

eBiss 2.6.84 (21.03.2011)

Inhalt

1. Schnittstellenänderungen	1
2. Fehlerbehebungen.....	1
2.1. eBiss Update scheitert bei Version 2.0.5 (s. ANF-02362-1BG2ZD)	1
2.2. TradeltemManagerDirectSql – Setzt Status auf geändert	1
2.3. Case-Sensitive DB mit neuen Prozeduren (@SupplierGln-Skalarvariable muss deklariert werden).....	2
2.4. Dump und Import.....	2
3. Neue Features	2
3.1. MessageDirection für den Report Generator	2
3.2. XMLWriter	2
3.3. Retail Artikel Pool.....	2
3.3.1. Überschreiben von Artikeln	2
3.3.2. Doppelte EANs im PRICAT.....	3
3.4. Supplier Artikel Pool – Performance Verbesserung.....	3
3.4.1. Performance (mehr als 490.000 Artikel mit Details in unter 3 Min.)	3
3.4.2. Neu Job Steps TradeltemManagerFillCache/-BookCache	3
3.4.3. TradeltemManagerBookCache	3

1. Schnittstellenänderungen

Diese eBiss Version enthält keine Schnittstellenänderungen zu den vorigen Version 2.4.* und 2.5.*.

Das Datenbankschema wurde geändert.

2. Fehlerbehebungen

2.1. eBiss Update scheitert bei Version 2.0.5 (s. ANF-02362-1BG2ZD)

Lösung: Das eBiss-Configure prüft die DB und gibt bei Versionen < 2.0.28 die Meldung aus, dass zuerst zur Version 2.0.28 aktualisiert werden muss.

Das Configure für die Version 2.0.28 steht auf der Pranke Web-Seite.

2.2. TradeltemManagerDirectSql – Setzt Status auf geändert

Der Fehler trat auf, wenn die EffectivePurchasePriceCurrency NULL ist und daher der ListPurchasePriceCurrency für die PurchasePriceCurrency verwendet wurde.

2.3. Case-Sensitive DB mit neuen Prozeduren (@SupplierGln-Skalarvariable muss deklariert werden)

Beim Einspielen der neuen Prozeduren für den Artikel-Pool trat der folgende Fehler auf, wenn die DB auf eine Case-Sensitive „Collation“ stand:

@SupplierGln-Skalarvariable muss deklariert werden.

Die Prozeduren wurden geändert und der Fehler ist behoben.

2.4. Dump und Import.

Fehler behoben, wenn beim Containerizer z.B. der CoreObjectWriter nicht gesetzt ist
Geschwindigkeit für den Import von Look-Tabellen verbessert.

3. Neue Features

3.1. MessageDirection für den Report Generator

Der Report Generator (Job Step ReportMessageCreator) wurde um die Eigenschaft MessageDirection erweitert.

3.2. XMLWriter

- Neues Encoding ISO-8859-1
- Neuer Modus zum Schreiben von leeren XML-Tags. Zu „EmptyTags“ gibt es nun folgende Einstellungen:
 - **DontWrite**: Leerwerte werden nicht ausgegeben.
 - **Short**: Ausgabe als <xxx/>
 - **Long**: Ausgabe als <xxx></xxx>

3.3. Retail Artikel Pool

3.3.1. Überschreiben von Artikeln

Ist der Artikel mit der EAN und der Kombination aus Nummer, Farbe, Größe schon vorhanden, dann findet wie bisher ein Abgleich der Daten im Pool statt.

Wird die EAN aber mit einer anderen Kombination aus Nummer, Farbe, Größe importiert, d.h. die EAN wurde wiederverwendet, dann werden die alten Informationen zu dem Artikel Detail im Pool gelöscht und das Artikel Detail wird neu hinzugefügt.

Dieses Verhalten kann durch die Variable **OverwriteEan** von der Location deaktiviert werden. Mit **OverwriteEan=False** wird ein Fehler geworfen, wenn die EAN mit einer anderen Kombination aus Nummer, Farbe, Größe importiert wird.

3.3.2. Doppelte EANs im PRICAT

Falls in einem PRICAT eine EAN oder die Kombination aus Nummer, Farbe, Größe mehrfach vorhanden ist, dann wird eine Fehlermeldung mit der Angabe des doppelten Wertes ausgegeben und der Import abgebrochen.

S. auch in der Hilfe unter „TradeltemManager Retail“

3.4. Supplier Artikel Pool – Performance Verbesserung

Diese Version verfügt über ein neues Verfahren Artikel und ArticleDetails im Supplier.ArticlePool zu verbuchen. Hierfür wurden zwei neue Jobsteps geschaffen, welche die zu verbuchenden Daten in eine Cache Tabelle lesen, um diese mittels Stored Procedures in die eigentlichen Article und ArticleDetail Tabellen einzutragen.

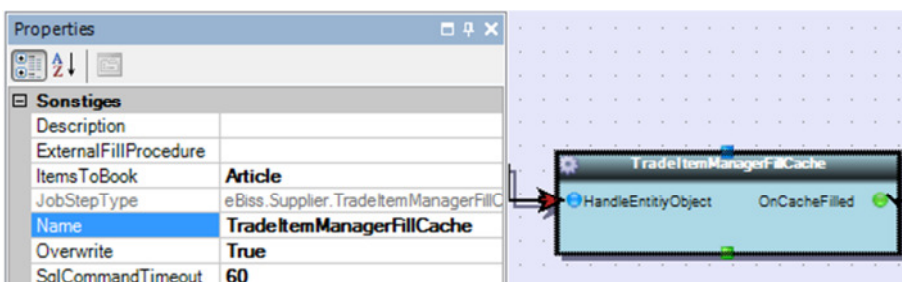
3.4.1. Performance (mehr als 490.000 Artikel mit Details in unter 3 Min.)

Das Verbuchen von über 490.000 EANs (Artikeln mit Details) dauert auf einem Entwicklungs-PC mit SQL-Server 2008 weniger als 3 Minuten und bei einer „warm“ gelaufenen DB sogar **unter 1 Minute**. Sowohl das Einbuchen als auch das Verbuchen bei Änderungen.

Hierbei wurden die Daten vom ERP-System über den View direkt bereitgestellt.

3.4.2. Neu Job Steps TradeltemManagerFillCache/-BookCache

Dieser JobStep speichert alle ArticleDetail und Article-Daten in einer Cache-Tabelle zum späteren Verbuchen durch den **TradeltemManagerBookCache-JobStep**.



Der TradeltemManagerFillCache JobStep kann entweder in den Workflow integriert, oder direkt mit Article-Daten über die View „**ArticlePool_FillArticleCache_View**“ gefüllt werden. Hierbei muss diese als Eigenschaft unter „**ExternalFillProcedure**“ eingetragen werden. Im Falle des direkten Preislinien-Verbuchens sollte dort dann die View „**ArticlePool_FillPriceLineCacheView**“ eingetragen sein. Ist die **Overwrite**-Eigenschaft auf **True** gesetzt, werden doppelte Ean-Positionen nach dem „Last-Wins“-Prinzip in die Cache-Tabelle geschrieben.

3.4.3. TradeltemManagerBookCache

Dieser JobStep ruft die Stored Procedures „**ArticlePool_UpdateArticles**“ und „**ArticlePool_UpdateArticleDetails**“ auf, um Article und ArticleDetails in die ArticlePool-Tabellen zu buchen. Ist die **Overwrite**-Eigenschaft auf **True** gesetzt, werden recycelte Ean-Positionen, welche im ArticlePool bereits vorhanden sind, gelöscht und die neuen ArticleDetails mit den geänderten Eigenschaften hierfür in den ArticlePool geschrieben.

Um Preislinien zu verbuchen sollte die Eigenschaft **ItemsToBook** auf **Price** gestellt werden. Für diese wird die Stored Procedure „**ArticlePool_UpdatePriceLines**“ verwendet.

