



```

--- src/entityinspector.cpp
+++ src/entityinspector.cpp
@@ -31,6 +31,7 @@ EntityInspector::~EntityInspector() {

    bool EntityInspector::checkRegisteredEntities() {
        QStringList classes = EntityInstanceFactory::getRegisteredClasses();
+       classes.sort();
        QString msg = QDateTime::currentDateTime().toString(Qt::ISODate) +
            " - Start checking entities\n";
        this->logger->logMsg(msg, MsgType::INFO);
@@ -90,8 +91,17 @@ Entity *EntityInspector::instantiateEntity(const QString name) {
    void EntityInspector::checkMetaProperties(QHash<QString, QMetaProperty>
        &metaProperties, bool &ok, QHash<QString, Relation> &relations) {
        QString msg = "";
+   #ifdef DIFFFABLEOUTPUT
+       QMap<QString, QMetaProperty> metaPropertiesMap;
+       for (auto i = metaProperties.constBegin(); i != metaProperties.constEnd();
+           ++i) {
+           metaPropertiesMap[i.key()] = i.value();
+       }
+       for (auto i = metaPropertiesMap.constBegin(); i != metaPropertiesMap.constEnd(); i++) {
+   #else
+       for (auto i = metaProperties.constBegin(); i != metaProperties.constEnd();
+           ++i) {
+   #endif
        QString typeName = QString(i.value().typeName());
        if (!i.value().isWritable()) {
            ok = false;
@@ -118,7 +128,15 @@ bool EntityInspector::verifyRelations(Entity *&entity) {
        auto relations = entity->getRelations();
        QString msg = "";
        this->checkMetaProperties(metaProperties, ok, relations);
+   #ifdef DIFFFABLEOUTPUT
+       QMap<QString, Relation> relationsMap;
+       for (auto i = relations.constBegin(); i != relations.constEnd(); i++) {
+           relationsMap[i.key()] = i.value();
+       }
+       for (auto i = relationsMap.constBegin(); i != relationsMap.constEnd(); ++i) {
+   #else
+       for (auto i = relations.constBegin(); i != relations.constEnd(); ++i) {
+   #endif
        this->checkRelationTypos(i.key(), i.value(), ok);
        if (!metaProperties.contains(i.key())) {
            msg += "For relation " + i.key() + " no property exists!";
--- src/entitymanager.cpp
+++ src/entitymanager.cpp
@@ -223,6 +223,9 @@ bool EntityManager::save(QList<QSharedPointer<Entity>> &entities,

    bool EntityManager::startup(QString version, QStringList toInitialize,
        bool createIndices) {
+   #ifdef DIFFFABLEOUTPUT
+       qSetGlobalQHashSeed(1);
+   #endif
        QSharedPointer<Entity> dbm = QSharedPointer<DatabaseMigration>
            (new DatabaseMigration());
        QHash<QString, QVariant> map = QHash<QString, QVariant>();
--- src/logger.cpp
+++ src/logger.cpp
@@ -38,8 +38,12 @@ void Logger::lastError(const QSqlQuery &q, bool logQuery) {
    if (logQuery || q.lastError().isValid()) {
        const QString errorMsg = this->generateLogMsg(q.lastError());
        if (logQuery || !errorMsg.isEmpty()) {
-           QString msg = "{" + QString("\"time\":\") +
-               QDateTime::currentDateTime().toString(Qt::ISODate) + QString("\") + errorMsg;
+           QString timeStamp;
+   #ifndef DIFFFABLEOUTPUT
+           timeStamp = QString("\"time\":\") +
+               QDateTime::currentDateTime().toString(Qt::ISODate) + QString("\");
+   #endif
+           QString msg = "{" + timeStamp + errorMsg;
            msg += this->generateLogMsg(q) + "}";
            this->logMsg(msg, errorMsg.isEmpty() ? MsgType::DEBUG : MsgType::WARNING);
        }
--- src/logger.h
+++ src/logger.h

```

```

@@ -27,6 +27,7 @@ enum class MsgType { DEBUG, INFO, WARNING, CRITICAL, FATAL};
class Logger {
#ifdef QT_DEBUG
#define DEFAULTMSGTYPE MsgType::DEBUG
+#define DIFFABLEOUTPUT
#else
#define DEFAULTMSGTYPE MsgType::CRITICAL
#endif

```

**#2 - 23.05.2017 00:22 - Sebastian Diel**

- Datei mergeNonDiffable.png wurde gelöscht

**#3 - 23.05.2017 00:22 - Sebastian Diel**

- Datei mergeDiffable.png wurde gelöscht

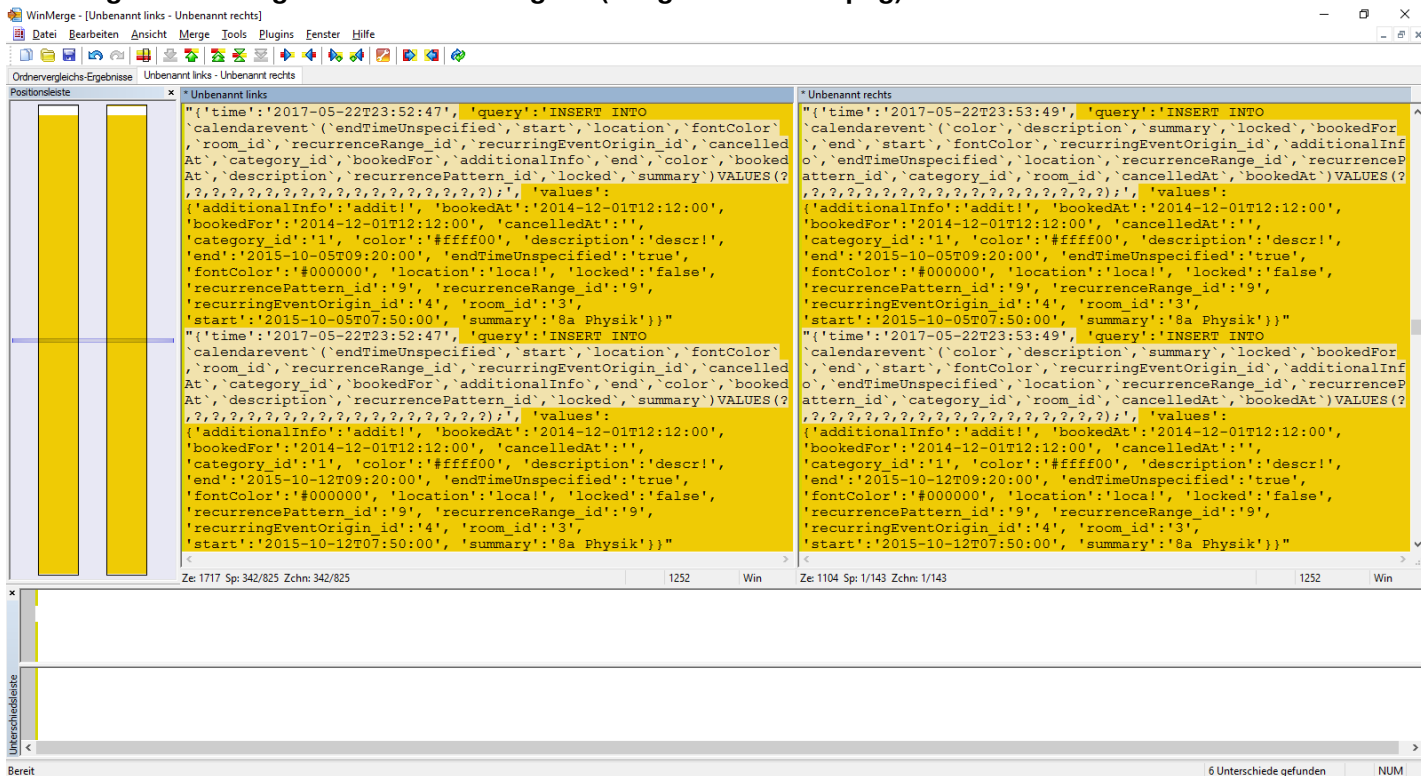
**#4 - 23.05.2017 00:28 - Sebastian Diel**

- Datei mergeNonDiffable.png wurde hinzugefügt

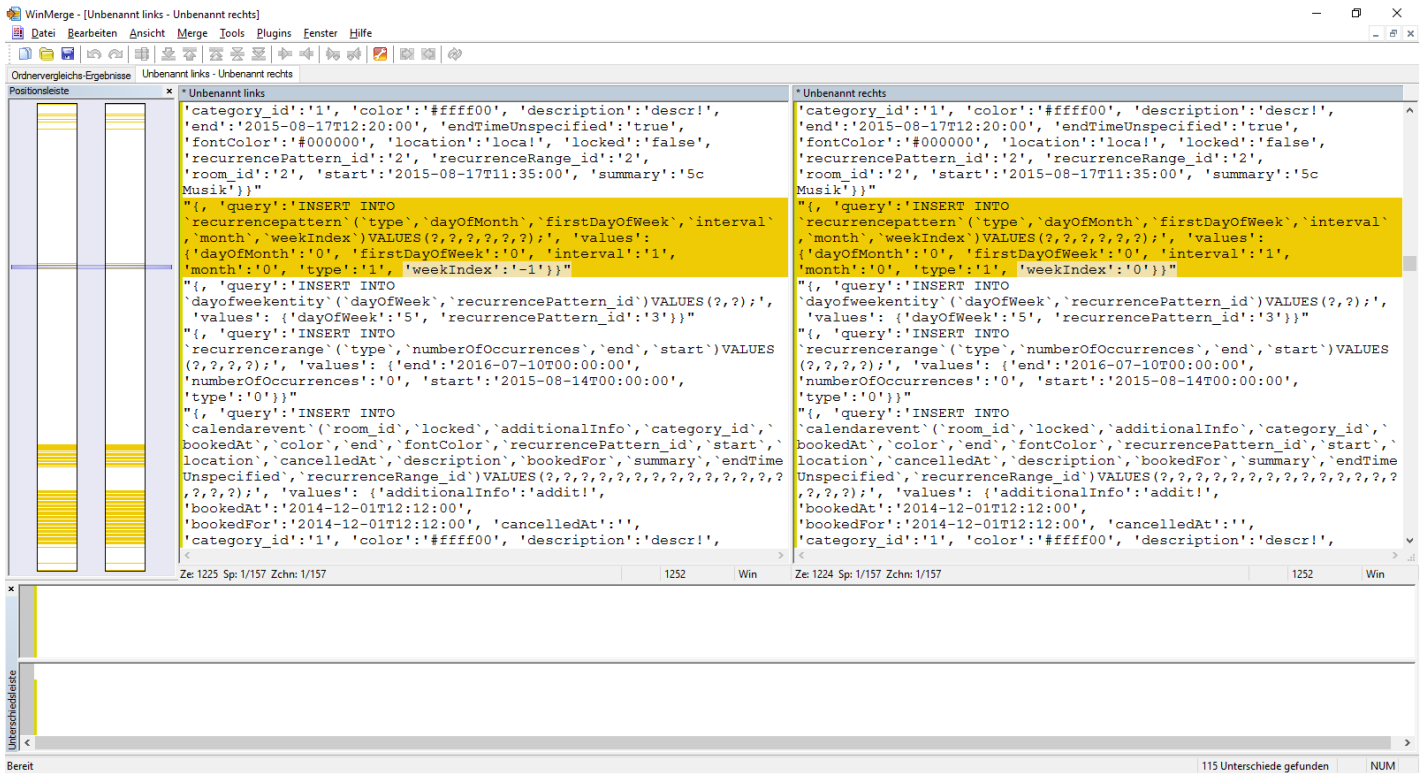
- Datei mergeDiffable.png wurde hinzugefügt

**Hiermit wird eine sinnvolle Nutzung von Diffs des Log-Outputs möglich.**

Ohne diese Änderungen ist nach zwei identischen Starts von CiT (jeweils incl. Neuerzeugung der Datenbank) der Vergleich der Logs nicht sinnvoll möglich (mergeNonDiffable.png)



**h3. Mit den angegebenen Änderungen ist schnell und klar ersichtlich, wo es zu Veränderungen gekommen ist (mergeDiffable.png):**



## h2. Es fällt sofort auf, welche Veränderungen es an welcher Stelle tatsächlich gegeben hat.

### #5 - 14.07.2018 22:34 - Anonym

- Status wurde von Neu zu Erledigt geändert
- % erledigt wurde von 0 zu 100 geändert

Status geändert durch Changeset [2cdded5d4cefef897161a2a202c06b4aade36cb4b](#).

### Dateien

Datei	Größe	Datum	Autor
mergeNonDiffable.png	109 KB	22.05.2017	Sebastian Diel
mergeDiffable.png	98,7 KB	22.05.2017	Sebastian Diel