



**Compare and Contrast with
postgresql-autodoc and diff.**

Presented by: Perry Clark

Software Engineer

pclark@xtuple.com

www.xtuple.org



LoTR Model –

A single feature rich client to rule them all.
(well over 800K lines of client code)



PostBooks Edition

Core Postbooks - All Editions

- Database reporting
- International Language Packs
- Contacts
- Opportunities
- Incidents / Trouble Tickets
- ToDo's / Tasks
- Quotes
- Sales Orders
- Accounts Receivable
- Purchase Orders
- Accounts Payable
- General Ledger
- Shipping / Receiving
- Inventory (Single Warehouse)
- Item Master Information
- Bills of Materials (BOM)
- Work Orders
- Project Management
- Multi-Currency
- Financial reporting
- Multi-Locations in Warehouse
- Assemble-to-Order Configurator
- Kitting



Standard Edition

Standard Edition

- Multi-Company Financial Consolidation
- Multi-Warehouse Inventory
- Warehouse Transfer Orders
- Lot / Serial Control
- Item Transformations
- Returns / Service
- Revision Control
- Sales Order Reservations
- Drop-Ship Purchasing
- Consolidated Shipping by Customer
- Requirements Planning (DRP/MRP)



Manufacturing Edition

Manufacturing Edition

- Batch Manager
- Emailing Documents / EDI
- Multi-Level MRP
- Master Production Schedule (MPS)
- Bills of Operations (BOO)
- Breeder BOMs - Process Mfg.
- Capacity Planning
- Lean / Constraint Management



Some Statistics

```
grep CREATE pb350rc_schemadump.sql | cut -f-2 -d\ | sort | uniq -c
```

- 2 CREATE AGGREGATE
- 1278 CREATE FUNCTION
- 105 CREATE INDEX
- 278 CREATE RULE
- 1 CREATE SCHEMA (the 1 CREATE SCHEMA is for the api schema)
- 257 CREATE SEQUENCE
- 252 CREATE TABLE
- 121 CREATE TRIGGER
- 19 CREATE TYPE
- 12 CREATE UNIQUE (where all of the CREATE UNIQUE are indices)
- 123 CREATE VIEW

15(+/-) Developers

- **4 internal xTuple Developers**
- **3-4 casual internal contributors**
- **2-3 Contracted developers**
- **4 VAR developers**
- **Fluctuating end-user/community member developers**

How do we know what's different?

The Hard Question.

< How do I make sense of what's in database A and not in database B? I have autodoc output from each db.

> Use 'diff'.

< K...

< I used 'diff a b'... That's horrible. I think I need some plaintext template for AutoDoc.

> [shrug] make one.

< I created one for just the functions. I just stripped out all the HTML from the html.tmpl template. It works. But, now diff is still horrible.

```
<!-- TMPL_LOOP name="schemas" -->
  <!-- TMPL_LOOP name="functions" -->
    <!--TMPL_VAR ESCAPE="HTML" name="schema"-->.<!--TMPL_VAR ESCAPE="HTML" name="function"-->
    <!-- /TMPL_LOOP name="functions" -->
<!-- /TMPL_LOOP name="schemas" -->
```

Some real diff magic evolves.

< How do I get diff to tell me what's in file A, but not in file B?

> man diff, google, RTFM, etc... I'm trying to get some work done.

< K. What's this mean? It's from 'man diff'. I don't understand it at all.

`--changed-group-format=format`

Use format to output a line group containing differing lines from both files in if-then-else format.

> Google it. It means just what it says.

< Google doesn't know.

> You're impossible and you lie! Try this:

```
diff --unchanged-group-format="" --changed-group-format="" --old-group-format="%<"  
--new-group-format="" a b
```

< works good. :-)

The magic.

diff

--unchanged-group-format=""

--changed-group-format=""

--old-group-format="%<"

--new-group-format="" fileA fileB

Hacked up Autodoc templates – I.

<http://html-template.sourceforge.net>

One for tables. (name this tableOnly.tpl)

```
<!-- TMPL_LOOP name="schemas" -->
<!-- TMPL_LOOP name="tables" -->
    TABLE:<!--TMPL_VAR ESCAPE="HTML" name="table"-->
<!-- /TMPL_LOOP name="tables" -->
<!-- /TMPL_LOOP name="schemas" -->
```

One for functions. (name this funOnly.tpl)

```
<!-- TMPL_LOOP name="schemas" -->
<!-- TMPL_LOOP name="functions" -->
    FUNCTION: <!--TMPL_VAR ESCAPE="HTML" name="schema"-->.<!--TMPL_VAR ESCAPE="HTML"
name="function"-->
<!-- /TMPL_LOOP name="functions" -->
<!-- /TMPL_LOOP name="schemas" -->
```

Hacked up Autodoc templates - 2.

One for everything. (name this allPlain.tmpl)

```
<!-- TMPL_LOOP name="schemas" -->
SCHEMA: <!-- TMPL_VAR ESCAPE="HTML" name="schema" -->
<!-- TMPL_LOOP name="tables" -->
TABLE: <!-- TMPL_VAR ESCAPE="HTML" name="schema" -->.<!-- TMPL_VAR ESCAPE="HTML" name="table" --><!-- /TMPL_LOOP name="tables" -->
<!-- TMPL_LOOP name="functions" -->
FUNCTION: <!-- TMPL_VAR ESCAPE="HTML" name="schema" -->.<!-- TMPL_VAR ESCAPE="HTML" name="function" --><!-- /TMPL_LOOP name="functions" -->
<!-- /TMPL_LOOP name="schemas" --><!-- TMPL_LOOP name="schemas" --><!-- TMPL_LOOP name="tables" -->
TABLE: <!-- TMPL_IF name="number_of_schemas" --><!-- TMPL_VAR ESCAPE="HTML" name="schema" -->.<!-- TMPL_VAR ESCAPE="HTML" name="table" --><!-- /TMPL_IF name="number_of_schemas" --><!-- TMPL_IF name="number_of_schemas" --><!-- /TMPL_IF name="number_of_schemas" -->
<!-- /TMPL_LOOP name="schemas" -->
=====
<!-- TMPL_LOOP name="columns" --><!-- TMPL_VAR ESCAPE="HTML" name="column" -->.<!-- TMPL_VAR ESCAPE="HTML" name="column_type" -->
<!-- /TMPL_LOOP name="columns" -->
<!-- TMPL_IF name="view_definition" -->
VIEW DEFINITION: <!-- TMPL_VAR ESCAPE="HTML" name="schema" -->.<!-- TMPL_VAR ESCAPE="HTML" name="table" -->
=====
<!-- TMPL_VAR ESCAPE="HTML" name="view_definition" --><!-- /TMPL_IF name="view_definition" -->
=====
<!-- TMPL_LOOP name="functions" -->
=====
FUNCTION: <!--TMPL_VAR ESCAPE="HTML" name="schema"-->.<!--TMPL_VAR ESCAPE="HTML" name="function"-->
=====
<!-- TMPL_VAR ESCAPE="HTML" name="function_source" -->
<!-- /TMPL_LOOP name="functions" -->
=====
THE END!
<!-- /TMPL_LOOP name="schemas" -->
```

Putting it all together. Pt I

First, run postgresql-autodoc on the dbs you want to compare.

```
postgresql_autodoc -u postgres -h localhost -p 5432 -l /templates -t allPlain -d pb322  
postgresql_autodoc -u postgres -h localhost -p 5432 -l /templates -t allPlain -d mfg322
```

The -l option is important if you're using a custom template.
-t is the name of your template without the .tmpl extension.

Putting it all together. Pt 2

Then, run diff on the autodoc output.

```
diff --unchanged-group-format="" --changed-group-format="" --old-group-format="%<" --new-group-format="" mfg322.allplain2 pb322.allplain2 > output.txt
```

One way to check is to swap file B with A. In our application's case, there is nothing in PostBooks that is not in the Manufacturing edition, therefore there should be no output.

How we use PostgreSQL-Autodoc to help us?

- **Troubleshooting database issues** – We may not be able to immediately recognize a customer's internal changes to our functions, but we can detect if something totally foreign has been introduced (table, schema, function, etc.)

How we use PostgreSQL-Autodoc to help us?

- **Find differences between versions of our app** – PostBooks Vs. Standard Vs. Manufacturing or “What's in Manufacturing Edition that isn't in Standard or PostBooks?”

How we use PostgreSQL-Autodoc to help us?

- **Downgrade a database** – In current versions, we can remove a package (schema) quite easily, but previously everything was in one schema and required more thought before removing. It has always been easy to upgrade from PB to STD to MFG (additive). If a customer downgrades from MFG to STD or PB, typically the table data stays, only key functions are removed?

Can Autodoc help me do my dirty work?

Sure thing.

Suppose you want to downgrade a database by removing functions. You can simply create a template. (name this one dropFunctions.tpl)

```
<!-- TMPL_LOOP name="schemas" -->
<!-- TMPL_LOOP name="functions" -->
SELECT dropifexists('function','<!-- TMPL_VAR ESCAPE="HTML" name="function"--
>','<!-- TMPL_VAR ESCAPE="HTML" name="schema" -->');
<!-- TMPL_LOOP name="schemas" -->
<!-- TMPL_LOOP name="functions" -->
```

You end up with a script you can redirect into psql.

Thanks to...

Autodoc Contributors - Alan Williams, Andrew Dunstan, Andrew McMillan, Antanas Karashenski, Arthur Ward, Dirk Jagdmann, Hans Schou, Ilya Goldberg, Jacob Smullyan, Leonardo Cezar, Matt Bosworth, Neil Prockter, Paul Bort, Piebe de Vries, Rod Taylor, Tony Wasson, Toshio Kuratomi

The PostgreSQL Team - www.postgresql.org

The xTuple Team - www.xtuple.org and www.xtuple.com

