Aeos Analytics

Aeos Analytics is built on the TIBCO Spotfire analytics platform. It was added to baseline in Aeos version 15.2.

There are baseline Spotfire reports that are available, an Analytics security role type, and three Analytics roles (see Workflow, Reporting, and Analytics Roles).

For a description of these reports, see the Aeos Analytics section of the Aeos Reports Guide in the Product Management SharePoint Product Documentation/Aeos xx.x/Internal/Technical/Analytics folder.

For more information, see the Spotfire Administrator Guide in the Product Documentation/Analytics folder.

Custom Field Configuration

You can set up custom fields to be displayed in Aeos Analytics reports.

The following database objects exist in the schema in the ReportingDW database to support custom field configuration for Aeos Analytics reports.

Tables

Table: AnalyticsObject - All analytics objects (stored procedures) used to generate the analytics reports.

Table: AnalyticsTable - All Aeos tables that contain the CustomFieldXml column.

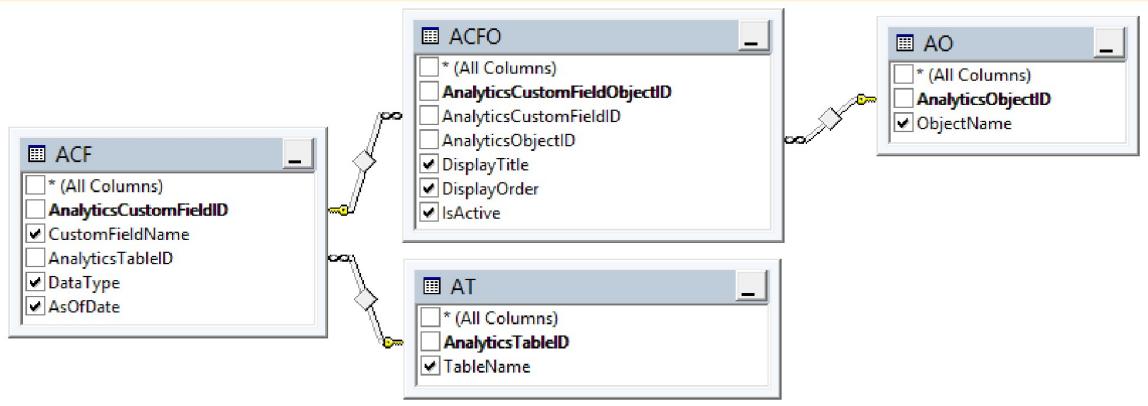
Table: AnalyticsObjectTable - Configuration that determines the association between Analytics objects (stored procedures) and Aeos tables. This table is important when configuring a new custom field. When a new custom field is configured using the stored procedure 'Analytics_ConfigureCustomFields', the stored procedure uses this table to looks up which analytics objects is associated with the new custom field.

Table: AnalyticsCustomField - Custom fields needed for the Aeos Analytics reports. The AnalyticsTableID column is a foreign key reference to the table AnalyticsTable which associates the custom field and its table.

Table: AnalyticsCustomFieldObject - Configuration between the custom fields and the analytics objects (stored procedures). It determines which custom fields to show on analytics reports. The value of the DisplayTitle column shows up on the report. For example, a custom field called 'CustomAccountAssignment' can be displayed on the report as 'Custom Account Assignments' if the DisplayTitle column holds that value. If the DisplayTitle is NULL, then the actual custom field name is shown on the report. The DisplayOrder field determines the order that the custom field is displayed on the report. By setting this correctly, you can set which custom fields to show on the report before other custom fields. The IsActive flag determines whether the custom field shows in the report.

Custom Fields View

View: vAnalyticsCustomFields - Existing custom fields that are configured to show on different Analytics reports.



	CustomFieldName	TableName	DataType	AsOfDate	ObjectName	DisplayTitle	DisplayOrder	IsActive
1	CustomAccountAssignment	Account	VARCHAR(MAX)	2015-01-30 10:19:06.3830000	Analytics_SelectAccountListing	CustomAccountAssignment	NULL	1
2	CustomAccountAssignment	Account	VARCHAR(MAX)	2015-01-30 10:19:06.3830000	Analytics_SelectVisitListing	CustomAccountAssignment	NULL	1
3	CustomAccountAssignment	Account	VARCHAR(MAX)	2015-01-30 10:19:06.3830000	Analytics_SelectUnitListing	CustomAccountAssignment	NULL	1
4	CustomAccountBoolean	Account	VARCHAR(MAX)	2015-01-30 10:19:06.4000000	Analytics_SelectAccountListing	CustomAccountBoolean	NULL	0
5	CustomAccountBoolean	Account	VARCHAR(MAX)	2015-01-30 10:19:06.4000000	Analytics_SelectVisitListing	CustomAccountBoolean	NULL	0

Configure Custom Fields Stored Procedure

Stored Procedure: Analytics_ConfigureCustomFields - Configures new custom fields for Analytics reports. It is not intended to update an existing custom field configuration. It takes six input parameters – CustomFieldName, CustomFieldTableName, CustomFieldDataType, CustomFieldDisplayTitle, CustomFieldDisplayOrder, and CustomFieldActiveFlag.

1. The mandatory parameter ‘CustomFieldName’ provides the custom field name for the configuration. It can take a string of up to 500 characters long. If this parameter is not supplied while executing the stored procedure or if a value of null / empty is passed, the following error message is returned by the stored procedure.

   "Custom field name cannot be null."

2. The mandatory parameter ‘CustomFieldTableName’ associates the custom field to the Aeos table that contains the custom field. For example, custom field name ‘CustomAccountBoolean’ is a part of the ‘Account’ table. So the parameter value of CustomFieldTableName should be Account. It can take a string of up to 250 characters long. If this parameter is not supplied while executing the stored procedure or if a value of null / empty is passed, the following error message is returned by the stored procedure.

   "Custom field table name cannot be null."

3. The mandatory parameter ‘CustomFieldDataType’ provides the data type of the custom field. The data type of the custom field is used while generating the analytics report. It can take a string of up to 50 characters long. If this parameter is not supplied while executing the stored procedure or if a value of null / empty is passed, the following error message is returned by the stored procedure.

   "Custom field data type cannot be null."

   The value of this parameter should be a valid SQL datatype with the correct precision. If not, the stored procedure returns error messages similar to the ones below.

   “The size (10000) given to the type 'varchar' exceeds the maximum allowed for any data type (8000)”

   "Type FAKE is not a defined system type."

4. The optional parameter ‘CustomFieldDisplayTitle’ provides a different custom field name to display on the reports as compared to the actual custom field name. For example, a custom field called ‘CustomAccountAssignment’ can be displayed in the report as ‘Custom Account Assignments’ if this parameter value is set to ‘‘Custom Account Assignments’. If it contains a value, then that value

is displayed on the report as the custom field name instead of the actual custom field name. If this parameter is not provided or its value is null/empty, the actual custom field name ('CustomAccountAssignment' is the above example) is displayed on the report.

5. The optional parameter 'CustomFieldDisplayOrder' controls the display order of the custom fields in the report. For example, if we have two custom fields named CustomAccountDecimal and CustomAccountDueDate, and the CustomAccountDueDate has a display order of 1 and CustomAccountDecimal has a display order of 2. Both these custom fields are associated to the 'Account' table. When the Account Listing Report is rendered, it shows the CustomAccountDueDate column before the CustomAccountDecimal column on the report. If the display order is not set on both these custom fields (DisplayOrder = NULL), then the display order is determined alphabetically. If this parameter is not provided, then a value of NULL is inserted into the DisplayOrder field on the table.
6. The mandatory parameter 'CustomFieldActiveFlag' determines whether to show or hide the custom field in the analytics reports. It takes a BIT value (0 or 1) as input. A value of 0 makes the custom field inactive and any other value (be it positive or negative) makes the custom field active.

Example of how to execute the stored procedure:

```
Use ReportingDW

GO

EXEC [dbo].[Analytics_ConfigureCustomFields]

   @CustomFieldName = 'CustomABCAssignment',

   @CustomFieldTableName = 'AccountBillingCycle',

   @CustomFieldDataType = 'VARCHAR(MAX)',

   @CustomFieldDisplayTitle = 'Custom Account Billing Cycle Assignment',

   @CustomFieldDisplayOrder = 1,

   @CustomFieldActiveFlag = 1
```

Custom Field Configuration Generation Script

Script: PBI 75696 - Aeos Analytics - Script To Generate Custom Field Configuration Data.sql - When setting up the custom fields for the first time. It generates the SQL to configure the custom fields to show in the analytics reports. This script scans through all the tables in the HealthcareWorkManagement database and determines which ones have a CustomFieldXml column. It then parses each custom field xml and reads the custom field names and generates the output results as shown below. This script does not directly insert data into the analytics custom field configuration tables. It only generates the SQL which needs to be executed manually after making sure that the entire configuration looks right. By default the generated SQL defaults the custom field active flag to 0 (inactive) and the custom field data type to VARCHAR(MAX).

Example Output:

```
EXEC [dbo].[Analytics_ConfigureCustomFields] @CustomFieldName =
'CustomAccountAssignment', @CustomFieldTableName = 'Account',
@CustomFieldDataType = 'VARCHAR(MAX)', @CustomFieldActiveFlag = 0

GO

EXEC [dbo].[Analytics_ConfigureCustomFields] @CustomFieldName =
'Unit_CustomAccountBillingCycleAssignment', @CustomFieldTableName =
'AccountBillingCycle', @CustomFieldDataType = 'VARCHAR(MAX)',
@CustomFieldActiveFlag = 0

GO
```

```
EXEC [dbo].[Analytics_ConfigureCustomFields] @CustomFieldName =
'CustomMedicalRecordString1', @CustomFieldTableName = 'AccountSponsorPlan',
@CustomFieldDataType = 'VARCHAR(MAX)', @CustomFieldActiveFlag = 1

GO
```

Location

This script is checked-in to TFS at:

$/Phoenix/Releases/15/15.2.0/15.2.0.00 Rel Aeos_Main_12.1.15143.180317/Database/PatchScripts/Client/15.2.0/Utilities

Hierarchy Setup

The Reporting DW database in Aeos uses a hierarchical structure to roll up data from the facility level up through the top level of the hospital. This structure allows for viewing or filtering data on various levels. Multiple hierarchies can be created which structure the organization in different ways.

For Aeos Analytics, this capability was added to the listing reports (Account, Unit and Visit). A hierarchy box presents a tree view of the nodes in the hierarchy and allows you to select various filters by clicking checkboxes in the tree. A drop-down list was added to which displays the various hierarchies. When selecting a hierarchy in the drop-down list, the tree view is refreshed with the appropriate nodes and levels in the tree. This section describes how to set up a hierarchy in a visualization.

The hierarchies were set up for a “Facility” based structure, but hierarchies can also be set up for “Areas and Responsibilities” with slight modifications.

Components

Stored Procedures

These two stored procedures return the data needed for the hierarchies.

- Analytics_DistinctHierarchies: Returns the distinct hierarchies on the client.

  Parameter: @LevelType – Identifies the lowest level of the hierarchy to find all hierarchies that use it. For example, “Facility” would return all hierarchies that use the facility DimNode.

- Analytics_Hierarchy: Returns hierarchy node and levels based on hierarchy name passed in. The hierarchy can be up to 10 levels deep.

  Parameter: @Hierarchy – Name of hierarchy

Procs (Spotfire)

Procs reference the stored procedures in the database.

- AnalyticsDistinctHierarchies: Spotfire procedure that references ReportingDW stored procedure.

  Input Parameter: @LevelType, string, Do Not Permit Null, Single Value

- AnalyticsHierarchy: Spotfire procedure that references ReportingDW stored procedure.

  Input Parameter: @Hierarchy, string, Do Not Permit Null, Single Value

Info Links

Info Links reference the Procs and are used directly in visualizations to display the data.

- IL_DistinctHierarchies: Info link returning data for distinct hierarchies
- IL_HierarchyHospital: Info link displaying hierarchy and nodes

Drop-Down List Control

This control contains a list of all the hierarchies returned. It also refreshes the tree view hierarchy control when a hierarchy is selected.

- Properties
  - Hierarchies
  - LevelType
- Script Parameters:
  - SelectedHierarchy: Property – "$(Hierarchies)"
  - distinctHierarhies: Data Table – IL_DistinctHierarchies
  - tableHierarchy: Data Table – IL_HierarchyHospital
- Script: SwitchHierarchies (Script Code)

```
import Spotfire.Dxp.Application.Filters as filters
import Spotfire.Dxp.Data as data
from System.Collections.Generic import List
from Spotfire.Dxp.Data import DataValueCursor

scheme = Document.FilteringSchemes[0]
#print(selectedHierarchy)
#print(distinctHierarchies.RowCount)
distinctHrCol = distinctHierarchies.Columns.Item["Hierarchy"]
distinctLevelsCol = distinctHierarchies.Columns.Item["MaxLevels"]
distinctHrColCursor = DataValueCursor.Create[str](distinctHrCol);
distinctLevelsColCursor = DataValueCursor.Create[int](distinctLevelsCol);

for row in distinctHierarchies.GetRows(distinctHrColCursor, distinctLevelsColCursor):
    if distinctHrColCursor.CurrentValue == selectedHierarchy:
        maxLevels = distinctLevelsColCursor.CurrentValue
    hrExpNew = List[str]()
    for i in range(maxLevels):
        hrExpNew.Insert(0, '['+str(i+1)+'] AS ['+str(i+1)+']')

    hr = scheme.Item[tableHierarchy].Item[tableHierarchy.Columns.Item["Hospital
    Hierarchy"]].As[filters.CheckBoxHierarchyFilter]()
    hrCol = hr.DataColumnReference.As[data.HierarchyColumn]()

    hrNew = data.HierarchyDefinition(hrCol.HierarchyDefinition.NestingMode, hrExpNew)
    hrCol.HierarchyDefinition = hrNew
    #for dv in hrNew.LevelExpressions:
```

```
            #print(str(dv))

        tableHierarchy.Refresh()
```

Hierarchy Filter Control

This control lists all the nodes in the hierarchy in a tree view. It is linked to the data by the lowest node ("1").

Setup

You can set up a hierarchy using the TIBCO Spotfire application on the Spotfire server.

Add Data Tables

In the Spotfire application click File | Add Data Tables

Select the 'Add' drop-down list and select 'Information Link' for the following tables in Report Objects:

1. Distinct Hierarchies: Add IL_DistinctHierarchies to visualization.
   When the prompt for parameter value comes up, click the 'Use On-demand' button
   Click the 'Settings' button
   Click the 'Define Input' button
   With the 'Fixed value' selected, enter "Facility" in the text box
   Click 'OK'

   Select the checkbox: Load Automatically
   Un-select the checkbox: Allow Caching

   Click 'OK'
   Click 'OK'
   Delete the newly created page for IL_DistinctHierarchies

2. Hospital Hierarchy: Add IL_HierarchyHospital
   When the prompt for parameter value comes up, click the 'Use On-demand' button
   Click the 'Settings' button
   Click the 'Define Input' button
   Check the input from 'Property' option button.
   Click 'Select'
   Click 'New'
   Enter "Hierarchies" as Property name.
   Select "String" as Data Type
   Enter "Hospital" as value.
   Click 'OK'
   Click 'OK'

Select the checkbox: Load Automatically

Un-select the checkbox: Allow Caching

Click ‘OK’

Click ‘OK’

Set up Relations

Go to Edit | Data Table

Click on ‘Relations’ tab

Click on ‘Manage Relations’

Click ‘New’ button

On the ‘New Relations’ screen

- Left data table
  - Left column: “Facility”
  - Data table such as “IL_VisitListing”
- Right data table
  - Data table: “IL_HierarchyHospital”
  - Right column: “1”

Click ‘OK’

Click ‘OK’

Click ‘OK’

Add Hierarchy Control

Click Insert | Hierarchy …

Set the Data table: IL_HierarchyHospital

Move column “1” to right Hierarchy box

Set Hierarchy name: “Hospital Hierarchy”

Set Filter by Hierarchy

Show Filters if not shown

Click on icon for filtering data tables next to name of info link

For ‘Filtering in IL_HierarchHospital’ select ‘Include Filtered Rows Only’

Select Edit | Organize Filters

For the newly added data tables hide all columns except “Hospital Hierarchy”

(Note. The “IL_HierarchyHospital” checkbox must be checked for the control to show)

Add Drop-Down List Control

Go to the text area and click ‘Edit Text’

Add a ‘Drop Down List’ control.

Update the ‘Set property value through: ‘ to “Unique values in column”

Under settings, set the Data table to “IL_DistinctHierarchies”

On the Document Properties tab, click ‘Script’

Click on ‘New Script’ and enter the script name: “switchHierarchies”

Copy the script code from above into the script area of the tab

Add the following script parameters

- SelectedHierarchy: Property – “$(Hierarchies)”
- distinctHierarchies: Data Table – IL_DistinctHierarchies
- tableHierarchy: Data Table – IL_HierarchyHospital

Click ‘Run Script’. If no errors appear in the Output window, click ‘OK’

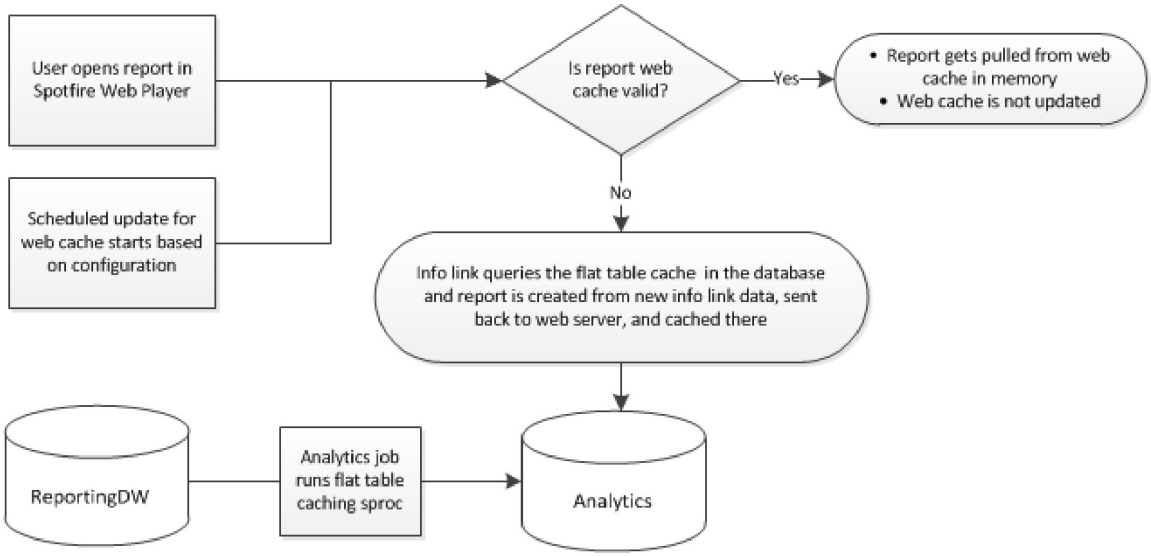
Type “Hierarchy: “ in front of the drop-down list as a label.

Spotfire Caching

To reduce Aeos Analytics report load times in Spotfire, two types of caching are implemented: scheduled updates and flat table caching.

- Scheduled updates are used to preload the report (analysis) files into the web cache in memory. When a user opens a report, the Spotfire Web Player reads the report directly from memory rather than loading from the information link cache or having to query the database. The web cache is updated at scheduled intervals.
- Flat table caching pulls snapshots of detail-level data used by Aeos Analytics reports from the Aeos reporting database into flat tables in the Analytics database. Rather than from the reporting database directly, these reports pull data from the flat tables.

Caching Process Flow



This diagram depicts overlapping processes:

- Report loading process
- Web cache scheduled update process
- Flat table caching process

The flat table caching process is described in detail in the Flat Table Caching section below.

1. The processes are initiated by one of the following:
   a. User opens report in Spotfire Web Player
   b. Scheduled update polls the web cache according to its configured schedule.
2. Is report web cache valid?

   If the report web cache is valid:
   a. The report loads from the web cache in memory on the Spotfire Web Server.
   b. The scheduled update process does nothing since an update is unnecessary.

   If it is not, these processes attempt to use the flat table cache in the Analytics database.

Scheduled Updates

Scheduled updates are used to preload report files into the web cache in memory on the Spotfire Web Player server. You need to configure which analyses to preload and when they are available in memory.

Read 6.6 Scheduled Updates in the TIBCO Spotfire Web Player Installation and Configuration Manual (*TIB_sfire_webp_6.5.2_InstallationManual.pdf*).

Configuration

Setting up the web cache involves editing the Spotfire Web Player configuration file, scheduling the report files that are preloaded, and starting the TIBCO Spotfire Web Player Keep Alive Service.

Default Settings

Spotfire Web Player web.config file

scheduledUpdates enabled: false

concurrentUpdates: 1

updateIntervalSeconds: 300

documentCache purgeInterval: 3600

LibraryCache_Enabled: false

ScheduledUpdatesUsername: sa_user

ScheduledUpdatesPassword: <*sa_user_password*>

ImpersonationUsername: sa_user

ImpersonationPassword: <*sa_user_password*>

Scheduled Update

Report preloaded every day between: 3:00 – 23:00

To configure the scheduled updates Web caching

1. Edit the highlighted settings in the following sections in the Spotfire Web Player web.config file (C:\Program Files\TIBCO\Spotfire Web Player\6.5.2\webroot).

   Section 1:

   <scheduledUpdates enabled="false" useLibrary="true" libraryFileName="ScheduledUpdates" settingsFile="App_Data\ScheduledUpdates.xml" concurrentUpdates="1" updateIntervalSeconds="300">

   <forcedUpdate enabled="true" maximumRejectedUpdates="2" />

   <stopUpdatesAfterRepeatedFail enabled="false" failsBeforeStop="3" />

```
    <externalUpdate keepAliveMinutes="10">
        <webService enabled="false" />
        <ems enabled="false" serverUrl="" topic="" clientId="" reconnectAttemptCount="10"
        reconnectAttemptDelayMilliseconds="1000" reconnectAttemptTimeoutMilliseconds="1000"
        />
    </externalUpdate>
</scheduledUpdates>
```

concurrentUpdates: When I tried setting the concurrent updates to a number greater than 1, the Account Listing throws an exception while trying to cache, because at some point, two or more instances are trying to cache the same report. Ideally, when this is set to 2, it should cache two different types of reports at the same time. It works fine for Unit listing and Visit Listing. They both begin their caching process at the same time. But later when only Account listing is yet to be cached, Spotfire checks and identifies that Account listing is still not cached and starts another instance to cache the same report. Since we have two instances trying to cache the same report, we get errors and the caching never completes.

updateIntervalSeconds: Setting the updateIntervalseconds to a higher value (say 3600, which is 1 hour), is not recommended. I tried this. What seems to happen if the account listing fails is that Spotfire tries to re-cache it only after 1 hour. The default value is 60 (1 minute). I have changed this setting to 300 (5 minutes). So whenever a report fails to cache with an error, it tries again in 5 minutes. There is also a setting to stop retrying after a specific number of retries. We have disabled that for now. Let us know if you think that we should set that value.

Section 2:

```
<documentCache purgeInterval="3600" itemExpirationTimeout="00:00:00" />
```

purgeInterval: Specify the number of seconds between server searches to identify unused, open documents (templates) to be purged. Default value is 300 seconds (5 minutes), range is 60 to 3600.

Set the purge interval to the max value possible (1 hour).

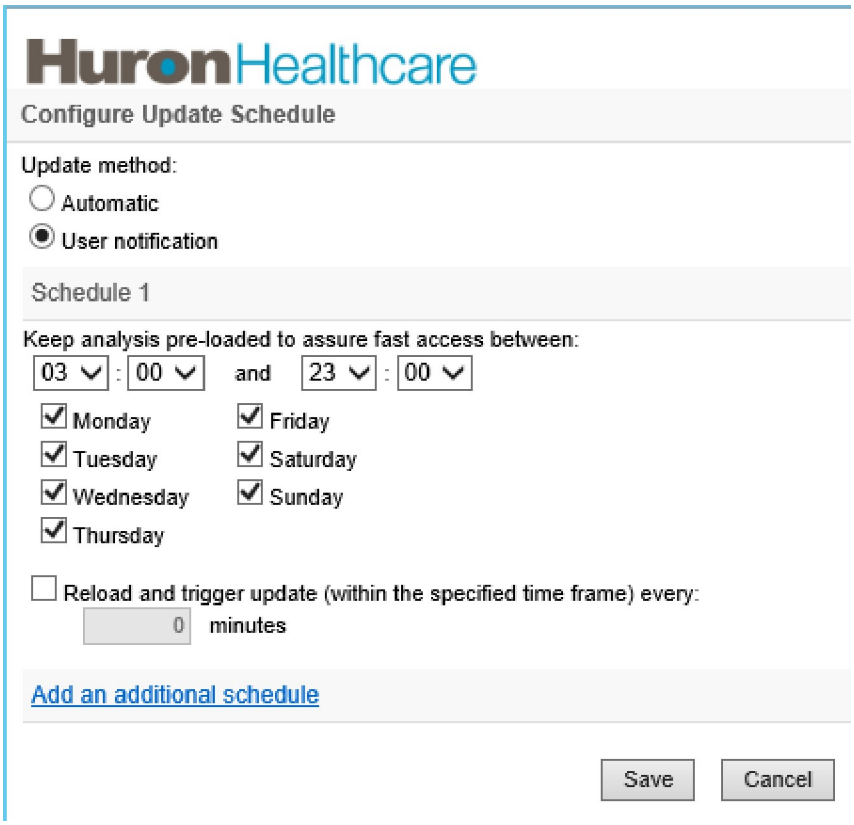
Section 3:

```
<setting name="LibraryCache_Enabled" serializeAs="String">
    <value>False</value>
</setting>
```

2. Configure the schedule:
    a. Go to the library by entering the address of the Spotfire Web Player in the web browser.

       Example: http://localhost:8080/spotfire/

       Note: To be able to configure the update schedule, you must be logged in as an administrator or a Spotfire Web Player administrator.
    b. Click **Tools > Scheduled updates**.
    c. If there are existing files scheduled for updates, click them to edit their update schedule or click **Add Analysis File**.
    d. Browse to the report file to set up an update schedule and click it to display the update schedule dialog.
    e. Configure the schedule and click **Save**.



3. Configure and start the Keep Alive Service
   a. Edit the Spotfire.Dxp.Web.KeepAlive.exe.config file from C:\Program Files\TIBCO\Spotfire Web Player\6.5.2\webroot\bin\Tools.
   b. Configure the following settings:

```
<applicationSettings>
   <Spotfire.Dxp.Web.KeepAlive.Properties.Settings>
      <setting name="SettingsFilePath" serializeAs="String">
         <value>C:\Program Files\TIBCO\Spotfire Web
         Player\6.5.2\webroot\app_data\ScheduledUpdates.xml</valu
         e>
      </setting>
      <setting name="PingIntervalMinutes" serializeAs="String">
         <value>10</value>
      </setting>
      <setting name="WindowsUserName" serializeAs="String">
         <value/>
      </setting>
      <setting name="WindowsPassword" serializeAs="String">
         <value/>
      </setting>
      <setting name="WebPlayerUrl" serializeAs="String">
```

```
        <value>http://localhost:80/SpotfireWeb/KeepAlive.ashx</v
        alue>
    </setting>
```

c. Start the Keep Alive Service.

   Go to Services and start the *'TIBCO Spotfire Web Player Keep Alive Service'* service.

4. Perform an IISReset.
5. Restart the *'TIBCO Spotfire Server'* service. Newly created schedules don't seem to get active until the *'TIBCO Spotfire Server'* service gets restarted.
6. Restart the *'TIBCO Spotfire Web Player Keep Alive Service'* service.
7. Make sure the schedule gets kicked off at the configured time and the analyses are cached.

Flat Table Caching

Flat table caching pulls snapshots of detail-level data used by Aeos Analytics reports from the Aeos reporting database into flat tables in the Analytics database. Rather than from the reporting database directly, these reports pull data from the flat tables. Flat table caching was introduced in 16.2, and replaces information link caching. Unlike information link caching, you must configure and run the flat table caching process in order to have functioning Aeos Analytics reports.

You can schedule flat table caching to run automatically as a SQL Agent Job. It should be configured so that flat table caching completes prior to the time that Spotfire web caching begins. To schedule flat table caching, follow these steps:

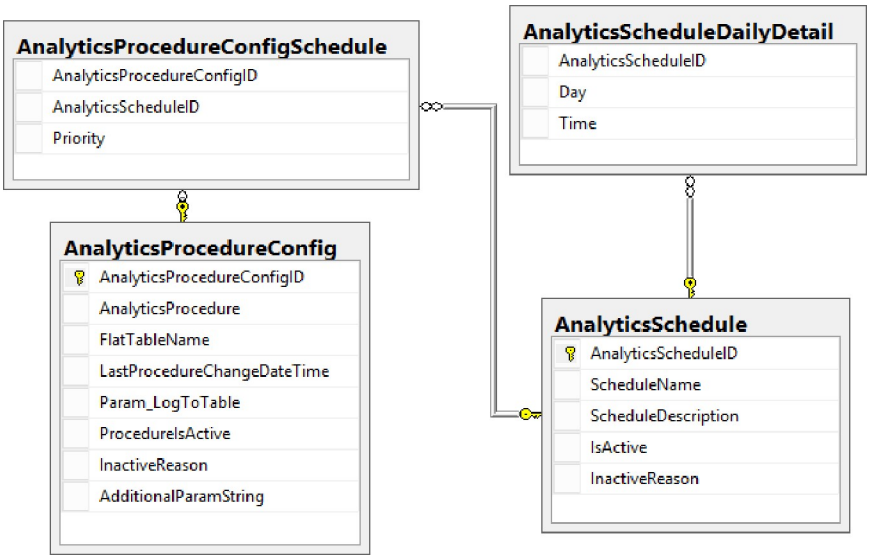
1. Create a schedule by adding a record to the AnalyticsSchedule table.
2. Create a job for the schedule by copying an existing Analytics – Run Flat Table Procedures job, and updating the @Schedule parameter passed into the Run Analytics Master step.
3. Configure schedules to run at the desired times by adding records to the AnalyticsScheduleDailyDetail table.
4. Link an Analytics sproc with one or more schedules by adding records to the AnalyticsProcedureConfigSchedule table.

Each schedule is associated with a job whose name begins with the word Analytics. It shouldn't be necessary to modify the schedules on these jobs themselves.

Components

Flat Table Caching Tables

This diagram shows the relationships between tables involved in flat table caching.



In the Control database:

- AnalyticsProcedureConfig table:

  Contains one row for each Analytics procedure and stores the associated flat table names. When an Analytics procedure runs, it creates a flat table with the FlatTableName contained in the AnalyticsProcedureConfig table, if that flat table does not already exist. This table also contains a ProcedureIsActive flag that allows you to activate or deactivate a sproc for all its associated schedules.

- AnalyticsSchedule table

  Contains one row for each schedule, and stores the name and description for that schedule. This table also contains an IsActive flag that activates or deactivates a schedule.

- AnalyticsScheduleDailyDetail table

  Contains configuration for the scheduled run for each day of the week. The Day values are integers between 0 (Sunday) and 6 (Saturday).

- AnalyticsProcedureConfigSchedule table

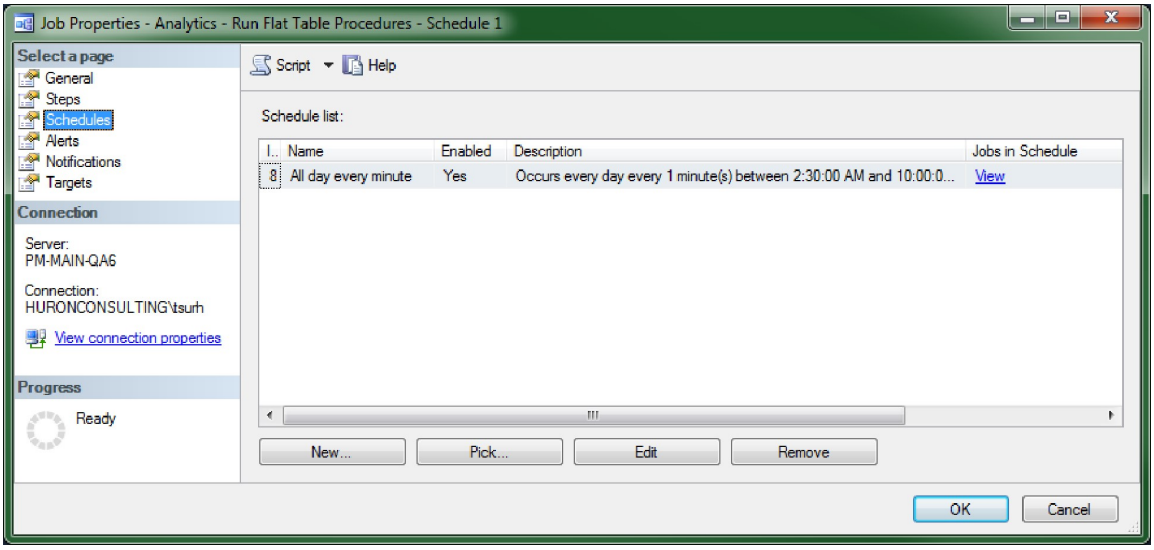
  This table links Analytics procedures with their schedules. If multiple sprocs are associated with the same schedule, they run sequentially, according to a configurable Priority field in the AnalyticsProcedureConfigSchedule table. You can configure multiple sprocs to run concurrently by associating each sproc with distinct, coinciding schedules.

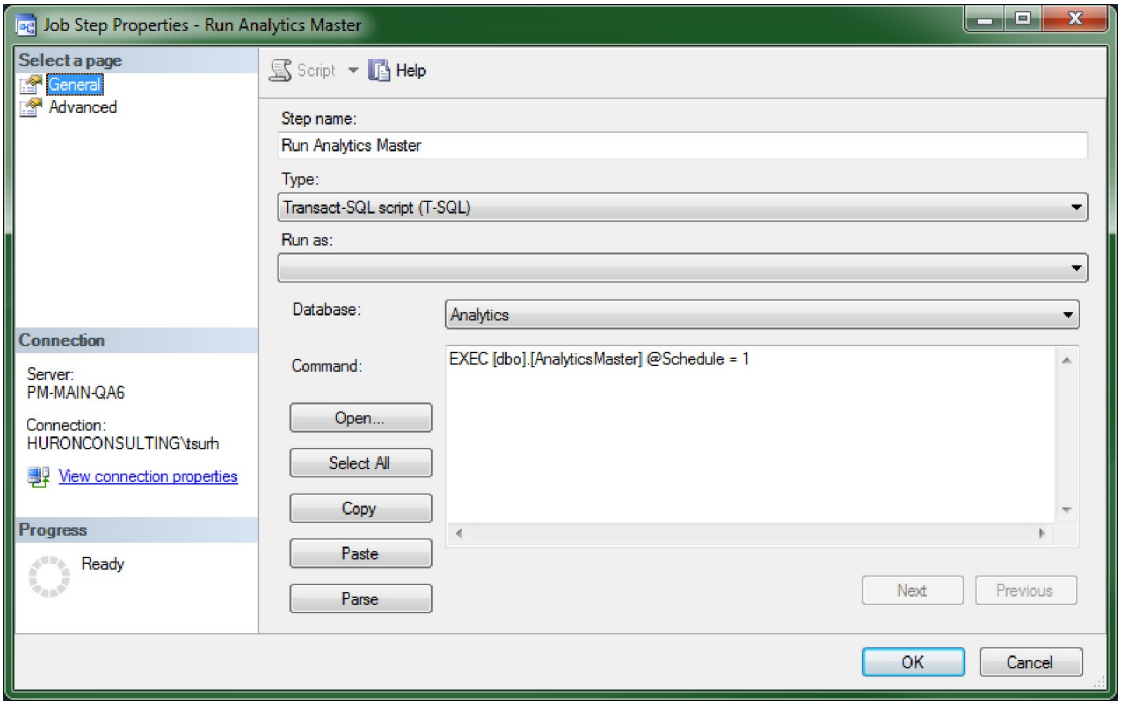
- SQL Agent jobs: Analytics – Run Flat Table Procedures

  On the Aeos database application server, runs flat table caching procedures linked to the schedule associated with the job. More specifically, the job will run any procedures that have not run since the scheduled time for that particular day. These jobs should be set to run every minute during the business day. Analytics scheduling should be adjusted by making updates to the config tables in the Analytics database. Do not modify the schedules for these jobs.

SQL Agent jobs: Analytics – Run Flat Table Procedures

On the Aeos database application server, runs flat table caching procedures linked to the schedule associated with the job. More specifically, the job will run any procedures that have not run since the scheduled time for that particular day. These jobs should be set to run every minute during the business day. Analytics scheduling should be adjusted by making updates to the config tables in the Analytics database. Do not modify the schedules for these jobs.





Process

- The Analytics – Run Flat Table Procedures jobs run every minute. They call the AnalyticsMaster sproc with a Schedule parameter matching their name. The AnalyticsMaster sproc checks whether there are any procedures linked to their particular schedule that have not run since the last scheduled time.
- Each procedure runs according to the job's schedule. It is executed from the AnalyticsMaster sproc in sequential order according to the Priority value in the AnalyticsProcedureConfig table.

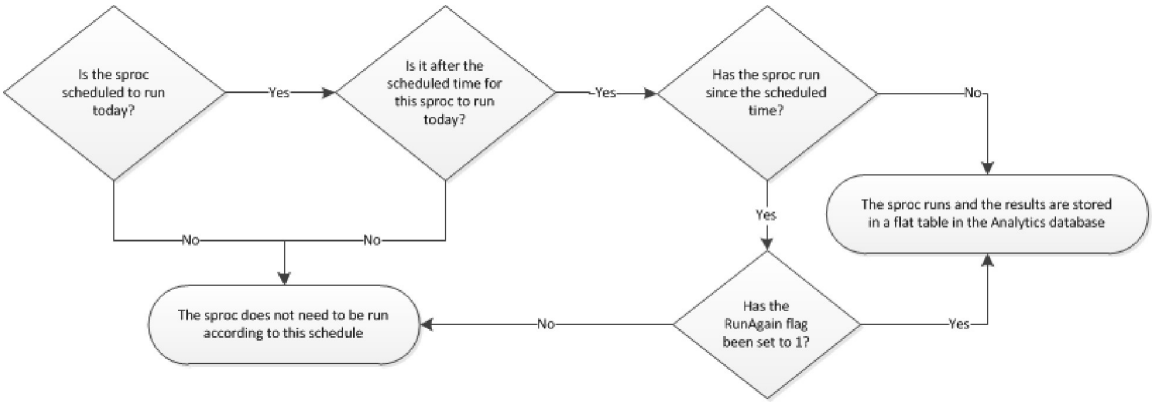
Default Settings

Job: enabled

Schedule: every day every minute from 2:30 AM to 10:00:00 PM

Analytics Schedule Check Process

When the Analytics – Run Flat Table Procedures jobs run, they check to see which analytics procedures are due to be run. The diagram below outlines how this check is performed.



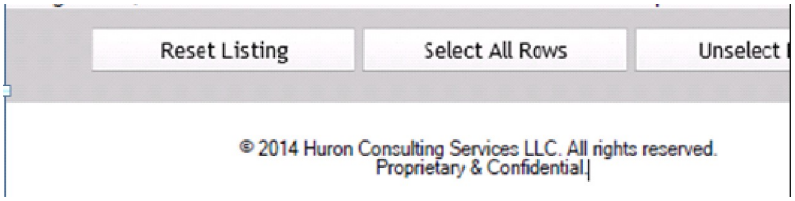
Configuring Analytics Copyright Information

The Analytics copyright information can be configured at the client site. It is displayed in the footer of all reports exported using 'Huron PDF export' tool, as well as on the HuronAbout.html page.

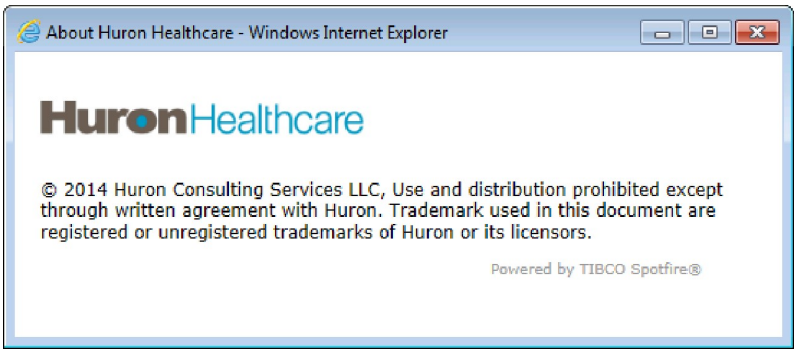
Important: Unless there are special circumstances at your client and management approval has been obtained, the copyright information should not be changed. It is updated in baseline when appropriate.

Screenshots of default copyright information:

- In the footer of an exported PDF report



- In the About.html box



Components

Analytics Setting Table

New in 16.1, there is an Analytics setting table: Control.dbo.AnalyticsSettings. Currently the only record it has is the copyright information:

ID	Name	Value	IsBaseline
1	Copyright	© 2014 Huron Consulting Services LLC. All rights reserved. Proprietary & Confidential.	1

*Note the “©” and “&” HTML codes represent the “©” and “&” symbols.

Spotfire Huron About and PDF Export Files

During the installation or upgrade and configuration process, the copyright information is copied from the AnalyticsSettings table to the following files for the TIBCO Spotfire application:

- C:\Program Files\TIBCO\Spotfire Web Player\6.5.2\webroot\Custom\HuronAbout.html
- C:\Program Files\TIBCO\Spotfire Web Player\6.5.2\webroot\bin\Modules\Huron.Branding_15.12.10.0\HuronAbout.html
- C:\Program Files\TIBCO\Spotfire Web Player\6.5.2\webroot\bin\Modules\Huron.Spotfire.PDFExportWeb_15.12.10.0\ HuronPDFExport.config’

Changing the Copyright Information

If you need to change the copyright information from its default, use the following procedure.

To customize the copyright information for Aeos Analytics

1. Update the Value field for the Copyright record in the AnalyticsSettings table.
2. Set the IsBaseline flag to 0 (false) so that the Huron Upgrader application knows not to overwrite this record with the new baseline value when this Aeos system is upgraded in the future.
3. Edit the ‘HuronAbout.html’ file to match the copyright value in the AnalyticsSetting table in the following locations, except use © and &, and not the HTML codes:
   - C:\Program Files\TIBCO\Spotfire Web Player\6.5.2\webroot\Custom
   - C:\Program Files\TIBCO\Spotfire Web Player\6.5.2\webroot\bin\Modules\Huron.Branding_15.12.10.0
4. Edit the ‘HuronPDFExport.config’ file (the same as HuronAbout.html) in the following location:

   C:\Program Files\TIBCO\Spotfire Web Player\6.5.2\webroot\bin\Modules\Huron.Spotfire.PDFExportWeb_15.12.10.0
5. Reset IIS on the Analytics Web Server.
6. Restart the TIBCO Spotfire Server Service.
   - On the Apps screen, click **Services** under Administrative Tools.
   - Right-click **TIBCO Spotfire Server 6.5.2** and select **Restart**.

Reference

For more details, see *PBI 87686 - Make Analytics Copyright Configurable.docx* in the Technical Documents/Aeos 16.1 folder in SharePoint.

For more information, see Legal Notices.

Spotfire Server Migration

You can migrate the Spotfire Server application (along with Spotfire thick client and automation services) from one server to another.

FAQ:

- Will the move retain bookmarks?

  Yes, as long as GUIDs have not been changed on reports.

- Will the move retain schedules for web caching?

  Yes.

- Will the move retain custom folders and templates?

  Yes.

- Will the move retain folder permissions?

  Yes.

- Will the move retain custom java or JVM specific settings implemented by PIT or the Analytics Team?

  Probably not, PIT/Analytics will likely have to make these adjustments again.

Process

The process involves first removing the current installation of Spotfire Server, Thick Client, and Automation Services (but leaving the Spotfire database intact!) and then utilizing the Aeos Spotfire Installer and Spotfire Configurator to re-install those components.

Testing Process

To verify the process retains the important Spotfire artifacts, some test items were created.

Bookmarks and Filters

Bookmarks were created in the Web Player by opening a report template (e.g. Account Listing Report), creating a filter (e.g. Patient Name or Hierarchy) and saving the bookmark.

Schedules

Schedules were adjusted in the Web Player by removing a schedule from one report (e.g. Write-Offs and Analysis report) and modifying the schedules of other reports to different date/times.

Folders and Permissions

To test folders, you must go into the Library Administrator from the Spotfire Thick Client. From there, a copy of the baseline Report Template was created (including bookmarks) into a new folder (e.g. Report Templates Copy). The permissions on the new folder was changed (e.g. only give Admin and Analytics Executive browse+access rights).

Remove Spotfire

To uninstall Spotfire in preparation for the move, follow these steps:

1. Go the server you're uninstalling from
2. Open **appwiz.cpl** (**Programs and Features** in the control panel)
3. **Uninstall** these items in the order shown:
   a. TIBCO Spotfire Automation Services
   b. TIBCO Spotfire

c. TIBCO Spotfire Server 6.5.2

4. Go to C:\TIBCO and remove the **tss** directory.
5. Go to C:\Program Files\TIBCO and remove the **Automation Services** directory
6. Go to C:\Program Files (x86) and remove the **TIBCO** directory

   CAUTION: Make sure you're in the **(x86)** Program Files and NOT the regular Program Files from the preceding step.
7. Go to windows services (**services.msc**) and verify the TIBCO Spotfire Server 6.5.2 service is no longer visible.

   If you're on a web server, you will see TIBCO Spotfire Web Player Keep Alive Service. This is ok.

Re-install Spotfire

To re-install Spotfire on the intended server, follow the process below by utilizing the Aeos Spotfire Installer and Configurator.

Installer.config

1. Open the Installer.config file in C:\Huron.Installer.Root\Config and modify the following areas:
2. Under Environment\Servers change the Server node for AnalyticsAppServers to the new server.

   Under Tokens, change these tokens to point to the new analytics server:
   a. HURON_SPOTFIRE_AUTH_SERVER
   b. HURON_SPOTFIRE_AUTHENTICATION_URL
   c. HURON_SPOTFIRE_SERVICE_API_URL
   d. HURON_SPOTFIRE_AUTOMATION_URL
3. Save the Installer.config file.

SpotfireInstaller

To re-install Spotfire components, simply make sure the binaries (MSI files) are in the correct locations (C:\Tibco\Binaries) on the new server then run the SpotfireInstaller.

SpotfireConfigurator

Once SpotfireInstaller has been run on the new server, run the SpotfireConfigurator on the new server and then on the web server(s). By running it on the web servers, we ensure all the config files for Spotfire are updated with the correct (new) server name.

Verification

After Spotfire components have been installed on the new server, go into Aeos and the Spotfire Web Player. Verify the bookmarks were kept intact and the caching schedules didn't change. On the Spotfire Server, open the thick client and verify the folders and their permissions also didn't change.