

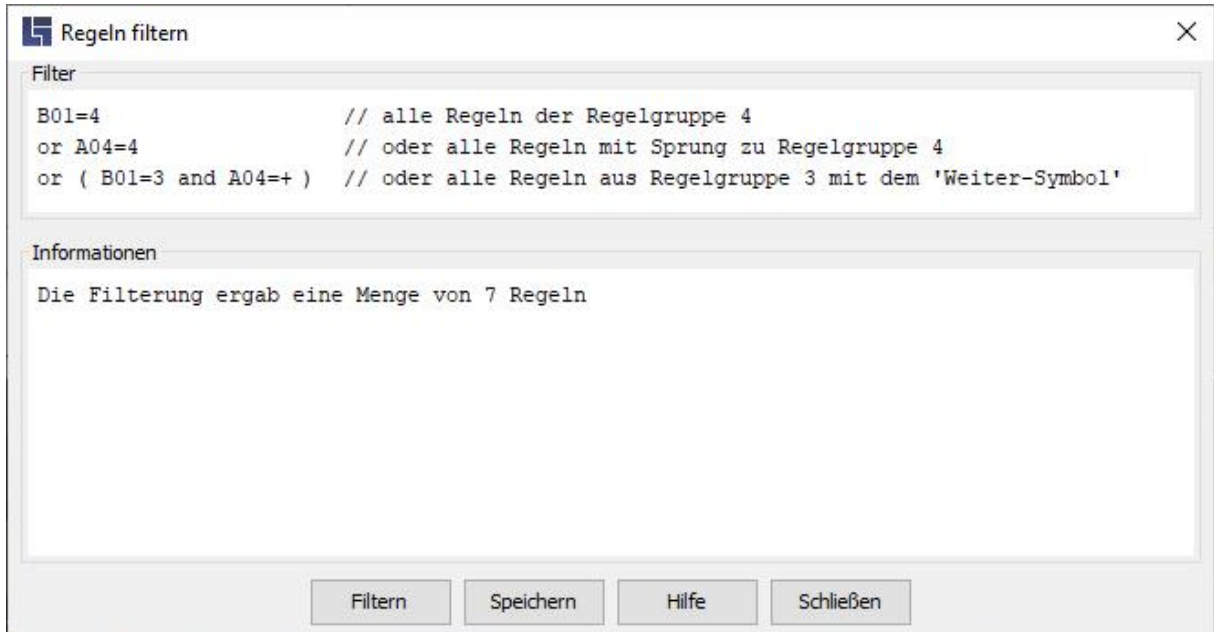
PRODUKT
INFO

LF-ET 2.3.0

Release Notes

Regeln filtern

Mit einer neuen Filterfunktion können Regeln jetzt auch nach beliebig komplexen Kriterien gefiltert werden:



Die Filterungen können dauerhaft als Schnappschüsse gespeichert werden.

Simulation von Regel-Abläufen

Protokollierung und statistische Auswertung

Die einzelnen Simulationsschritte und die dabei durchlaufenen Regeln werden jetzt automatisch protokolliert und statistisch ausgewertet. Die Protokolle und Statistiken können beim Beenden der Simulation wahlweise gespeichert oder verworfen werden.

Backup-Dateien

Neu: Automatisches Zippen

Alle Backup-Dateien zu einer Entscheidungstabelle werden jetzt automatisch in eine ZIP-Datei gepackt und liegen entpackt nur noch vor, solange eine Entscheidungstabelle in LF-ET geöffnet ist.

Neu: Zweites, alternatives Management der Backup-Dateien

Wenn Entscheidungstabellen auf Netzlaufwerken liegen, dann können in seltenen Fällen beim Speichern störende Verzögerungen auftreten. Als Ursache konnte die Verwaltung der Backupdateien identifiziert werden, was zur Bereitstellung einer zweiten, alternativen Version des Backup-Datei-Managements führte.

Programm-Generierung

Erweitert: Extraktion von Informationen aus Entscheidungstabellen

Die aktuell für Java und Groovy verfügbare Übernahme von Informationen aus den Entscheidungstabellen wurde erweitert. So werden bei der Enum-Generierung für Intervalle jetzt z.B. auch automatisch ***boolean isInterval(double/int value)*** generiert.

Neu: Stateless, Generic & Annotations

Für die Sprachen Java und Groovy wird jetzt auch die Generierung von zustandslosen und generischen Softwaremodulen unterstützt, ebenso die Verwendung von Annotations.

Neu: Verhinderung unnötiger Neu-Generierungen

Bei automatischen Massen-Generierungen, z.B. im Kontext von CI/CD mit dem LF-ET-Maven-Plugin, kann jetzt das Überschreiben von bereits existierenden und funktional identischen Softwaremodulen unterdrückt werden: SkipUnchanged

Erweitert: Sprachen

Neu hinzugekommen sind Generatoren für Python, JavaScript und TypeScript.

Die dazugehörigen Kapitel haben es noch nicht von der TODO-Liste in das Handbuch geschafft, bei Interesse stehen wir aber gerne für Rückfragen, Demos oder mit Beispielprojekten zur Verfügung.

Testfall-Generierung

Erweitert: Export-Formate

Die Export-Formate wurden erweitert, aktuell sind verfügbar: Gherkin/Cucumber, Robot Framework, TestLink, aqua und Excel.

Neu: Datengetriebene Testfallgenerierung

Werden Testdaten bei der Testfallgenerierung verwendet, dann ist es möglich, dass mehr als ein Datensatz geeignet ist, um einen konkreten Testfall zu generieren. Mit einer neuen Option können jetzt zusätzlich zu den empfohlenen Testfällen auch datengetriebene Testfälle generiert werden.

Erweitert: Zustandsbasierte Testfallgenerierung

Beim "Zustandsbasierten Testen" mit LF-ET wird eine Regel als ein Zustandsübergang betrachtet. Bei der n-Switch-Überdeckung steht n für die Anzahl von Zuständen zwischen dem Anfangs- und dem End-Zustand eines Testfalles.

Seither war die Switch-Überdeckung auf 0-Switch und 1-Switch begrenzt. Dieses Limit wurde aufgehoben und LF-ET ermittelt für jede Entscheidungstabelle individuell die höchste wählbare Switch-Überdeckung.

Die Konfigurationsdatei Project.ini

Neu: Alternativer Name - LFET.Project.ini

Um die Konfigurationsdatei deutlicher dem Tool LF-ET zuzuordnen, kann der Datei jetzt der Präfix "LFET." vorangestellt werden: LFET.Project.ini. Die Groß-/Kleinschreibung ist dabei ohne Bedeutung.

Neu: Alternatives Format - LFET.Project.yaml

LFET.Project.yaml und Project.yaml sind alternative Namen für die Datei Project.ini und ermöglichen anstelle des klassischen Property- Formates (Schlüssel=Wert) die Verwendung der leichtgewichtigen Auszeichnungssprache YAML.

Neu: Private INI-Datei für eine Entscheidungstabelle

Wenn LF-ET bei der Suche nach Project.ini-Dateien für die aktuell geöffnete Entscheidungstabelle im selben Verzeichnis eine gleichnamige Datei findet, die zusätzlich zur Dateiendung .lfet noch die Endung .ini oder .yaml hat, dann fügt LF-ET diese Datei als erste in die Liste der gefundenen Project.ini-Dateien ein.

Neu: Project.ini-Dateien verschachteln

Mehrere Project.ini-Dateien können jetzt verschachtelt werden. So können gezielt einzelne globale Einstellungen, wie z.B. die Package-Definition für Java-Module, für eine Gruppe von Entscheidungstabellen in einem Unterordner gezielt übersteuert werden.

Verbessert: Suche nach Project.ini-Dateien

Der Algorithmus zur Suche nach Project.ini-Dateien wurde komplett überarbeitet und optimiert. In seltenen Fällen vorgekommene Verzögerungen beim Öffnen von Entscheidungstabellen, oder bei der Generierung von Programmen, sollten damit der Vergangenheit angehören.

Sonstiges

Korrekturen und Verbesserungen

Neben einigen Korrekturen wurden auch zahlreiche interne Prozesse überarbeitet, was u.A. auch in einer weiter verbesserten Performance resultiert. Die Benutzeroberfläche wurde leichtgewichtiger gestaltet und berücksichtigt jetzt die mit Java 11 eingeführte bessere Skalierung.

Systemumgebungen

Unverändert wird als einzige Systemvoraussetzung nur eine Java-Laufzeitumgebung ab Version 8 benötigt. LF-ET ist aktuell getestet und freigegeben bis Java Version 21.