

Konzept (Entwurf)

WAMAS Konfigurations- Overview, Konfigurations- Import und Dashboard

WAMAS Lift & Store 2.8

Standard · Deutsch (de-DE)

Konzept-Entwurf für drei Werkzeuge rund um die WAMAS-Lift-&-Store-
Datenbank

DB-Experte-Bot für Marek

 **WAMAS®**

 **SSI SCHÄFER**

V 0.1 · 16.10.2026

Inhaltsverzeichnis

1 Über dieses Dokument.....	4
1.1 Zweck.....	4
1.2 Dokumentinformationen.....	4
2 Ausgangslage.....	5
3 Werkzeug 1: Konfigurations-Overview.....	6
3.1 Ziel und Prinzip.....	6
3.2 Bereiche (Prototyp v0.2).....	6
3.3 Vergleichsmodus (Konzept).....	6
4 Werkzeug 2: Konfigurations-Import (nur Konzept).....	7
4.1 Excel-Vorlage.....	7
4.2 Metadatengetriebene Validierung (zur Laufzeit aus der Ziel-DB).....	7
4.3 Ablauf.....	7
4.4 Risiken.....	8
5 Werkzeug 3: WAMAS Dashboard (PWA).....	9
5.1 Architektur.....	9
5.2 Inhalte.....	9
5.3 Health-Checks (Prototyp).....	9
6 Stolperfallen.....	10
7 Praxisbeispiele.....	11
8 Offene Fragen.....	12
9 Quellenverzeichnis.....	13

1 Über dieses Dokument

1.1 Zweck

Dieses Konzept beschreibt drei Werkzeuge:

- **Konfigurations-Overview:** liest die Konfiguration eines Systems nur lesend aus und kann zwei Systeme vergleichen.
- **Konfigurations-Import:** legt Stammdaten aus einer Excel-Vorlage geprüft per SQL an.
- **WAMAS Dashboard:** zeigt Kennzahlen und Health-Checks als installierbare Web-App (PWA).

Für Overview und Dashboard gibt es bereits einen lauffähigen Prototyp. Der Import ist nur beschrieben, er wurde weder gebaut noch ausgeführt.

1.2 Dokumentinformationen

Merkmal	Angabe
Titel	WAMAS Konfigurations-Overview, Konfigurations-Import und Dashboard
Produkt / Release	WAMAS Lift & Store 2.8
Dokumentversion	0.1
Stand	16.10.2026
Autor	DB-Experte-Bot für Marek
Zielgruppe	Marek, Consulting, Inbetriebnahme
Grundlage	DB wms28_prod/wms28_arch (Testsystem), Prototyp WamasConfigOverview v0.2, Test GoodsCategory ROUTER vom 16.10.2026

Version	Datum	Änderung
0.1	16.10.2026	Erster Entwurf, inkl. Dashboard/PWA und Prototyp-Ergebnissen

Tab. 1: Änderungshistorie

2 Ausgangslage

- Die Zieldatenbank ist SQL Server 2025 Express. Alle Tabellen liegen im Schema dbo: 1207 Tabellen und 42 Views. Release 2.8 arbeitet mit Schnittstelle 2.7 (logimat.interfaceMode=ERP_2_7_0).
- Test vom 16.10.2026: Die Warengruppe ROUTER wurde per SQL angelegt, und zwar in GoodsCategory (Attribute von SWITCH kopiert, history_createdBy=root) und in GoodsCategoryDesc (ger, „Router“). Sie erscheint in **MD017** ohne Neustart.
- Bekannte Abhängigkeiten von GoodsCategory:
 - Fremdschlüssel clientId → Client, *_unit_id → DurationUnit, Strategie-IDs → WlsStorageStrategyCfg, GoodsCategoryDesc.langId → Language.
 - Check-Constraints auf id, state, recordState, verKind und monKind.
 - Die Spalten-Defaults weichen von den vorhandenen Daten ab: equivPeriod (10 yr) und itemAbcCalculation=1.
- Auf Marek's PC ist kein sqlcmd installiert. Alles läuft über Windows PowerShell mit .NET System.Data.SqlClient.

Quelle: Test vom 16.10.2026; Bericht WAMAS_wms28_prod_Ueberblick_v1.1.md

3 Werkzeug 1: Konfigurations-Overview

3.1 Ziel und Prinzip

- Nur lesend. Das Skript führt ausschließlich SELECT aus und prüft das selbst (Guard). Die Verbindung wird mit ApplicationIntent=ReadOnly und Application Name=WamasConfigOverview aufgebaut.
- Ausgabe:
 - Excel-Mappe mit einem Blatt pro Bereich und einem Blatt _Info, über das Modul ImportExcel.
 - Ohne ImportExcel: ein CSV pro Bereich (UTF-8, ;).
 - Immer zusätzlich ein JSON als Datenquelle für das Dashboard.
- Zugangsdaten werden nie in Dateien gespeichert. Das Skript fragt sie über Get-Credential ab oder nimmt ein -Credential-Objekt entgegen. Später ist auch der Windows-Anmeldeinformations-Manager möglich (Modul CredentialManager bzw. SecretManagement).
- Verschlüsselte Parameter (Typ ENCRYPTED) werden als *** ausgegeben.

3.2 Bereiche (Prototyp v0.2)

Blatt	Quelle
GlobalSystemParameter	GlobalSystemParameter
LogimatSystemParameters	LogimatSystemParameters
Strategien, Kriterien, ABC	WlsStorageStrategyCfg, ...CfgCriteria, WlsAbcStorageStrategy
Bedarfstypen	LogimatStorageDemandType, LogimatPickingDemandType, LogimatInventoryDemandType
Lagerort, Zonen, LOGIMAT, Öffnungen	WhLoc, LogimatZone, LogimatDcCfg, DcCfg, LogimatOpening
Bereiche, Platztypen	PhysArea, StratArea, StoLoc (Aggregation)
Tablarvorlagen	LmTrayTemplate
Stammdaten	GoodsCategory(+Desc), QuantityUnit(+Desc), ItemType, Client, SystemPartner, PostingKey
Rechte	Role, SystemUser (ohne Passwortfelder), Login2Role, DialogAccess
Schnittstelle	WlsInterfaceMappingProfile/Entry, *OutSpmDest, RecordTypeConfig
Druck, Nummernkreise, Jobs	DocumentTypeConfiguration, PrinterConfiguration, NumberRanges, JobConfig

Tab. 2: Bereiche des Prototyps (38 Bereiche + 26 Kennzahlen)

3.3 Vergleichsmodus (Konzept)

- Das Overview läuft gegen System A und System B, oder es lädt zwei vorhandene JSON-Dateien.
- Pro Bereich werden die Zeilen über ihren fachlichen Schlüssel abgeglichen. Den Primärschlüssel liest das Tool aus sys.indexes.
- Ergebnis: ein Blatt Diff_<Bereich> mit Status *nur A*, *nur B* oder *abweichend*. Bei Abweichung werden die betroffenen Spalten markiert.
- Technische Spalten werden ignoriert: history_*, versionCounting_* und *_normValue.

Quelle: Prototyp WamasConfigOverview_v0.2.ps1

4 Werkzeug 2: Konfigurations-Import (nur Konzept)

Nur Testsysteme

Direkte SQL-Schreibzugriffe gehen an der WAMAS-Applikationslogik vorbei. Das Werkzeug darf zuerst nur auf Testsystemen laufen. Vor einem produktiven Einsatz ist die Freigabe durch SSI SCHÄFER (Support/Gewährleistung) zu klären.

4.1 Excel-Vorlage

- Für jeden Stammdatentyp gibt es ein Blatt. Der Start ist mit GoodsCategory, QuantityUnit und ItemType.
- Pro Zeile sind diese Angaben vorgesehen:
 - der Schlüssel
 - die Spalte **Referenz** („Attribute kopieren von“, z. B. SWITCH)
 - nur die abweichenden Attribute
 - die Texte je Sprache (für die *Desc-Tabelle)
 - die Spalte **Modus** mit NEU (Standard) oder ÜBERSCHREIBEN. Überschreiben geschieht nur, wenn es ausdrücklich gewählt ist.

4.2 Metadatengetriebene Validierung (zur Laufzeit aus der Ziel-DB)

- Spalten, Typen, Längen und NULL-Fähigkeit kommen aus INFORMATION_SCHEMA.COLUMNS.
- Fremdschlüssel kommen aus sys.foreign_keys. Das Tool prüft, ob der referenzierte Wert existiert (z. B. DurationUnit, Language, WlsStorageStrategyCfg).
- Check-Constraints kommen aus sys.check_constraints (definition). Die zulässigen Enum-Werte werden aus IN (...) -Ausdrücken gelesen.
- Defaults kommen aus sys.default_constraints. Weichen sie vom Referenzdatensatz ab (z. B. equivPeriod, itemAbcCalculation), gilt der Referenzdatensatz, und das Tool zeigt einen Hinweis.
- Die Historienfelder werden automatisch gesetzt: history_createdAt/modifiedAt = jetzt, history_createdBy/modifiedBy = konfigurierbar (Standard root, wie im Test).

4.3 Ablauf

1. **Dry-Run** (Standard): Die Validierung läuft, das Tool erzeugt eine Vorschau mit einem Blatt Vorschau (INSERT/UPDATE/übersprungen/Fehler) und schreibt nichts.
2. **Apply** ist nur mit -Apply und einer Bestätigung möglich. Alle Änderungen laufen in **einer** Transaktion (SERIALIZABLE). Bei einem Fehler wird alles zurückgerollt.
3. Ein Log enthält jede Anweisung mit ihren Parametern, Benutzer und Zeit.
4. Ein Rollback-Skript wird automatisch erzeugt: DELETE für neue Zeilen und UPDATE mit den alten Werten für überschriebene.
5. Prüfung danach: Die Werte werden erneut gelesen, und es kommt ein Hinweis, den Dialog (z. B. MD017) zu prüfen.

```
$cols = SELECT COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH, IS_NULLABLE FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME=@t
$fks = SELECT fk.name, pc.name AS col, OBJECT_NAME(fk.referenced_object_id) AS refTab, rc.name AS refCol FROM
sys.foreign_keys ... WHERE parent=@t
$chk = SELECT name, definition FROM sys.check_constraints WHERE parent_object_id=OBJECT_ID(@t)
foreach ($row in $excel) {
    $neu = Merge(Referenzzeile, $row) # Attribute kopieren von
    Pruefe Laenge/Typ/NOT NULL; Pruefe FK-Werte per SELECT COUNT(*); Pruefe Enum gegen IN(...)
    if (Existiert($neu.Key) -and $row.Modus -ne 'ÜBERSCHREIBEN') { Status='übersprungen' }
}
Export Vorschau.xlsx # kein INSERT/UPDATE
```

Abb. 1: Skizze Dry-Run-Validierung (nur lesend, nicht implementiert)

4.4 Risiken

- **Cache der Applikation:** Im Test erschien `GoodsCategory` sofort. Für andere Typen (z. B. Strategien, Zonen, Parameter) ist das nicht belegt. Eventuell ist ein Neustart oder ein Cache-Reload nötig.

Cache-Verhalten

Welche Tabellen von WAMAS gecacht werden, ist nicht dokumentiert. Das muss pro Typ getestet werden.

- **WCM-/Systemsynchronisation und Versionszähler:** Spalten wie `versionCounting_counter` werden von der Applikation gepflegt. Ein falsches Setzen kann zu Konflikten führen.
- **Support/Gewährleistung:** Direkte DB-Änderungen sind eventuell nicht vom Support gedeckt.
- **Fehlende Fachlogik:** Plausibilitäten, die nur die Oberfläche prüft, werden umgangen.

Quelle: Test vom 16.10.2026; Systemkatalog der DB

5 Werkzeug 3: WAMAS Dashboard (PWA)

5.1 Architektur

1. WamasConfigOverview_v0.2.ps1 schreibt das JSON (meta, areas, stats, info). Optional wertet es mit -ArchDatabase wms28_arch auch das Archiv aus.
2. Das Dashboard besteht aus einer einzelnen HTML-Datei mit Vanilla JS und eigenen SVG-Diagrammen, ohne CDN. Es nutzt Hell/Dunkel-Modus, SSI-Gelb #FEEE19 und Arial und ist mobile-first.
 - WamasDashboard.html hat die Daten eingebettet und läuft lokal per Doppelklick.
 - index.html lädt data.json und wird für den PWA-Betrieb ausgeliefert.
3. PWA-Bestandteile:
 - manifest.webmanifest (Name „WAMAS Dashboard“, theme_color #FEEE19, Icons 192/512)
 - sw.js: speichert die App-Hülle im Cache. data.json wird zuerst aus dem Netz geholt, offline wird der zuletzt geladene Stand angezeigt.
4. Bereitstellung:
 - Ein nginx-Container wird **nur an 127.0.0.1:8088** gebunden.
 - tailscale serve --bg --https=443 http://127.0.0.1:8088 macht ihn per HTTPS im Tailnet erreichbar.
 - **Kein** funnel, **kein** Login. Der Schutz ergibt sich allein daraus, dass nur Geräte im Tailnet zugreifen können; optional zusätzlich eine Tailscale-ACL.

5.2 Inhalte

Reiter	Inhalt
Übersicht	KPIs (LOGIMAT, Tablare, Plätze, externe Plätze, Artikel, Bestand), Health-Checks, Aufträge nach Status, Archiv je Monat, Job-Läufe
Strategien	ABC → Strategie, Bedarfstypen mit Reservierungs-/Aktivierungsstrategie, Strategieliste, Strategie am Artikel, Kriterien
Lager	Stützplätze je LOGIMAT, Plätze je Komponente, Platzzweck, Behältertypen, Bestand Top 10
Artikel	Lagermodus, Mengeneinheiten, Warengruppen, Artikeltypen, ABC, Vollständigkeit Maße/Gewicht
Schnittstelle	OutSpm nach Satzart/Status, Systemmeldungen, Mapping-Profile, Schnittstellenparameter
Rechte / Parameter	Dialogrechte je Rolle, Benutzer, Rollen, alle globalen Parameter

Tab. 3: Reiter des Dashboards

5.3 Health-Checks (Prototyp)

- Telegramme im Status CONVEERROR, z. B. WLSSTOCKADJ002
- offene SysMsg
- fehlgeschlagene JobRun
- Warengruppen bzw. Mengeneinheiten ohne *Desc
- Anteil der Mengeneinheiten mit Maßen und Gewicht
- Stützplätze unbelegt trotz Bestand (offene Frage DB-Q01)

Geplant: Dialoge ohne Rechte und Artikel ohne Text.

Quelle: Prototyp WamasDashboard.html, Daten vom 16.10.2026 15:21

6 Stolperfallen

- Ohne das Modul ImportExcel schreibt das Overview CSV statt XLSX. Installation: `Install-Module ImportExcel -Scope CurrentUser`.
- Ein `PSCredential` darf nicht an einen neuen powershell-Prozess übergeben werden, weil dabei das Passwort verloren geht („Login failed“). Das Skript muss im selben Prozess laufen.
- Die Kennzahlen kommen aus einem **Testsystem**: Die Mengen sind Testdaten.
- Für die PWA-Installation ist HTTPS nötig, `file://` reicht nicht. Der Service Worker wird nur unter `https:` registriert.

7 Praxisbeispiele

1. **Vergleich Test ↔ Produktion vor Go-live:** Das Overview läuft auf beiden Systemen. Im Vergleichsmodus zeigt sich, dass `logimat.workingMode` abweicht (FREE gegenüber FORCE).
2. **Neue Warengruppe:** In der Excel-Vorlage steht eine Zeile ROUTER mit Referenz SWITCH und Text `ger=„Router“`. Der Dry-Run zeigt „INSERT, 0 Fehler“. Mit `-Apply` wird angelegt, das Rollback-Skript wird abgelegt, und anschließend wird **MD017** geprüft.

8 Offene Fragen

- Welche Stammdatentypen werden ohne Neustart von der Applikation übernommen? (Test pro Typ)
- Gibt es eine Freigabe durch SSI SCHÄFER, im Testsystem direkt per SQL zu schreiben?
- Bekommt das Dashboard regelmäßig neue Daten, z. B. über eine geplante Aufgabe auf dem PC, oder nur manuell?

9 Quellenverzeichnis

- DB srv01/wms28_prod und wms28_arch (Testsystem), nur lesende Abfragen vom 15./16.10.2026
- C:\Dev\Anleitungen\Datenbank\WAMAS_wms28_prod_Ueberblick_v1.1.md
- **Prototypen:** C:\Dev\Anleitungen\Datenbank\Tools\WamasConfigOverview_v0.2.ps1, ...
 \Tools\Dashboard\