

## Price v12.0.5 Patch Readme

### Build IDs:

Price-12.0.5-262-200801291650-213927  
Installer-2.0.6-61-200707051230-201822  
Suite-1.3.6-205-200710081237-208448  
Enginepkg-4.0.10-424-200801170005-213409

This Patch is an update to Price v12.0. It contains all the fixes and enhancements from previous patch releases. Refer to the individual patch readmes for details.

The most recent update of the Price documentation set occurred for the v12.0.3 release.

### Important Upgrade Notes:

- This patch includes a fix to User Management to comply with the Sarbanes-Oxley Act (**TT 24556**). If you need to revert to the prior User Management behavior, see the next section under TT 24556 for instructions.
- A new view, `ir_item_daily_weights_override`, provides a means to configure daily weights (**TT 24527**). This new view must be added to your existing customized `ir.sql` file. Details about configuring this new view are provided at the end of this document.
- Price v12.0.5 is certified to work on Red Hat Enterprise Linux 4.0.

In order for Price v12.0.5 to work on Red Hat Enterprise Linux 4.0, you must:

1. Upgrade from Red Hat Enterprise Linux 3.0 to Red Hat Enterprise Linux 4.0 (It is recommended that you do a clean installation of Red Hat Enterprise Linux 4.0 and not upgrade from an existing Red Hat Enterprise Linux 3.0 version.)
2. Upgrade from Price v12.0.x to Price v12.0.5.

Complete the following steps to upgrade from Price v12.0.x to Price v12.0.5.

1. Back up the database, configroot, and other custom scripts.
2. Upgrade Red Hat Enterprise Linux from 3.0 to 4.0. (It is recommended that you do a clean installation of Red Hat Enterprise Linux 4.0 and not upgrade from an existing Red Hat Enterprise Linux 3.0 version.)
3. Install Oracle Database 10g 10.2.0.3 on the Red Hat Enterprise Linux 4.0 machine.
4. Install WebLogic 8.1 SP6 or Oracle Application Server 10.1.3.2 - as determined by your requirements – on the Red Hat Enterprise Linux 4.0 machine.
5. To install Price v12.0.5 on WebLogic: Verify that the JVM version is WL Jrockit 1.4.2\_12 (R27.1.0) (packaged with WebLogic).

To install Price v12.0.5 on Oracle Application Server: Verify that the JVM version is Sun JDK 1.4.2\_13. (You may have to download it from Sun's website.)

6. Create a new WebLogic domain with typical configurations.

or

Create a new Oracle Application Server instance with typical configurations.

7. Install the Price v12.0.5 application using the backed-up database from Step 1.  
KEY: Select the option to upgrade the database in the installer or in the install properties file. You must choose the upgrade option because the Price v12.0.5 patch contains changes to the application schema.
8. Replace configroot with the copy that was backed up in Step 1.
9. Replace all other customizations that were backed up in Step 1.

### Impact of Bug Fix on Price

The following table shows the TestTrack issues that were fixed in Price v12.0.5 and that impact either the Price front-end configuration (specifically the p4p-column-list.xml, p4pgui-config.xml, and config.properties files), the inference rules, load statements, schema, Standard Load, or Calc Engine. (Note that some issues do not fall into any of these categories.)

TT#	Front End Configuration	Inference Rules	Load Statements	Schema Change	Standard Load	Calc Engine
22715					X	
23887					X	
24166	X					
24390					X	
24485					X	
24527		X				
24533					X	

### Bug Descriptions

Following is a list of the TestTrack issues that were addressed in Price v12.0.5. This list includes all the bugs in the table above, as well as other bugs that were closed in the Price v12.0.5 release:

**22715** – Changes to address duplicate rows in location\_hierarchy\_tbl in a clustered implementation have been reverted.

**23887** – The LoadPlannedPromos procedure has been modified that the planned\_promos\_tbl and the planned\_promo\_maps\_tbl tables now match on planned\_promo\_id.

**24166** – The What If screen now displays consistent forecast metrics.

**24211** – Percentage values are not being converted to dollar values in the Worksheet display.

**24390** - The LoadSales procedure has been modified to address the issues of activities not reflecting correct sales in a clustered implementation.

**24485** – The LoadCollectionsSendback procedure has been modified to eliminate duplicate items.

**24527** – Daily weights have been made configurable instead of being hard-coded. The configuration is provided via `ir_item_daily_weights_override`. Details can be found in `ir.sql` and in the next section of this document.

**24533** – The `LoadCollectionsSendback` procedure has been modified to handle the addition of new items to a collection.

**24556** – In order to comply with the Sarbanes-Oxley Act, the Price Approver and Submitter roles now work in the following manner. This is the default behavior for Price 12.0.5.

- Submitter – The submitter role can submit worksheets but cannot approve worksheets.
- Approver – The approver role can approve worksheets but cannot submit worksheets.

In order to return to the behavior of previous releases of the application, in which the Approver role can also submit worksheets, apply the following SQL statement to the application database:

```
insert into role_action_tbl (role_id, action_id) values (1003, 6001);
commit;
```

**24653** – Pricing group names display correctly in the UI.

**24819** - Price v12.0.4 to Price v12.0.5 upgrade completes successfully.

### **Additional Information About `ir_item_daily_weights_override`**

The Calc Engine forecast model provides daily forecasts in support of markdown optimization. Prior to this release, hard-coded daily weights (seven values – one for each day of the week) were used in combination with the weekly forecasted sales units to determine the daily forecasts. However, the hard-coded daily values did not address the needs of all our customers (for example, stores that close on Sunday).

In this release, a new IR view, `IR_ITEM_DAILY_WEIGHTS_OVERRIDE`, provides different daily weights to override the default ones for each item. This view serves as the only configuration point, and no Front End change is required.

`IR_ITEM_DAILY_WEIGHTS_OVERRIDE` has 8 columns: `ITEM_ID`, `SUNDAY_WT`, `MONDAY_WT`, `TUESDAY_WT`, `WEDNESDAY_WT`, `THURSDAY_WT`, `FRIDAY_WT`, and `SATURDAY_WT` to accommodate a weight for each weekday and for each item whose daily weights need to be overridden. By default, this view has no records, so no daily weights are overridden.

The CE will apply the default daily weight to every item. The default values are:

<b>Day of Week</b>	<b>Default Daily Weight Value</b>
Sunday	0.2
Monday	0.1
Tuesday	0.1
Wednesday	0.1
Thursday	0.1
Friday	0.2
Saturday	0.2

The view is not required to have any record unless daily weights need to be overridden for some items. For items that do not have record in the view, the CE will apply the default daily weights.

To override the daily weights of an item, this view must contain exactly one record for the ITEM\_ID. The CE checks the validity of the daily weight record in two steps:

First, the weight for each weekday must be a non-negative double value, and weights for the same item must add up to 1. If any weekday has a negative value or a NULL value, or the values do not add up to 1, the CE will throw a “badDailyWeightsOverride” error and will not generate a forecast or markdown recommendation.

Second, the daily weights are compared with the historic data. If a particular weekday has a daily weight value of 0, but the historic daily sales on the same weekdays do not have a value of 0, the CE will throw a “badSeasonality” error. However, the CE does allow the daily weight and the daily sales on the same weekday to both have a value of 0 at the same time; in this case, it will forecast zero sales on the same weekday.

Here is the default ir\_item\_daily\_weights\_override from ir.sql:

```
CREATE VIEW IR_ITEM_DAILY_WEIGHTS_OVERRIDE
(
    ITEM_ID,
    SUNDAY_WT,
    MONDAY_WT,
    TUESDAY_WT,
    WEDNESDAY_WT,
    THURSDAY_WT,
    FRIDAY_WT,
    SATURDAY_WT
)
AS
SELECT
    i.item_id,
    0.2 as SUNDAY_WT,
    0.1 as MONDAY_WT,
    0.1 as TUESDAY_WT,
    0.1 as WEDNESDAY_WT,
    0.1 as THURSDAY_WT,
    0.2 as FRIDAY_WT,
    0.2 as SATURDAY_WT
FROM
    items_tbl i
WHERE i.end_dt IS NULL
AND 1 = 0
%{YA_TD}%
```

Here is an example of a customized version of the view in which all the items share the same daily weights that are different from the default ones.

```
CREATE VIEW IR_ITEM_DAILY_WEIGHTS_OVERRIDE
(
    ITEM_ID,
    SUNDAY_WT,
    MONDAY_WT,
    TUESDAY_WT,
```

```
WEDNESDAY_WT,  
THURSDAY_WT,  
FRIDAY_WT,  
SATURDAY_WT  
)  
AS  
SELECT  
    i.item_id,  
    0.0 as SUNDAY_WT,  
    0.1 as MONDAY_WT,  
    0.1 as TUESDAY_WT,  
    0.2 as WEDNESDAY_WT,  
    0.1 as THURSDAY_WT,  
    0.2 as FRIDAY_WT,  
    0.3 as SATURDAY_WT  
FROM  
    items_tbl i  
WHERE i.end_dt IS NULL;
```

### **Known Issues**

**24812** - What If - Doing a recalc without changing any other value produces a different output.