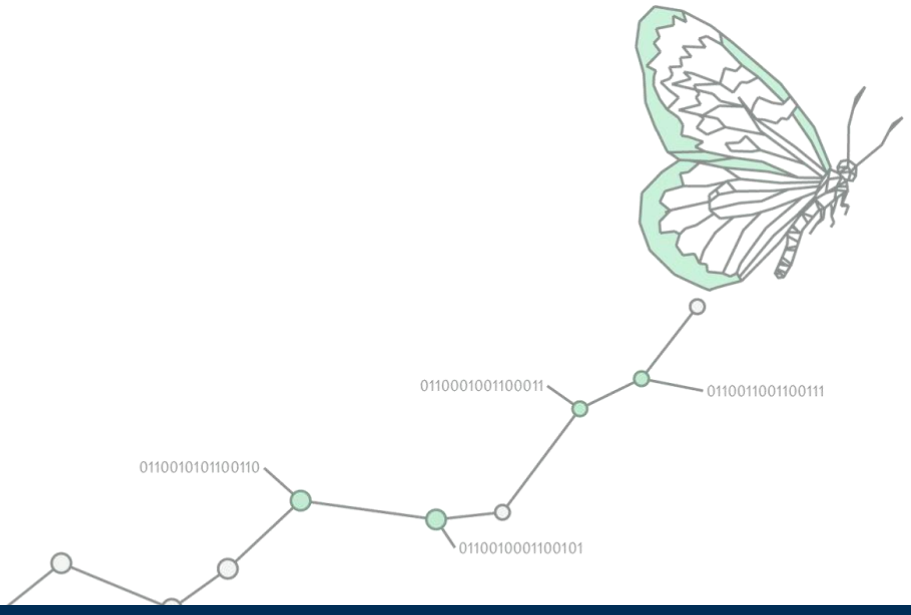
newcomp.com



NEWCOMP ANALYTICS CREATING A CHORE INFORMATION CUBE

Prepared by
Newcomp Analytics
www.newcomp.com
info@newomp.com

OVERVIEW

Chores in IBM Planning Analytics

Chores are a great way to automate the execution of TI processes in IBM Planning Analytics, but it can be hard to get visibility into their details once they are created. The steps below will guide you through the process of creating a Chore Information cube, creating TI processes to populate the cube from the existing .cho files, creating a new chore to ensure the cube is always up to date, and creating subsets to help you navigate the cube.

Creating a Chore Information cube

First we must create the cube so we have a place to store our data.

1. Create a "Chore Info Measures" dimension. It should have the following measures:
    a. Treated (simple)
    b. Chore Time (string)
    c. Next Run Time (string)
    d. Next Run Date (string)
    e. Relative Time Zone (string)
    f. Run Every (string)
    g. Number of Direct Processes (simple)
    h. Activated (simple)
    i. Commit Mode (string)
    j. Today (string)
2. Create a "Chore Info" cube with the following two dimensions:
    a. }Chores
    b. Chore Info Measures
3. Create a rule for the cube like this: ['Today'] = S: TODAY(1);

Creating the load TI processes

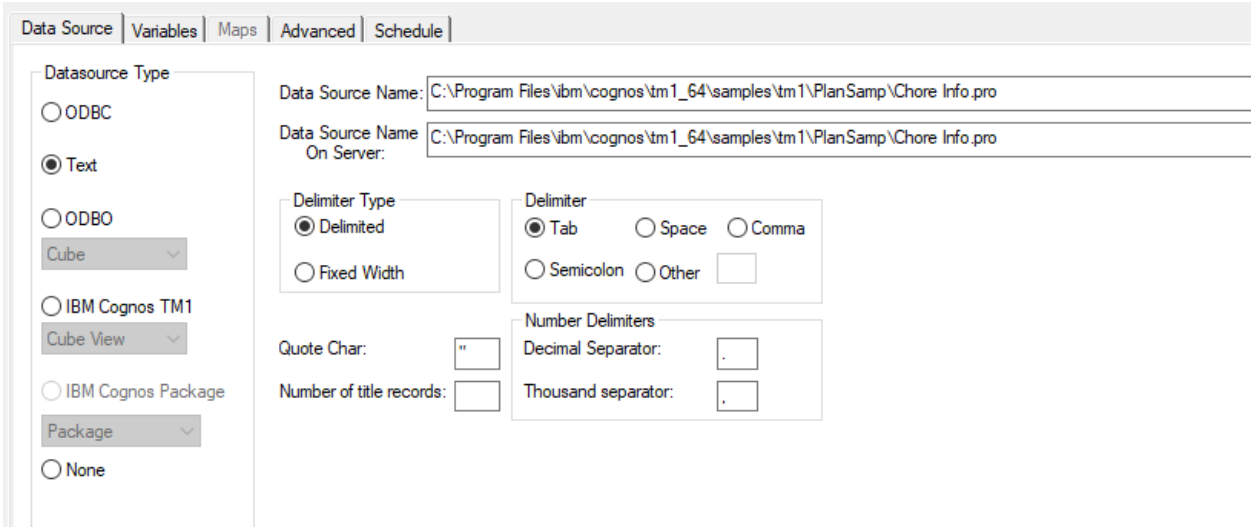
Next we must create some processes to load the data into the cubes.

1. Create a TI process called "Chore Info Parent" with the following code in the prolog tab:

```
i = 1;
WHILE(i <= DIMSIZ('}Chores'));
   thisChore = DIMNM('}Chores', i);
   ExecuteProcess('Chore Info', 'pChore', thisChore);
   i = i + 1;
END;
```

2. Create a TI process called "Chore Info". Set the data source to "Text", and navigate to a placeholder file with one column. I used the Chore Info.pro file (after first saving the process).





3. Name the variable vChoLine and set it to type String. Set Contents to Other.
4. Create a string parameter called pChore.
5. Paste the following code in the prolog tab:

```
DataSourceAsciiDelimiter = Char(1);
DataSourceAsciiQuoteCharacter = '';

# Constants

cChores_CubeName = 'Chore Info';
cChores_Dim1 = '}Chores';
cChores_Dim2 = '}Processes';

# Set datasource

IF(FileExists( pChore | '.CHO') = 0);
   vErrorMessage = 'The CHO file for the chore called ''' |
pChore | ''' could not be found.';
   ProcessQuit;
ELSE;
   DatasourceNameForServer = pChore | '.CHO';
   vNextRun = 0;
   vNrOfProcesses = 0;
   ndf = NewDateFormatter( '' );
   CellPutN(1, cChores_CubeName, pChore, 'Treated');
ENDIF;
```

6. Paste the following code in the data tab:

```
vChoLine = Trim(vChoLine);

# Add Chore Time

IF(Scan('530,', vChoLine) = 1);
   vChoLineValue = Delet(vChoLine, 1, 4);
```

```
   CellPutS(vChoLineValue, cChores_CubeName, pChore, 'Chore
Time');
   vRunTime = ParseDate( vChoLineValue, 'yyyyMMddHHmmss', ndf );
   Itemskip;

# Add Relative Time Zone

ELSEIF(Scan('537,', vChoLine) = 1);
   vChoLineValue = Delet(vChoLine, 1, 4);
   IF( vChoLineValue @= '0');
      CellPutS('UTC Time', cChores_CubeName, pChore, 'Relative
Time Zone');
   ELSEIF( vChoLineValue @= '1');
      CellPutS('Local Server Time', cChores_CubeName, pChore,
'Relative Time Zone');
   ENDIF;
   Itemskip;

# Subtract 0 hours if UTC Time and 5 hours if Local Server Time

ELSEIF(Scan('531,', vChoLine) = 1);
   IF( CellGetS( cChores_CubeName, pChore, 'Relative Time Zone')
@= 'UTC Time' );
      vRunTimeOffset = 0/24;
   ELSE;
      vRunTimeOffset = -5/24;
   ENDIF;
   vRunTime = vRunTime + vRunTimeOffset;

   # Add Run Every

   vChoLineValue = Delet(vChoLine, 1, 4);
   vDay = SUBST(vChoLineValue, 1, 3);
   vHr = SUBST(vChoLineValue, 4, 2);
   vMin = SUBST(vChoLineValue, 6, 2);
   vSec = SUBST(vChoLineValue, 8, 2);
   CellPutS(vDay | ' Day, ' | vHr | ' Hr, ' | vMin | ' Min, ' |
vSec | ' Sec', cChores_CubeName, pChore, 'Run Every');

   # Add Next Run Time

   vInterval =
             Numbr( Subst( vChoLineValue, 1, 3 ))
          + Numbr( Subst( vChoLineValue, 4, 2 )) / 24
          + Numbr( Subst( vChoLineValue, 6, 2 )) / 24 / 60
          + Numbr( Subst( vChoLineValue, 8, 2 )) / 24 / 60 / 60;
   IF( vRunTime < Now );
      vNextRun = vRunTime - ( Int( - ( Now - vRunTime ) /
vInterval )) * vInterval;
   ELSE;
      vNextRun = vRunTime;
   ENDIF;
   IF( vNextRun <> 0 );
      vNextRun_Formatted = FormatDate( vNextRun, 'yyyy-MM-dd
HH:mm:ss EEE', ndf );
```



```
      CellPutS( vNextRun_Formatted, cChores_CubeName, pChore,
'Next Run Time');
      vNextRun_FormattedDate = FormatDate( vNextRun, 'yyyy-MM-
dd', ndf );
      CellPutS( vNextRun_FormattedDate, cChores_CubeName, pChore,
'Next Run Date');
   ENDIF;
   Itemskip;

# Add Number of Direct Processes

ELSEIF(Scan('532,', vChoLine) = 1);
   vChoLineValue = Delet(vChoLine, 1, 4);
   vNrOfProcesses_OfChore = StringToNumber(vChoLineValue);
   CellPutN(vNrOfProcesses_OfChore, cChores_CubeName, pChore,
'Number of Direct Processes');
   Itemskip;

# Add Activated

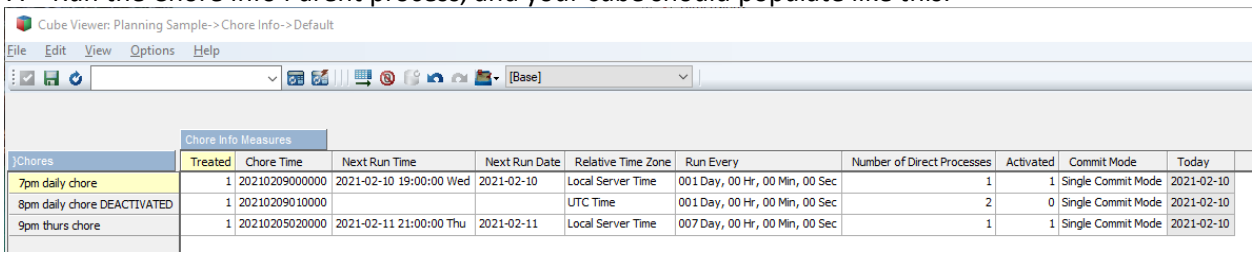
ELSEIF(Scan('533,', vChoLine) = 1);
   vChoLineValue = Delet(vChoLine, 1, 4);
   CellPutN(StringToNumber(vChoLineValue), cChores_CubeName,
pChore, 'Activated');
   IF( StringToNumber(vChoLineValue) = 0 );
      CellPutS( '', cChores_CubeName, pChore, 'Next Run Time');
      CellPutS( '', cChores_CubeName, pChore, 'Next Run Date');
   ENDIF;
   Itemskip;

# Add Commit Mode

ELSEIF(Scan('1405,', vChoLine) = 1);
   vChoLineValue = Delet(vChoLine, 1, 5);
   IF( vChoLineValue @= '0');
      CellPutS('Single Commit Mode', cChores_CubeName, pChore,
'Commit Mode');
   ELSEIF( vChoLineValue @= '1');
      CellPutS('Multiple Commit Mode', cChores_CubeName, pChore,
'Commit Mode');
   ENDIF;
   Itemskip;

ENDIF;
```

7. Run the Chore Info Parent process, and your cube should populate like this:



Creating a new chore

Now we will create a chore so our Chore Information cube is updated daily.

1. Create a new chore and have it run “Chore Info Parent”. Pick a time for it to run (in this example 6:00AM was chosen), and set it to run every day. Save the chore (in this example it is called
“6am daily chore info update”). Activate the chore.
2. Optionally run the Chore Info Parent process again to populate this new chore with data in your cube.

Creating subsets

Finally, we will create two subsets to help us navigate our cube more easily.

1. Open the subset editor for the }Chores dimension and paste the following MDX into the expression window. Save the subset as “Running Today” and ensure that “Save Expression” is selected. Save again, and when prompted to overwrite, select “Yes”. This is a list of all of the chores that will run today.

```
{FILTER({TM1FILTERBYLEVEL( {TM1SUBSETALL( [}Chores] )}, 0)},
[Chore Info].([Chore Info Measures].[Next Run Date]) @= [Chore Info].([Chore
Info Measures].[Today]) )}
```

2. Open the subset editor for the }Chores dimension and paste the following MDX into the expression window. Save the subset as “Upcoming” and ensure that “Save Expression” is selected. Save again, and when prompted to overwrite, select “Yes”. This is a list of all upcoming scheduled chores, in the order they will be executed.

```
INTERSECT(
{ORDER({ TM1FILTERBYLEVEL({TM1SUBSETALL( [}Chores] )} ,0)}, [Chore
Info].([Chore Info Measures].[Next Run Time]), ASC)},
{FILTER({TM1FILTERBYLEVEL( {TM1SUBSETALL( [}Chores] )}, 0)},[Chore
Info].([Chore Info Measures].[Next Run Time]) @<> "" )}
)
```

Notes

Portions of “Chore Info” code inspired by “TECH_TM1_Documenter_ChoreInfo” by Wim Gielis.
https://www.tm1forum.com/viewtopic.php?f=21&t=13373

