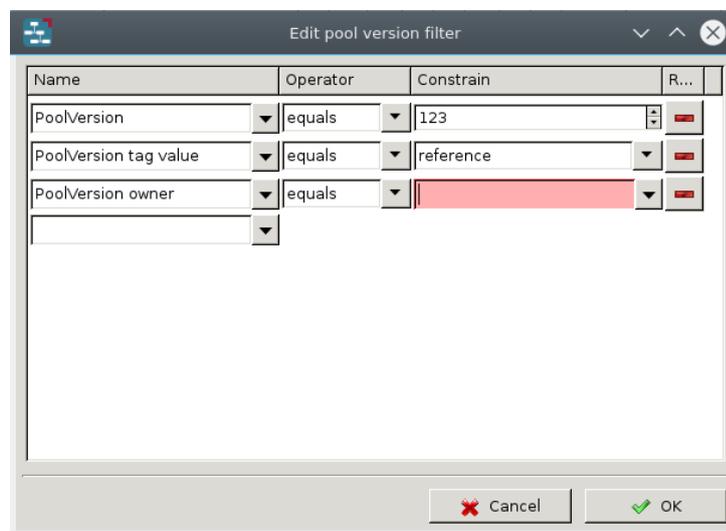
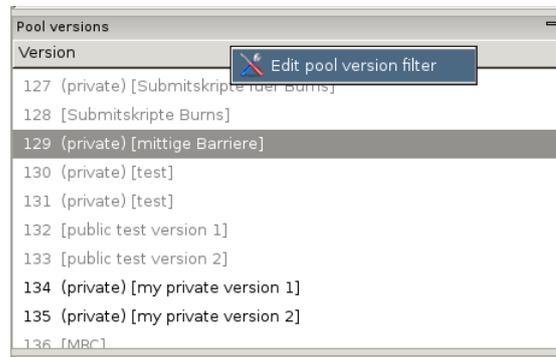
- Pool und Baum/Tree
- Pool Version
- Pool Gruppe/Komponenten

- Run/RunConfig(s)



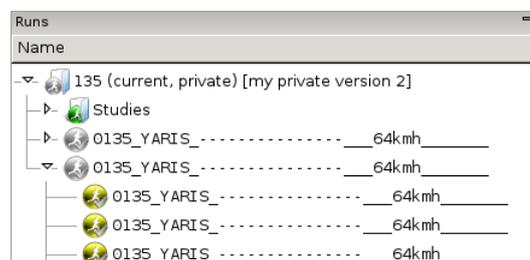
Auf der *Pool versions* Ebene sind aktuell bereits folgende Filter verfügbar:

- Pool Version
- Pool Version tag value
- Pool Version owner



Die Filter werden bei aktivierter "Staged Tree" Navigation für die Bereiche *Pool Versions* und *Runs* jeweils separat eingestellt. Die Filter welche im *Pool Versions* eingestellt werden können, beziehen sich nur auf die PoolVersionen. Die bisherigen Filter werden auf der *Runs* Ebene weiter verwendet und beziehen sich immer nur auf die Runs, welche zur aktuell gewählten PoolVersion gehören.

Im *Runs* werden ab sofort, die aufgebauten RunConfigs (Runs) als untergeordnete Knoten der RunConfigs angezeigt. Da ein aufgebauter Run immer zu einer bestimmten RunConfig gehört, wird somit sichergestellt, dass dem Anwender dieser Zusammenhang bewusst wird.



1.4 Kontext Filter

Über die Kontext Filterung kann die Konfiguration des LoCo-Servers so eingerichtet werden, dass beispielsweise Attribute, Komponententypen, Namensschemas, usw. nur dann sichtbar werden, wenn sie

für den Kontext, in dem der Anwender sich bewegt, auch relevant sind. So kann z.B. ein Attribut so definiert werden, dass es nur für bestimmte Disziplinen sichtbar wird. Oder ein Pool kann für einen bestimmten OEM vorgesehen werden und es wird verhindert, dass Attribute und Komponententypen sichtbar werden, die ausschließlich von anderen OEMs verwendet werden.

Auf diese Weise soll sichergestellt werden, dass der Anwender in Zukunft nicht mehr mit so vielen Attributen und Komponententypen konfrontiert wird. Stattdessen sollen nach Möglichkeit nur noch die Elemente sichtbar sein, welche für den jeweiligen Anwender auch relevant sind. Dies soll die Komplexität für den einzelnen Anwender reduzieren.

Ab der LoCo Version 1.67.11 werden die Kontext Filter per default aktiviert.

Hinweis: Dieses Feature hängt in hohem Maße davon ab, wie gut es durch die KeyUser konfiguriert worden ist. Sollten Sie Attribute, Komponententypen, Namensschemas oder ähnliches vermissen, so melden Sie dies bitte ihrem Ansprechpartner. Im Zweifelsfall können sie das Feature beim Start über die `--disable-context-filtering` deaktivieren.

Verbesserungen

- [#4811] - Erhöhung der SQLite `page_size` zur Performancesteigerung

Die verwendete Datenbank-Software SQLite ermöglicht die Einstellbarkeit der sog. `page_size`. Bei SCALE wurden Tests für eine optimale Größe der `page_size` durchgeführt. Der ermittelte Wert wird nun beim erstmaligen Start dieser LoCo-Version verwendet. Diese tiefgreifende Änderung in der Datenbank kann in Abhängigkeit von der Datenbank-Größe einige Zeit in Anspruch nehmen. Die Änderung der `page_size` sollte eine Performance-Verbesserung bei größeren Pools bewirken.

- [#4737] - Prozess-Feedback: Status der Kinder-Knoten an übergeordnete Knoten propagieren

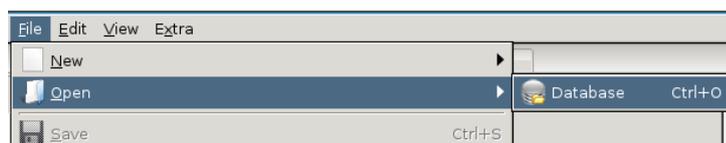
Das Prozess-Feedback zeigt nun das schlechteste Ergebnis der Unterknoten im übergeordneten Knoten an. Damit wird sichtbar, ob es Probleme in einem Unterprozess gegeben hat.

- [#4818] - Prüfen der Betriebssystem-Version beim Starten von LoCo

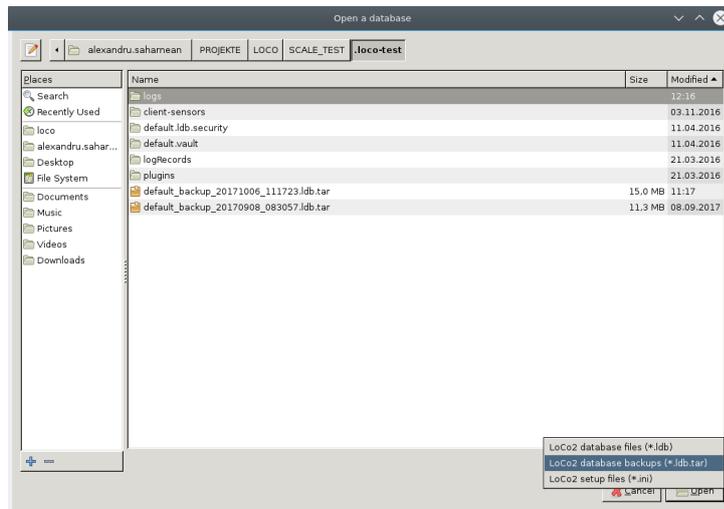
Die LoCo Executable prüft nun, ob die Betriebssystem-Version mit der Version übereinstimmt, für die LoCo gebaut wurde. Diese ist insb. für Minor-Releases bei den Linux-Distributionen relevant.

- [#4804] - Abrufen der BugReports durch ServiceADMs wieder ermöglichen
- [#4686] - Compare&Merge für mehr als zwei PoolVersionen ermöglichen
- [#4552] - Datenbank-Backup als `.tar`-Datei anlegen

Die neuen LoCo-Versionen werden bei den Datenbank Upgrades, Datenbank Backups als `.tar`-Dateien erstellen, sofern dies vom User nicht explizit abgelehnt wird. In der `.tar`-Datei wird LoCo nicht nur die alte Datenbank, sondern auch andere notwendige Dateien wie z.B. `default.ldb.key.gpg` bei einer verschlüsselten Datenbank miteinpacken. Die erzeugte `.tar`-Backup-Datei hat ein Namenformat `default_backup_DATUM_ZEIT.ldb.tar`. Wenn es nötig wird, eine Backup-Datei in LoCo zu öffnen -etwa wenn das Upgrade fehlgeschlagen ist -, kann es über *File > Open > Database* und dem *Open a database*-Dialog geöffnet werden:



In dem Dialog unten rechts kann ein Dateienformat selektiert werden:



Wenn das Format `LoCo2 database backups` selektiert wird, und zum Ordner, in dem das Backup erstellt wurde, navigiert wird, wird der User die Backup-.tar-Datei direkt in LoCo öffnen können. Es ist *nicht* notwendig, die .tar-Datei selbst zu entpacken und in LoCo nur die `default.ldb` einzulesen, oder die Backup-Dateien umzubenennen.

- [#4538] - Verbesserung der Lesbarkeit von Schrift für Menschen mit Rot-Grün-Schwäche
- [#4504] - Aktivierung der fortgeschrittenen Kompression im Vault für alle Brandings
- [#4437] - Entfernung der Warnung, das die Zeilenlänge bei Parameterersetzung länger als 80 Zeichen wird
- [#4408] - Bereitstellung des Filters für die PoolVersionen um *die letzten [n] PoolVersionen vor current* anzuzeigen
- [#4375] - Verfügbarmachung der in einer PoolVersion vorhandenen Dependencies wie *provides*, *requires* und *run_after* per Dropdown
- [#4348] - Berücksichtigung der Zeilen-Sortierung beim Exportieren von Parametertabellen als .csv-Datei
- [#4104] - Aufruf der `guilabel:News` für den Anwender besser auffindbar machen

Die News sind jetzt unter `guilabel:Extra > News` aufrufbar.

- [#4621] - Automatischer Import von Bildern nach Schließen eines Editors

Werden beim Bearbeiten einer Komponente Bilder im Arbeitsverzeichnis abgelegt, so liest LoCo diese nach dem Schließen des Editors ein und speichert sie an der Komponente. Die Bilder sind sichtbar, wenn die Komponente angewählt wird und unter *Properties* der Reiter *Images* ausgewählt wird. Dieses Feature muss beim Start von LoCo über die Kommandozeile aktiviert werden: `-enable-feature import_images_after_edit`

- [#4983] - Warnung anzeigen, wenn es keine aktiven Runs für den Assembly-Dialog gibt

Wenn alle Runs in einer Gruppe deaktiviert sind, bekommt der Anwender ein Warnungsfenster beim Aufruf des Assembly-Dialogs.

- [#4807] - ANSA Start abändern

Bereitgestellte Buttons der ANSA_TRANSL werden auch in ANSA geladen.

- [#3377] - Nachricht "evaluation of parameter x did not yield any result" nicht mehr loggen

Bugfixes

- [#5034] - Editieren einer Komponente führt zur Aktualisierung derselben, obwohl der Blob nicht verändert wurde
- [#5027] - Bearbeiten einer public Komponente im HG funktioniert nach dem *use as current* nicht
- [#5004] - LoCo Masterkey nur sicher neu schreiben
- [#4978] - Assembled Runs unter MultiRunConfigs nicht mehr aufgelistet
- [#4888] - Abfrage zur Erstellung von neuen Poolversionen erscheint doppelt
- [#4886] - BLOBNotObtainedError bei Diff aus HistoryGraph/C&M
- [#4848] - DeriveNew auf eingehängtem Pool wirft TB *Can't observe pool version until all modifications on pool version are finished*
- [#4843] - LoCo friert beim Import von mehr als 10 Komponenten ein
- [#4827] - Parametertabelle bei Leerzeichen im Parameternamen nicht mehr benutzbar
- [#4809] - LoCo kann abstürzen, wenn mehr als ein LoCo-Prozess in die selbe Log-Datei geschrieben wird
- [#4785] - LoCo friert, beim Versuch eine getoggelte Parameterliste anzuzeigen, unter Windwos ein
- [#4777] - RunStudyTree ist nach Ableiten einer neuen PoolVersion leer, wenn *before current*-Filter gesetzt ist
- [#4722] - Grid beim Setzen von Attributen extrem langsam -> bis zum Fehler recursion depth
- [#4685] - Deaktivierte RunConfigs werden nach Bearbeitung der Mutter-MultiRunConfig aktiviert
- [#4662] - LoCo submitted nicht die selektierten Runs, sondern nur den Run, der sich beim Auslösen des Submits unter der Maus befindet
- [#4651] - Unwirksamen Run Filter *Description* im RunStudyTree entfernen
- [#4641] - Änderungskommentarabfrage verhindert Löschen mehrerer Zeilen in Parametertabelle
- [#4578] - Einfügen einer ShortDescription per Copy&Paste in Komponenten mit mehreren Assoziationen führt zu Änderungen an mehreren weiteren Komponenten
- [#4577] - Refresh-Problem in neuer JobView bei Änderung der Größe
- [#4570] - Zeitintervall für *unattended cleanups* im Storage Janitor wird falsch gespeichert
- [#4566] - Falsches Encoding im Submitskript führt zu Abbruch beim SOAP RemoteSubmit
- [#4483] - Replace von private Komponente in HG schlägt fehl
- [#4284] - HistoryComment wird nicht abgefragt

- [#4239] - Kopieren von Parameterzeilen funktioniert nicht
- [#4164] - Account Manager: Filter nach User funktioniert nicht bei *automated requests*
- [#4110] - Beim Ändern einer Parametertabelle erscheint keine Aufforderung zur Eingabe eines History Kommentars
- [#4081] - *Replace with preprocessor component*: LoCo fragt zweimal nach History-Kommentar
- [#3800] - Copy&Paste von Snippets möglich, die einem Komponenten-Typ nicht zugewiesen wurden
- [#3057] - Bei der Verwendung von Pfeiltasten für die Navigation im Grid werden Properties nicht aktualisiert