BF Admin Module System Health Check.doc

Last Updated: 31-Jan-2014

TABLE OF CONTENTS

1 Modification History

Date	Who	Comments
15-Aug-12	Solveda	Initial
11-Oct-12	Solveda	Clarified sub-menu; Removed "truncate" options from "Entity Row Size" function; Functions are now auto-retrieve;
16-Oct-12	Solveda	Added SHOPPING_LIST and SHOPPING_LIST_ITEM to the Entity-Row-Size function.
31-Jan-14	Solveda	Enhanced Scheduled Jobs Health Check to pull by Job Name, Status.

2 System Health Check

2.1 General

- Admin -> Tools

- Multi-Store Considerations ??

2.2 Processing Overview

- Each screen "sub section" will:
    - Initial display a description and an initial "analysis" button
    - After the analysis button is clicked the results and further actions will be displayed within the same component
    - All the available System Health Check options are available on a single page

- For example:

Sub Heading
This is a section that will do something
Do Analysis

Another Section
This is a section that will do something else
Analyze

- "Do Analysis" button is checked, the appropriate analysis will take place and a "results and corrective actions" section will open up for that section:

Sub Heading
This is a section that will do something
Do Analysis

Results and Possible Corrective Actions
The analysis shows that there are problems with the xxx. This can be corrected by clicking on the Correct button below.
Correct

Another Section
This is a section that will do something else
Analyze

2.3   Screen Basics

■   Page Heading

system health check
[message-area]

■   Back Button
   ○   Will appear at the end of the page

Back

3   Menu

3.1   UI Guideline

system health check

System Health Check Tools	
Function	**Description**
Entity Row Size	There are many OFBiz entities that can collect logging, visitor tracking and other information. These tables may contain a large number of rows depending on the specific configuration. Large row counts many impact stability and performance. This function will check and display row counts for various entities.
Logging Parameters	This check will review the settings for general logging of your implementation with recommendations on production settings.
Scheduled Job	Recurring Scheduled Jobs can generate a large number of entries in the OFBiz entity JOB_SANDBOX. This function will display various Scheduled Jobs counts and has corrective action tools.
Critical Settings	This tool will display critical Store based settings, provide descriptions and recommended settings.

4   Entity Row Size Health Check

4.1   UI Guidelines

<table>
<tr><td colspan="3">Entity Row Size Health Check</td></tr>
<tr><td colspan="3">There are many OFBiz entities that can collect logging, visitor tracking and other information. These tables may contain a large number of rows depending on the specific configuration. Large row counts many impact stability and performance. This function will check and display row counts for various entities.

For more information please view the Online Help function. See the Guide about Entity Row Sizes and Cleanup.</td></tr>
</table>

Results		
DB Entity	**#Rows**	
CART_ABANDONED_LINE	23,000	
SERVER_HIT	207,249	
SERVER_HIT_BIN	198,407	
SHOPPING_LIST	120,386	
SHOPPING_LIST_ITEM	1,736	
VISIT	63,382	
VISITOR	53,361	

4.2   Processing

- Should auto-retrieve entity row counts

- **TRUNCATE IS REMOVED – DO NOT IMPLEMENT**

    o   Confirm with "Are you sure you want to remove all rows from [entity-name]?"

    o   SQL: delete * from [entity_name]

    o   Success Message:
    - "207,249 rows were successfully deleted from the SERVER_HIT entity"

    o   Error Message:
    - "An error occurred while trying to delete rows from [entity-name]: [db-error]"

- **QUESTIONS TO DEV:**

- I can easily remove rows from SERVER_HIT and SERVER_HIT_BIN. However, I'm unable to remove from VISIT and VISITOR because of a foreign key constraint. Please provide additional information.

- I also noticed that the entity SHOPPING_LIST can get very large. I had the same foreign key constraint issue as above. We should be able to safely remove rows, at least some "old" rows. What are the dependent entities and risk?

4.3   Helper Text

CART_ABANDONED_LINE	The **CART_ABANDONED_LINE** entity logs every Add-To-Cart action on your website. If an order is placed then rows are removed from this entity. This is used to generate the Abandoned Cart emails. Cleanup can
SERVER_HIT	**SERVER_HIT** logging can normally be safely truncated unless you have specific requirements to view the data.
SERVER_HIT_BIN	**SERVER_HIT_BIN** logging can normally be safely truncated unless you have specific requirements to view the data.
SHOPPING_LIST	A row is created in the **SHOPPING_LIST** entity for every visitor, regardless of whether that visitor places items in the cart. The row may have an association to a **PARTY**, which indicates a registered user. These rows are used for the extracting potential Abandoned Cart Emails. If an order is completed then the information is removed from this entity.
SHOPPING_LIST_ITEM	Whenever items are actually placed in the shopping cart, a row is created in **SHOPPING_LIST_ITEM**. If an order is completed then the information is removed from this entity.
VISIT	The **VISIT** table logs all website Visits, whether anonymous or for registered users. There are many relationships to other entities so proceed with caution if attempting to remove data. To change the setting look at the **serverstats.properties** file, specific property is **stats.persist.visit**
VISITOR	The **VISITOR** table provides additional logging information. To change the setting look at the **serverstats.properties** file, specific property is **stats.persist.visitor**

5   Scheduled Jobs Health Check

5.1   UI Guidelines

Scheduled Jobs Health Check
Recurring Scheduled Jobs can generate a large number of entries in the OFBiz entity JOB_SANDBOX. This function will display various Scheduled Jobs counts and has corrective action tools. For more details about specific jobs you can access via Admin -> Scheduled Jobs.

Results and Possible Corrective Actions

Job Status and Name	#Rows	
FINISHED (more than 30 days ago)	1,299	
sendMailFromScreen	342	
		Delete
genProductsIndex	415	
		Delete
sendOrderConfirmation	370	
		Delete
sendOrderCompleteNotification	172	
		Delete
FINISHED (within the last 30 days)	48	
sendMailFromScreen	12	
		Delete
genProductsIndex	30	
		Delete
sendOrderConfirmation	6	
		Delete
FAILED	27	
sendMailFromScreen	27	
		Delete
CRASHED	4	
sendMailFromScreen	4	
		Delete
QUEUED	3	

RUNNING	0
CANCELLED	17
sendMailFromScreen	3
	Delete
genProductsIndex	8
	Delete
sendOrderConfirmation	6
	Delete
PENDING	10
genProductsIndex	4
	Delete
bigFishOrderFeeds	6
	Delete

5.2 Processing

- Should auto-retrieve scheduled job information

- SQL:

```
SELECT SERVICE_NAME,count(*) numRows FROM JOB_SANDBOX WHERE
STATUS_ID='[JOB_STATUS]' GROUP BY SERVICE_NAME ORDER BY SERVICE_NAME
```

- Clicking on the #rows will redirect to the "View Scheduled Jobs" List page, passing in contextual information (job-name, job-status).
    - o See spec "*BF Admin Module Services and Scheduled Jobs.doc*"

- Delete Action

    - o Confirm with "Are you sure you want to remove the [n] rows from JOB_SANDBOX for Job [job-name] where the status is [job-status]?"

    - o Delete from JOB_SANDBOX as appropriate

    - o Success Message:
        - "1,299 rows were successfully deleted from the JOB_SANDBOX entity"

    - o Error Message:
        - "An error occurred while trying to delete rows from JOB_SANDBOX: [db-error]"

- Helper Text

FINISHED (more than 30)	Deleting rows from the **JOB_SANDBOX** for **FINISHED** jobs that completed more than 30 days ago is considered a very safe action.

FINISHED (within 30)	Deleting rows from the **JOB_SANDBOX** for jobs that have **FINISHED** within the last 30 days is very safe, unless you need this history for other purposes.

6   Logging Parameters

6.1   UI Guidelines

<table>
<tr><td>

Logging Parameters

Logging parameters are located in the following file:

      Framework / base / config / debug.properties

The various levels of logging can be described as follows:

      verbose: The Verbose Level designates fine-grained informational events that are most useful to debug an application.

      timing: The Timing Level designates fine-grained informational events that are most useful to debug an application.

      info: The Info level designates informational messages that highlight the progress of the application at coarse-grained level.

      important: The Important level designates informational messages that highlight the progress of the application at coarse-grained level.

      warning: The Warning level designates potentially harmful situations.

      error: The Error level designates error events that might still allow the application to continue running.

      fatal: The Fatal level designates very severe error events that will presumably lead the application to abort.

Recommended settings for Production are:

      print.verbose=false
      print.timing=false
      print.info=false
      print.important=true
      print.warning=true
      print.error=true
      print.fatal=true

</td></tr>
<tr><td>

Results and Possible Corrective Actions

Active settings are:

      print.verbose=false
      print.timing=**true**
      print.info=**true**
      print.important=true
      print.warning=true
      print.error=true

</td></tr>
</table>

> print.fatal=true
>
> The System Administrator should be involved in making any changes to these settings.

6.2 <u>Processing</u>

- The main section (Logging Parameters) is fixed text

- The "Results" section shows the current settings for this instance

- Items that mismatch our recommend should be **bold**