

# Base de Datos para Distribuidor

## WhitePaper

Abril 2016

# Contenidos

1. Propósito	3
2. Comandos	4
3. Operaciones adicionales	8

# 1. Propósito

El propósito de este documento consiste en enumerar y explicar los pasos requeridos para configurar una base de datos de distribuidor (punto de ventas). Este documento deberá ser leído por aquellas personas que quieran empezar una base de datos para un distribuidor o punto de venta.

## 2. Comandos

Ésta sección enumera los comandos necesarios para configurar una base de datos de distribuidor o punto de venta.

Para ejecutar estos pasos, abra una consola de comandos de Windows (*Inicio > Ejecutar > CMD.exe*) y diríjase al directorio donde los comandos están almacenados. (En este caso *C:\Archivos de Programa\Preference\PrefSuite\Commands*).

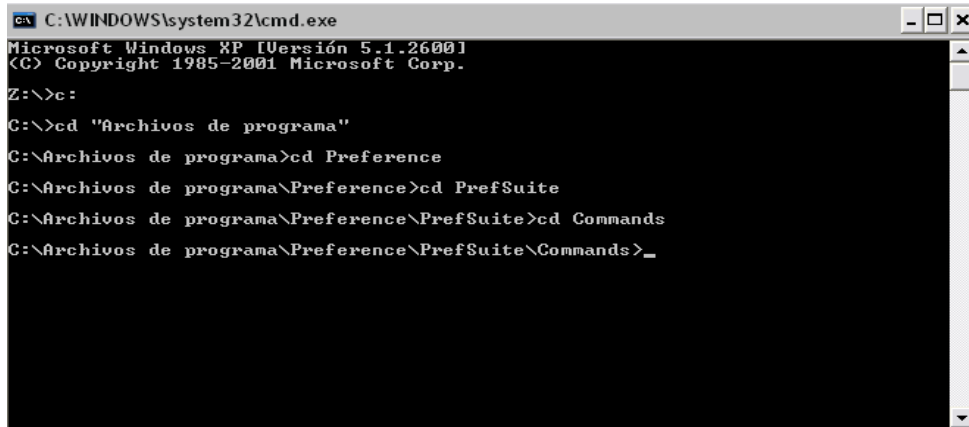


Figura 1. Acceso a comandos desde CMD.

1. Se debe crear desde la aplicación PrefDBManager una base de datos vacía. Éste es el único paso manual que se requerirá.

Esta aplicación la pueden encontrar en *Inicio > PrefSuite > Tools > PrefDBManager*.

Una vez abierta la aplicación, deben pinchar en el apartado de operaciones, sobre Nueva base de datos.

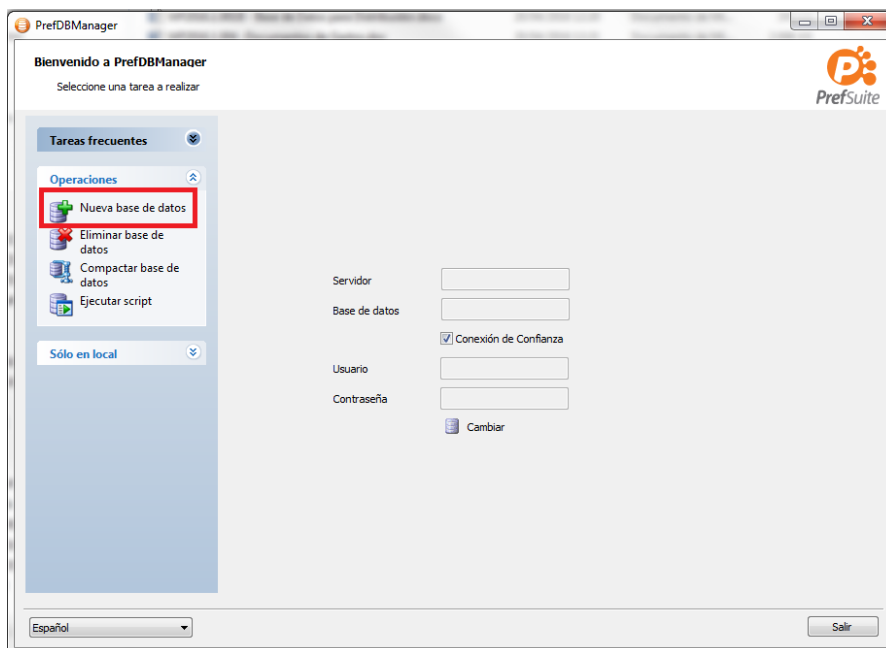
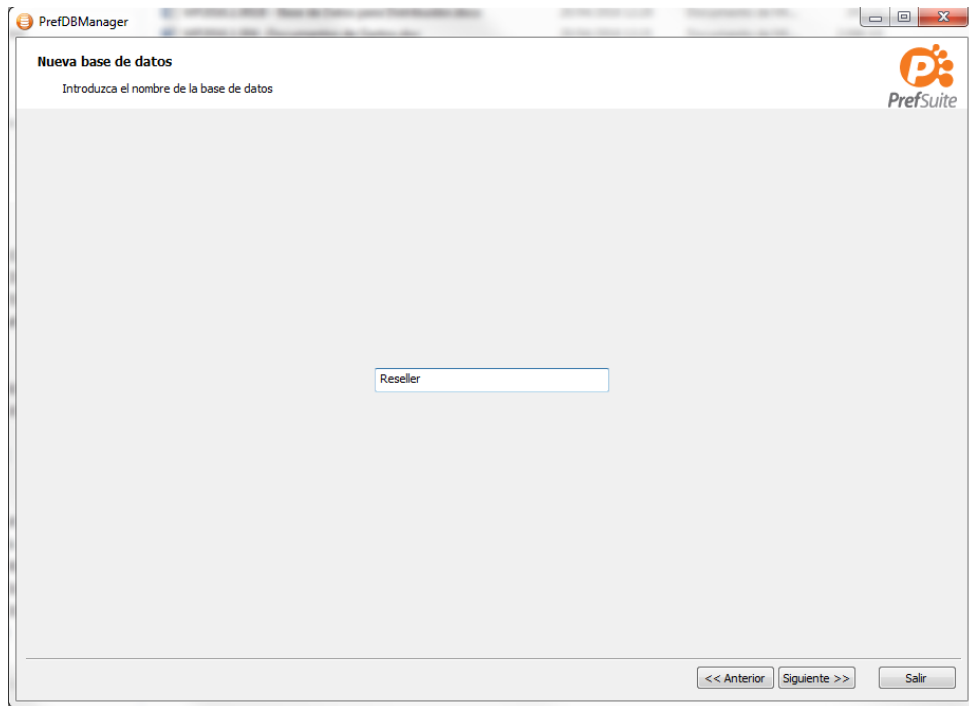


Figura 2. Nueva BD desde PrefDBManager.

Y luego indicar el nombre que desean que contenga la base de datos de tienda.



**Figura 3.** Indicar nombre de la BD de tienda.

**2.** UpdateDBStructure <Server> <ResellerDataBase>

Este comando actualiza la estructura de la nueva base de datos creada.

Ejemplo: UpdateDBStructure (local) Reseller1

**3.** InitializeDatabase <Server> <ResellerDatabase> <ResellerName> <DataversionName> 1

Este commando incializa la base de datos. Estamos insertando en la base de datos, unas configuraciones internas necesarias.

Ejemplo: InitializeDatabase (local) Reseller1 “Reseller 1 Name” “DV1R1” 1

**4.** IgnoreUserRights <Server> <ResellerDatabase>

Este comando deshabilita los permisos de usuario. Omita este comando si lo que desea es otorgarles distintos permisos a los usuarios.

Ejemplo: IgnoreUserRights (local) Reseller1

Aunque omitamos este paso, mediante la aplicación PrefUserRights, podremos marcar la opción de ignorar permisos, una vez terminada la base de datos.

**5. ExportResellerSettings <Server> <MasterDatabase> <TempDir>**

Este comando exporta algunas configuraciones de tarifas de la base de datos maestra. Es necesario y debe existir un directorio temporal, en este caso C:\Temp.

Ejemplo: ExportResellerSettings (local) Master C:\Temp

**6. ImportResellerSettings <Server> <ResellerDatabase> <TempDir>**

Este comando importa las configuraciones de tarifas del paso anterior en la base de datos del distribuidor o punto de venta.

Ejemplo: ImportResellerSettings (local) Reseller1 C:\Temp

**7. ExportDataversion <Server> <MasterDatabase> <DataversionName> <BKP File>**

Este comando exporta la versión de datos de la base de datos maestra. Un archivo .bkp, el cual contiene los datos, será creado.

Ejemplo: ExportDataversion (local) Master MasterDV1 "C:\Temp\Dataversion.bkp"

**8. ImportDataversion <Server> <ResellerDatabase> <Dataversion name> <BKP File>**

Este comando importa a la base de datos de distribuidor, o punto de venta, la versión de datos anteriormente creada.

Ejemplo: ImportDataversion (local) Reseller1 MasterDV1 "C:\Temp\Dataversion.bkp"

**9. UpdateResellerTariff <Server> <ResellerDatabase> <ResellerTariffName>**

Este script le dice a la base de datos del distribuidor o punto de venta la tarifa a utilizar de la base de datos maestra. Esta tarifa especial para distribuidor debe estar creada en la base de datos maestra antes de exportar la versión de datos.

Ejemplo: UpdateResellerTariff (local) Reseller1 "TarifaTienda"

**10. InsertResellerSalesTariff <Server> <ResellerDatabase> <SalesTariffName> <Currency>**

Este comando introduce una nueva tarifa de venta en la base de datos del distribuidor o punto de venta. Por defecto, ningún incremento se aplica. Este debe configurarse dentro de las tarifas en PrefGest. El nombre aplicado a la tarifa para la BD de tienda en este comando debe ser único. Es decir, siguiendo el ejemplo, no debe existir en la BD master una tarifa con nombre "PVPTienda".

Ejemplo: InsertResellerSalesTariff (local) Reseller1 "PVPTienda" "Euros"

**11. InsertDefaultData <Server> <ResellerDatabase> 1**

Este script introduce una forma de pago por defecto, un cliente por defecto, numeraciones por defecto, etc.

Ejemplo: InsertDefaultData (local) Reseller1 1

**12.** UpdateFieldsDescription <Server> <Database> <Language>

Este comando traduce los campos de la base de datos. Al momento de la escritura, solo el idioma Español está disponible.

Ejemplo: UpdateFieldsDescription (local) Reseller1 es

### 3. Operaciones adicionales

1. Si en nuestra base de datos master, existe algún proveedor que tenga asociada una materia prima con precios por kilo, se deberá traspasar dicho proveedor mediante un *prefbackup* de la tabla Proveedores.

También habrá que traspasar proveedores si en la base de datos master estos tienen definidos grupos de compra y las tarifas tienen definidos descuentos o incrementos por estos grupos.



Figura 4. PrefBackup de proveedores.

2. Hay que crear grupos de presupuestado en la tienda si se quieren presupuestar y valorar materiales sueltos, puesto que los grupos de la base de datos master no están disponibles para utilizarse en la Tienda.
3. Si se utilizan opciones genéricas en la base de datos master, estas se deben traspasar a la tienda mediante PrefBackup. Seleccionaremos la tabla *VariablesGlobales*, filtrando por el nombre *Opciones Genéricas*.





Figura 5. PrefBackup de opciones genéricas.

- Si en la base de datos Master utilizamos opciones de color, deberemos pasarlas a la base de datos de tienda mediante scripts. A continuación se muestra un ejemplo de este script.

Lo primero que haremos será visualizar las opciones de color que tenemos en la base de datos master, realizaremos una consulta filtrando por el tipo 3 (estas son las de color).

```
select * from opciones where tipo = 3
```

DataVerId	RowId	LCRowId	Nombre	Descripcion	Tipo	Nivel1	Nivel2	Nivel3	Nivel4	Nivel5	DefaultAl...	DefaultNumericalValue
1	CDF46D28-1F...	E03C89F...	Color Opcion		3							0
2	CDF46D28-1F...	0AA7DE0...	C_MOTORMANDO		3	Colores						0
3	CDF46D28-1F...	E0E7C9B...	Colores accesorios		3	Colores						0
4	CDF46D28-1F...	A96CEF2...	ALUSWING ColorAccesorios	Opcion de col...	3	Colores						0
5	CDF46D28-1F...	6BC8061...	hec		3	Corredera	Herraje					0

De esta consulta deberemos tener en cuenta el campo *nombre*, para realizar una consulta a la tabla *contenidoopciones*, filtrando por el nombre de la opción a insertar.

```
select * from contenidoopciones where opcion = 'Nombre_Opcion_Color'
```

DataVerId	RowId	LCRowId	Opcion	Orden	Valor	Texto	Disparadas	Flags	DesAuto	OrdenDesAuto	DesProd	OrdenDesProd	Esquema	IdGrupoPresupuestado	IdG
1	CDF46D28...	4ED6F...	Color Opcion	0	Color 1		0x424C4F425A4950001400000...	0	0	0	0	0	NULL	NULL	NU
2	CDF46D28...	D8CB6...	Color Opcion	1	Color 2		0x424C4F425A4950001400000...	0	0	0	0	0	NULL	NULL	NU

En este caso tenemos dos resultados, porque también disponemos de dos *dataversions* en esta base de datos.

Luego visualizaremos las versiones de datos para las que se tienen que crear las opciones, y su contenido. Deberemos utilizar el campo *ID* de esta consulta.

```
select * from dataversions
```

	DataVerId	MakerId	Name	DataVersionDate	CurrentDataVersion	QuotationDataVersion
1	CDF46D28-1FD9-40EC-87CB-A777FEAEF0E2	4155C29D-66F0-4692-A7BB-73A943CB3C16	version1	2005-06-30 00:00:00.000	1	1

En esta otra consulta, introduciremos los valores que hemos visualizado en la consulta anterior a la tabla *Opciones*, con el *DataVerId* de la *DataVersion* a la que queremos pasarlo.

```
INSERT INTO opciones (DataVerId, RowId, LCRowId, Nombre, Descripcion, Tipo, Nivel1, Nivel2, Nivel3, Nivel4, Nivel5, DefaultAlphanumericalValue, DefaultNumericalValue, VfoAlias, GenerateVariable, VariableName)
VALUES ('DataVerId', newid(), newid(), 'Nombre_opción', 'Descripción_de_la_opción', '3', 'Nivel1', 'Nivel2', 'Nivel3', 'Nivel4', 'Nivel5', 'DefaultAlphanumericalValue', '0', null, '0', null)
```

Nota: Los valores que introduciremos aquí no son aleatorios, son un duplicado de las opciones, cambiando el *DataVerId*.

Y por último insertaremos también el contenido de la tabla *ContenidoOpciones*, perteneciente a la opción que hemos insertado antes (cada línea hace referencia a cada uno de los valores que podemos ver en la opción).

```
INSERT INTO contenidoopciones (DataVerId, RowId, LCRowId, Opcion, Orden, Valor, Texto, Disparadas, Flags, DesAuto, OrdenDesAuto, DesProd, OrdenDesProd, Esquema, IdGrupoPresupuestado, IdGrupoProduccion, VfoAlias)
VALUES ('DataVerId', newid(), newid(), 'Nombre_Opcion', 'número_de_línea(Orden)', 'Valor', 'Texto (Descripción_de_la_línea)', 'Disparadas', '3', '', '0', '', '0', null, null, null, null, null)
```

Nota: Los valores que introduciremos aquí no son aleatorios, son un duplicado del contenido de las opciones, cambiando el *DataVerId* al mismo que hemos utilizado al insertar la opción.

Deberemos repetir estos dos últimos scripts por cada opción de color, y *DataVersion* distinta que tengamos en la base de datos.

5. Si en nuestra base de datos master existen datos de Mercado CE, y queremos traspasarlo para que la tienda pueda presentar presupuestos con esta información, se deberá traspasar a la BD de tienda mediante un *prefbackup* de las siguientes tablas:

- CE\_Params
- CE\_Params\_Values
- CE\_Test\_Params
- CE\_Tests
- CE\_Result\_Params
- CE\_Result\_Values
- CE\_Test\_Results
- CE\_ReductionRules
- CE\_UFrame

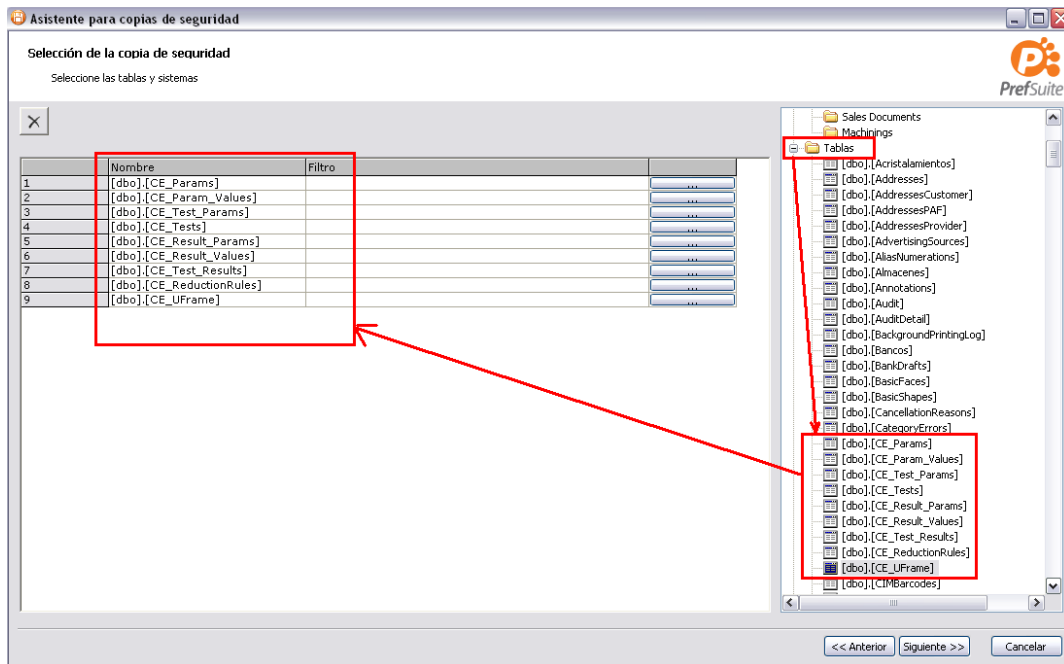


Figura 6. PrefBackup de marcado CE.

6. Si en nuestra base de datos master existen tablas de tarifas importadas, y queremos valorar materiales que podrán presupuestar nuestros clientes, para que la tienda pueda presentar presupuestos con esta información, se deberá traspasar a la BD de tienda mediante un *PrefBackup* de las siguientes tablas:

- Tarifario
- ContenidoTarifario

Una vez restaurado el PrefBackup, habrá que verificar que la DataverID de ContenidoTarifario es correcto (podemos compararlo, por ejemplo, con la tabla “Escandallos”, ejecutando las sentencias `SELECT * FROM ContenidoTarifario` y `SELECT * FROM Escandallos`).

En caso de que el campo DataverID de ambas tablas no coincida, se deberá modificar el valor de DataverID de tabla ContenidoTarifario, de forma que sea el mismo que el de la tabla Escandallos.

La forma de cambiarlo será con la instrucción UPDATE (por ejemplo, `UPDATE ContenidoTarifario SET DataverID=N'XXX'`, siendo XXX la dataversion correcta).