

Arbitration Table Hot Fix

Objectives

- Allow using arbitration function, as designed.
- Reduce data sent to arbitration process
 - Reduce columns to only those needed
 - Reduce rows to only those needed
 - By Name
 - By Tag
 - By Attribute
- Allow customization globally and per-process
- Adjust behavior without restarting servers

Treatment Table Columns

Default	With Option
<code>TREATMENT_ID,</code> <code>TREATMENT_DESC,</code> <code>TREATMENT_REF_URL,</code> <code>TREATMENT_REF_TEXT,</code> <code>TREATMENT_CODE,</code> <code>TREATMENT_NAME,</code> <code>TREATMENT_COLUMN,</code> <code>TREATMENT_SK,</code> <code>TREATMENT_CONTRIB_CELL_PACKAGE_SK,</code> <code>TREATMENT_GUID,</code> <code>TREATMENT_FOLDER,</code> <code>TREATMENT_PARENT_FOLDER</code>	<code>TREATMENT_ID,</code> <code>TREATMENT_CODE,</code> <code>TREATMENT_SK,</code> <code>TREATMENT_CONTRIB_CELL_PACKAGE_SK</code>

Note: the name of the DS2 package is the sas process id. This option and the ones that follow cannot be set to false.

Option:

```
-Dcom.sas.analytics.crm.rtdm.arbitration.skinnyTables = true
```

-or-

```
-Dcom.sas.analytics.crm.rtdm.arbitration.<sas_process_id>.skinnyTables = true
```

Custom UDF Table Columns

Default	With Option
<code>XREF_TREATMENTS_ID,</code> <code>NAME,</code> <code>TAG,</code> <code>VAR_ID,</code> <code>TYPE,</code> <code>COLUMN_NAME,</code> <code>DYNAMIC_FLAG,</code> <code>REQUIRED_FLAG,</code> <code>NUM_VALUE,</code> <code>STRING_VALUE,</code> <code>DATE_VALUE,</code> <code>BOOLEAN_VALUE,</code> <code>LINK_VALUE,</code> <code>XREF_LIST_ID</code>	<code>XREF_TREATMENTS_ID,</code> <code>NAME,</code> <code>TAG,</code> <code>TYPE,</code> <code>NUM_VALUE,</code> <code>STRING_VALUE,</code> <code>DATE_VALUE,</code> <code>BOOLEAN_VALUE,</code> <code>LINK_VALUE,</code> <code>XREF_LIST_ID</code>

Note: Per row, only one “_VALUE” column is populated depending on type. Empty columns have negligible impact. This option includes UDF fields assigned at runtime.

Option:

```
-Dcom.sas.analytics.crm.rtdm.arbitration.skinnyTables = true
```

-or-

```
-Dcom.sas.analytics.crm.rtdm.arbitration.<sas_process_id>.skinnyTables = true
```

Custom UDF List Table Columns

Default	With Option
<i>LIST_ID,</i> <i>VAR_ID,</i> <i>ORDER,</i> <i>DISPLAY_VALUE,</i> <i>LIST_VALUE,</i> <i>TAG</i>	<i>LIST_ID,</i> <i>ORDER,</i> <i>LIST_VALUE</i>

Option:

`-Dcom.sas.analytics.crm.rtdm.arbitration.skinnyTables = true`

-or-

`-Dcom.sas.analytics.crm.rtdm.arbitration.<sas_process_id>.skinnyTables = true`

Reducing Columns, Additional

- You can also explicitly specify exact columns to send to arbitration globally or on per-process basis.

e.g.

```
-Dcom.sas.analytics.crm.rtdm.arbitration.columns=TREATMENT_ID, TREATMENT_CODE  
-Dcom.sas.analytics.crm.rtdm.arbitration.<sas_process_id>.columns=TREATMENT_ID
```

```
-Dcom.sas.analytics.crm.rtdm.arbitration.udf_columns=XREF_TREATMENT_ID, NAME, NUM_VALUE  
(also, <sas_process_id>)
```

```
-Dcom.sas.analytics.crm.rtdm.arbitration.udf_list_columns=LIST_ID, LIST_VALUE, TAG  
(also, <sas_process_id>)
```

Note: The per process id option always wins over the global option. However, the skinnyTables options override this option altogether. If you have skinnyTables set globally, then the “columns” option will not work. If you use the columns option, do not set global skinnyTables. However, it is possible to set skinnyTables per-process-id and set the columns option for other process ids.

Reducing Rows

- “Empty” UDF tables:
 - No custom detail data is sent

```
-Dcom.sas.analytics.crm.rtdm.arbitration.emptyUdfTables=true
```

```
-or-
```

```
-Dcom.sas.analytics.crm.rtdm.arbitration.<sas_process_id>.emptyUdfTables=true
```

- “Dynamic” UDF tables:
 - Will send all the details which are editable in the campaign

```
-Dcom.sas.analytics.crm.rtdm.arbitration.dynamicDetails=true
```

```
-or-
```

```
-Dcom.sas.analytics.crm.rtdm.arbitration.<sas_process_id>.dynamicDetails=true
```

Note: “Dynamic,” above, means “set in the campaign by the campaign designer”

Reducing Rows, Additional

- Specific UDF names:
 - Specify name list per process or globally

```
-Dcom.sas.analytics.crm.rtdm.arbitration.variables=Rate, Propensity, Value  
-or-  
-Dcom.sas.analytics.crm.rtdm.arbitration.<sas_process_id>.variables=Rate
```

- Specific UDF tags:
 - Only send those details which are in the tags that are listed

```
-Dcom.sas.analytics.crm.rtdm.arbitration.tags=INT_RATE  
-or-  
-Dcom.sas.analytics.crm.rtdm.arbitration.<sas_process_id>.tags=PROPENSITY, CREDIT_SCORE
```

Note: If you use “variable=Rate, propensity” and “tag=int_rate,” these options are additive. In other words, all three options take effect. This “additive” property is true only across these two types, above.

Adding the System Properties

- In startup scripts
(e.g. SASSErver6/wrapper.conf)
 - Necessary to persist desired behavior across restarts.
- At runtime
JBoss: /jmx-console SystemProperties MBean
WebSphere: IBM console
 - When activity created
- Campaign must be “Marked for deployment” again to pick up changes.

Note: To set these options in the IBM WebSphere Application Server console, go to the options for the SASSErver6 jvm. If you start your web application server using a shell script instead, modify SASSErver6.sh.

All of the options above are saved with the flow when the flow gets created. If you need to change the options associated with a flow, change the options, regenerate the flow, and redeploy it.