

# Hands-On Lab Instructions Advanced Calculation for Forecasting & Driver Based Budgeting



## Table of Contents

<b>USER REQUIREMENTS.....</b>	<b>2</b>
Regression Forecasts for Products 4 and 5.	2
Penetration Curve (S-Curve) Forecasting	2
Pipeline and Burn Rate Forecasting	2
Forecasting HR Load	2
<b>INSTRUCTIONS.....</b>	<b>3</b>
<b>Revenue Forecasting</b>	<b>3</b>
Regression Forecasting	3
Penetration Curve Forecasting	3
Sales Pipeline and Burn Rate Forecasting	4
<b>Forecasting HR Load</b>	<b>4</b>
<b>CALCULATION SCRIPT CODE SAMPLES .....</b>	<b>5</b>
<b>Revenue Forecasting</b>	<b>5</b>
Regression Forecasting	5
Penetration Curve Forecasting	7
Pipeline and Burn Rate Forecasting	11
<b>HR Load Forecasting</b>	<b>13</b>
Spread Gross Pay by Month	13
401k and SSI Deductions	14
Step Functions for Federal Tax Withholding	15
Step Functions State Tax Withholding	17
Step Functions City Tax Withholding	19
<b>Set Up</b>	<b>20</b>
Essbase Databases;	20
Data Files: (all free form – no rules required)	20
<b>ABOUT THE AUTHOR.....</b>	<b>21</b>

## **User Requirements**

### ***Regression Forecasts for Products 4 and 5.***

Products 4 and 5 have historical actual data back to Jan 2004.

- In order to easily calculate the Total Year LatestEst value copy Actuals to LatestEst.
- Produce a regression based on a time trend for Product 4.
- Produce a regression based on planned marketing expenses for product 5. Store the forecasts in the LatestEst scenario. .

### ***Penetration Curve (S-Curve) Forecasting***

Products 6 and 7 are new products and have no historical actual data. However, Marketing estimates that these products will eventually grow to 10000 in sales each. In addition Marketing estimates growth in 3 stages: introduction (7 months), rapid growth (months 8 through 20), slower growth (remaining months).

- Produce penetration curve forecasts for products 6 and 7.
- Use substitution variables to provide flexible input parameters for the saturation point, and end of introduction and rapid growth stages.

### ***Pipeline and Burn Rate Forecasting***

The sales department is working on several specific deals for products 8, 9 and 10. The salespeople report expected close dates, amounts and probabilities for the deals they are working on each week in their pipeline reports. Once the deals are closed services are delivered in predictable increments over the following months according to percentages stored in the curves database.

- Produce forecasts for products 8, 9 and 10 based on the pipeline reports and the delivery curves.

### ***Forecasting HR Load***

Employee specific information such as base salary and 401k withholding elections are stored in the hrEmplye database. Global parameters such as tax rates are stored in the hrInput database.

- Produce HR load forecasts by month by employee and store them in the HR database.

# Instructions

## *Revenue Forecasting*

### Regression Forecasting

- Open the Excel file c:\MTGClass\Blakbelt\revforecast.xls and select the regression sheet
- Load revinput.txt no rule is required
- Run CpyAct.csc
- Lock and send planned marketing expenses using the marketing plan input sheet
- Setup substitution variables

Substitution Variable	Value
BegFC	"2009"->"Jan"
BegHist	"2004"->"Jan"
EndFC	"2011"->"Dec"
EndHist	"2008"->"Dec"
FCStartMonth	"Jan"
FCStartYear	"2009"

- Run Regressn.csc
- Review the results in the regression sheet

### Penetration Curve Forecasting

- In the revforecast.xls workbook switch to the SCurve sheet
- Lock the block using Essbase | lock. For product P06 enter the numbers 1 through 36 to the periods January 2009 through December 2011. Send the data using Essbase | send
- Setup substitution variables

Substitution Variable	Value
SCrvHG	7
SCrvSat	10000
SCrvTO	20

- Run SCurve.csc
- Review the results in the S-curve sheet

## Sales Pipeline and Burn Rate Forecasting

- In the revforecast.xls workbook switch to the pipeline input sheet. Lock and send the data. Ignore the warning messages caused by a column labels.
- Switch to the pipeline curves input sheet. Connect to the BlakBelt.curves database. Lock and send the data
- Switch to the pipeline sheet
- Set up the following location alias

Location Alias	From	Points to
xCurves	Rev	Curves

- Run pipeline.csc
- Review the results in the pipeline sheet

## Forecasting HR Load

- Open HR.xls workbook and switch to the SpreadGrossPay sheet
- Connect to BlakBelt.HR database
- Switch to the EmplInput sheet, connect to BlakBelt.hrEmplye database and lock and send the data
- Switch to the HR input sheet, connect to the BlakBelt.hrInput database and lock and send the data
- Create the following location aliases

Location Alias	From	Points to
xHRInput	HR	hrInput
xEmplInput	HR	hrEmplye

- Switch to the spread gross pay sheet
- Run each of the following scripts. Examine the results after each script.

Script	Purpose
1SprGrss.csc	Spread gross pay by month
2OtherWh.csc	Calculate 401(k) and Social Security withholdings
3FedWh	Calculate federal tax withholdings
4NYSWh	Calculate State withholdings
5NYCWh	Calculate city withholdings

## Calculation Script Code Samples

### *Revenue Forecasting*

```
/* COPY ACTUALS FOR ALL PRODUCTS AND CUSTOMERS TO LATESTEST */
```

```
Fix("2004":"2008")
    CLEARDATA "LatestEst";
    DATACOPY "Actual"TO "LatestEst";
Endfix
```

### **Regression Forecasting**

```
/****** Marketing Technologies Group -- Ron Moore *****/
/****** Kaleidoscope 2009 Hand-on Lab *****/

/****** Regression Forecast *****/
/*Regression1 uses time as independant variable */
/*Regression2 uses "independantVar" as independant variable */

/****** REQUIRES *****/
/**** Substitution Variables
    BegHist->"Sales"->"LatestEst" - Year->Month
    EndHist->"Sales"->"LatestEst" - Year->Month
    BegFC->"Sales"->"LatestEst" - Year->Month
    EndFC->"Sales"->"LatestEst" - Year->Month

*/

/* @TREND (Ylist,
           [Xlist],
           [weightList], [errorList],
           [XforecastList],
           YforecastList,
```

```

                                method
                                [, methodparameters] [, Xfilter1 [, parameters]] [, XfilterN [, parameters]][,
Yfilter1 [, parameters]] [, YfilterN [, parameters]]) */

/***** Simple regression dependant on time *****/

FIX("LatestEst",@UDA ("AllPrd", "Regression1"))
    "Sales"(
        @TREND (
            @xrange(&BegHist->"Sales"->"LatestEst", &EndHist->"Sales"->"LatestEst") /*
dependant variable (sales) history */
            '****'
            @xrange(&BegFC->"Sales"->"LatestEst", &EndFC->"Sales"->"LatestEst"), /*
dependant variable sales forecast */
            LR ); /*
method = Linear regression */
        );
ENDFIX

/***** Independant variable based regression *****/

Fix ("LatestEst",@UDA ("AllPrd", "Regression2"))
    "Sales"(
        @TREND (
            @xrange("Sales"->"LatestEst"->&BegHist, "Sales"->"LatestEst"->&EndHist),
/* YList - dependant variable (sales) history */
            @XRANGE("Marketing"->"LatestEst"->&BegHist, "Marketing"->"LatestEst"-
>&EndHist) /* XList - Dependant Variable ( Marketing) history*/
            '***'
            @XRANGE("Marketing"->"LatestEst"->&BegFC, "Marketing"->"LatestEst"->&EndFC ),
/* XList - Dependant Variable ( Marketing) future */
            @xrange("Sales"->"LatestEst"->&BegFC , "Sales"->"LatestEst"->&EndFC), /*
YForecast - dependant variable sales forecast */
            LR ); /*
method = Linear regression */
        );
ENDFIX

```

## Penetration Curve Forecasting

```
/****** Marketing Technologies Group -- Ron Moore *****/
/****** Kaleidoscope 2009 Hand-on Lab *****/

/* use VARs to get parameters for s-curve forecast */
/*** Four alternative approaches to providing the time period parameters *****/
/*1. Simple Calc: Requires stored input period numbers */
/*2. Calculate periods using a VAR */
/*3. Calculate period in-line */
/*4. Calculate period using a custom function */

/* 1. Simple Calc: Requires stored input period numbers *****/
/* =saturation/(1+81^((hypergrowth+takeover/2-year)/takeover)) */

SET UPDATECALC OFF;

Var Sat =&SCrvSat; /* Upper limit */
Var HG = &SCrvHG; /* Acceleration break point period number */
Var TOvr= &SCrvTO; /* Deceleration break point period number */
Var cPeriod; /* calculated period */

FIX("c01","P06","LatestEst")
    "sales" =@ROUND(
        sat/(1+@power(81, (((HG+TOvr/2)-"Period")/TOvr))),0);

ENDFIX
```

```

/* 2. Calculate Period using a VAR to reduce maintenance *****/

SET CREATENONMISSINGBLK ON;
FIX("c01", "P07", "LatestEst", @IRSIBLINGS (&FCStartYear))
"Sales" (
cPeriod=@count(SKIPNONE,
                @XRANGE(
                    &BegFC,
                    @CurrMBR("All Years")->@CurrMbr("Year Total")
                )
);

"sales" =@ROUND(
                sat/(1+@power(81, ((HG+TOvr/2)-cPeriod)/TOvr)),
                0);
)
ENDFIX

```

```

/* 3. Calculate Period inline to reduce maintenance *****/
/* SET CREATENONMISSINGBLK ON;
   FIX("c01", &SCrvProduct, "LatestEst", @IRSIBLINGS (&FCStartYear)
   "Sales" (
     "sales" =@ROUND(
       sat/(1+@power(81, (((HG+TOvr/2)
@count(SKIPNONE,
       @XRANGE (
         &FCStartYear->&FCStartMonth,
         @CurrMBR("All Years")->@CurrMbr("Year Total")
       )
     )
       )/TOvr))),
0);
)
ENDFIX

```

```

/* 4. Calculate period using a custom macro *****/

/*FIX("c01",
    &SCrvProduct,
    "LatestEst",
    @IRSIBLINGS(&FCStartYear)
)

    "sales" =@ROUND(
                                sat/(1+@power(81, ((HG+TOvr/2)- @bbPeriod(&FCStartYear,&FCStartMonth)
) /TOvr))),
                                0);
ENDFIX */
/* Custom macro Max1 Script*****/
/*
create or replace macro 'BlakBell'.'@bbPeriod'
(SINGLE, SINGLE) as
'@count(SKIPNONE,@XRANGE(@@1->@@2,@CurrMBR("All Years")->@CurrMbr("Year Total")))'
spec '@bbPeriod(fcstartyear,fcstartmonth)'

*/

```

## Pipeline and Burn Rate Forecasting

```
//ESS_LOCALE English_UnitedStates.Latin1@Binary
```

```
/****** Marketing Technologies Group -- Ron Moore *****/
```

```
/****** Kaleidoscope 2009 Hand-on Lab *****/
```

```
/* calculate SalesClosed (ExpVal) based on probabilities */
```

```
/* Spread sales closed into sales using predefined curve percentages xref'd from Blakbelt.curves */
```

```
/* requires location alias xCurves */
```

```
FIX("LatestEst")
```

```
/******
```

```
/* calc SalesClosed
```

```
*/
```

```
"SalesClosed"="Amount"*"Prob";
```

```
/******
```

```
/* Spread SalesClosed
```

```
*/
```

```
set updatecalc off;
```

```
VAR vFP1;
```

```
VAR vFP2;
```

```
VAR vFP3;
```

```
VAR vFP4;
```

```
VAR vFP5;
```

```
VAR vFP6;
```

```
Fix(@uda("AllPrd", "Pipeline"))
```

```
  "Sales" (
```

```
    vFP1=@XREF(xCurves, "Crv1", "Per01");
```

```
    vFP2=@XREF(xCurves, "Crv1", "Per02");
```

```
    vFP3=@XREF(xCurves, "Crv1", "Per03");
```

```
    vFP4=@XREF(xCurves, "Crv1", "Per04");
```

```
    vFP5=@XREF(xCurves, "Crv1", "Per05");
```

```
    vFP6=@XREF(xCurves, "Crv1", "Per06");
```

```
  )
```

```
  "Sales" (
```

```
    "sales"="SalesClosed"*vFP1
        + @Prior("SalesClosed",1) *vFP2
        + @Prior("SalesClosed",2)*vFP3
        + @Prior("SalesClosed",3)*vFP4
        + @Prior("SalesClosed",4)*vFP5
        + @Prior("SalesClosed",5)*vFP6;
    )

    endfix /* release fix on pipeline type products */
Endfix /* Release baseline fix on Latest Estimate */
//ESS_LOCALE English_UnitedStates.Latin1@Binary
```

## ***HR Load Forecasting***

### **Spread Gross Pay by Month**

```
/******Spread Gross Pay ******/  
/* requires input from hrEmplye cube */  
  
SET UPDATECALC OFF;  
SET CREATENONMISSINGBLK ON;  
  
Fix(@levmbrs("Year Total",0),@levmbrs("AllHeads",0))  
    "Base"=@xref(xEmpInput,"Base")/12;  
ENDFIX
```

## 401k and SSI Deductions

```
set updatecalc off;
VAR xSSIMax,vAccumTGI,vAccumSSIwh;

Fix(@levmbrs ("AllHeads",0), "2009":"2011")

    "401kwh"="GrossPay"*@xref(xEmpInput,"401kwh");
    "401KMatch"="GrossPay"*@xref(xhrinput,"401k Match",xRate);
    "Medicarewh"="TaxableGross"*@xref(xhrinput,"Medicarewh",xRate);
    "MedicareEmplyr"="TaxableGross"*@xref(xhrinput,"Medicarewh",xRate);

    "SSIwh" (
        xSSIMax=@xref(xHRInput,SSIwh,BaseTax);
        vAccumSSIwh=@SUMRANGE ("SSIwh", @CURRMBRRANGE ("Year Total", LEV,0,-12,-1));
        @If(vAccumSSIwh<xSSIMax)
            "SSIwh"=@MIN ("TaxableGross"*@xref(xHRInput,SSIwh,FedRate),xSSIMax-vAccumSSIwh);
            "SSIEmplyr"="SSIwh";
        ELSE
            "SSIwh"=xSSIMax- vAccumSSIwh;
            "SSIEmplyr"= "SSIwh";
        ENDIF;
    )
endfix

CALC DIM ("AllHeads");
```

## Step Functions for Federal Tax Withholding

```
/* calculates FedWH using rates from HRInput XREF Cube */
/* REQUIRES Location Alias xHRInput */
/* assumes monthly matching of rates to gross pay.

set updatecalc Off;
Var vCap1,vCap2,vCap3,vCap4,vCap5,vCap6;
Var ProjAnGross;
Fix (@levmbrs ("AllHeads",0), "2009":"2011")

"FedWH" (
    ProjAnGross = "TaxableGross"*12;
    vCap1=@XREF (xHRInput, Fed1, Cap);
    vCap2=@XREF (xHRInput, Fed2, Cap);
    vCap3=@XREF (xHRInput, Fed3, Cap);
    vCap4=@XREF (xHRInput, Fed4, Cap);
    vCap5=@XREF (xHRInput, Fed5, Cap);
    vCap6=@XREF (xHRInput, Fed6, Cap);

    If (ProjAnGross>vCap6)
        "FedWH"= (@xref (xHRInput, Fed6, BaseTax) + (@xref (xHRInput, Fed6, Fedrate) * (ProjAnGross-
vCap6))) /12;
    ElseIf (ProjAnGross>vCap5)
        "FedWH"= (@xref (xHRInput, Fed5, BaseTax) + (@xref (xHRInput, Fed5, Fedrate) * (ProjAnGross-
vCap5))) /12;
    ElseIf (ProjAnGross>vCap4)
        "FedWH"= (@xref (xHRInput, Fed4, BaseTax) + (@xref (xHRInput, Fed4, Fedrate) * (ProjAnGross-
vCap4))) /12;
    ElseIf (ProjAnGross>vCap3)
        "FedWH"= (@xref (xHRInput, Fed3, BaseTax) + (@xref (xHRInput, Fed3, Fedrate) * (ProjAnGross-vCap3))) /12
;
    ElseIf (ProjAnGross>vCap2)
        "FedWH"= (@xref (xHRInput, Fed2, BaseTax) + (@xref (xHRInput, Fed2, Fedrate) * (ProjAnGross-
vCap2))) /12;
    ElseIf (ProjAnGross>vCap1)
```

```
        "FedWH"=(@xref(xHRInput,Fed1,BaseTax) + (@xref(xHRInput,Fed1,Fedrate)*(ProjAnGross-
vCap1)))/12;
        endif )
endfix
CALC DIM ("AllHeads");
```

## Step Functions State Tax Withholding

```
/* calculates StateWH using rates from HRInput XREF Cube */
/* Requires Location Alias xHRInput */

/* assumes monthly matching of rates to gross pay.

set updatecalc Off;
Var vCap1,vCap2,vCap3,vCap4,vCap5,vCap6,vCap7,vCap8;
Var ProjAnGross;
Fix(@levmbrs("AllHeads",0))

"StateWH" (
    ProjAnGross = "TaxableGross"*12;
    vCap1=@XREF(xHRInput,State1,Cap);
    vCap2=@XREF(xHRInput,State2,Cap);
    vCap3=@XREF(xHRInput,State3,Cap);
    vCap4=@XREF(xHRInput,State4,Cap);
    vCap5=@XREF(xHRInput,State5,Cap);
    vCap6=@XREF(xHRInput,State6,Cap);
    vCap7=@XREF(xHRInput,State7,Cap);
    vCap8=@XREF(xHRInput,State8,Cap);

    If(ProjAnGross>vCap8)
        "StateWH"=(@xref(xHRInput,State8,BaseTax) + (@xref(xHRInput,State8,xRate) * (ProjAnGross-
vCap8)))/12;
    ElseIf(ProjAnGross>vCap7)
        "StateWH"=(@xref(xHRInput,State7,BaseTax) + (@xref(xHRInput,State7,xRate) * (ProjAnGross-
vCap7)))/12;
    ElseIf(ProjAnGross>vCap6)
        "StateWH"= (@xref(xHRInput,State6,BaseTax) + (@xref(xHRInput,State6,xRate) * (ProjAnGross-
vCap6)))/12;
    ElseIf(ProjAnGross>vCap5)
        "StateWH"= (@xref(xHRInput,State5,BaseTax) + (@xref(xHRInput,State5,xRate) * (ProjAnGross-
vCap5)))/12;
    ElseIf(ProjAnGross>vCap4)
        "StateWH"= (@xref(xHRInput,State4,BaseTax) + (@xref(xHRInput,State4,xRate) * (ProjAnGross-
vCap4)))/12;
```

```

        ElseIf (ProjAnGross>vCap3)
            "StateWH"= (@xref (xHRInput, State3, BaseTax) + (@xref (xHRInput, State3, xRate) * (ProjAnGross-
vCap3))) /12;
        ElseIf (ProjAnGross>vCap2)
            "StateWH"= (@xref (xHRInput, State2, BaseTax) + (@xref (xHRInput, State2, xRate) * (ProjAnGross-
vCap2))) /12;
        ElseIf (ProjAnGross>vCap1)
            "StateWH"= (@xref (xHRInput, State1, BaseTax) + (@xref (xHRInput, State1, xRate) * (ProjAnGross-
vCap1))) /12;
        Endif )

endfix

CALC DIM ("AllHeads");

```

## Step Functions City Tax Withholding

```
/* calculates CityWH usinf rates from HRInput XREF Cube */
/* Requires Location Alias xHRInput */

/* assumes monthly matching of rates to gross pay.

set updatecalc Off;
Var vCap1,vCap2,vCap3,vCap4;
Var ProjAnGross;
Fix(@levmbrs("AllHeads",0))

"CityWH" (
    ProjAnGross = "TaxableGross"*12;
    vCap1=@XREF(xHRInput,City1,Cap);
    vCap2=@XREF(xHRInput,City2,Cap);
    vCap3=@XREF(xHRInput,City3,Cap);
    vCap4=@XREF(xHRInput,City4,Cap);

    If(ProjAnGross>vCap4)
        "CityWH"= (@xref(xHRInput,City4,BaseTax) + (@xref(xHRInput,City4,xRate) * (ProjAnGross-
vCap4)))/12;
    ElseIf(ProjAnGross>vCap3)
        "CityWH"= (@xref(xHRInput,City3,BaseTax) + (@xref(xHRInput,City3,xRate) * (ProjAnGross-
vCap3)))/12;
    ElseIf(ProjAnGross>vCap2)
        "CityWH"= (@xref(xHRInput,City2,BaseTax) + (@xref(xHRInput,City2,xRate) * (ProjAnGross-
vCap2)))/12;
    ElseIf(ProjAnGross>vCap1)
        "CityWH"= (@xref(xHRInput,City1,BaseTax) + (@xref(xHRInput,City1,xRate) * (ProjAnGross-
vCap1)))/12;
    Endif )
Endfix

CALC DIM ("AllHeads");
```

## ***Set Up***

The MaxL script `SetUpBlakBelt.mxl` can be used to build the Essbase databases and load the data. Please note that the Essbase path and login information are defaults. Substitute the appropriate path. `SetUpBlakBelt.bat` can be used to run the MaxL script. Please note that the batch file uses *admin* as the username and *password* as the password. Substitute your own username and password.

To set up the BlakBelt application and databases:

- Edit login and path information in the `SetUpBlakBelt.mxl` and `SetUpBlakBelt.bat` files
- Execute `SetUpBlakBelt.bat`
- Verify that the following databases are created under the BlakBelt application

Note: The instructions require you to create substitution variables and location aliases. However, for convenience, the MaxL script sets them up automatically.

### **Essbase Databases;**

- BlakBelt.Rev
- BlakBelt.Curves
- BlakBelt.HR
- BlakBelt.hrInput
- BlakBelt.hrEmplye

### **Data Files: (all free form – no rules required)**

- RevInput.txt
- hrInput.txt
- hrEmplyeInput.txt
- CurvesInput.txt

## About the Author

**Ron Moore** was one of the first Essbase certified consultants in the world. He founded Marketing Technologies Group which specializes in Essbase training, consulting and collaborative development. MTG's Oracle authorized training facility is located in New York City's Wall Street district. Ron is known for his ability reveal the mysteries of Essbase using real world examples from his more than 14 years of hands-on field experience.

Prior to founding MTG Ron worked for Manageware, developers of Compete!, one of the earliest desktop OLAP tools, and for The Alcar Group who developed software for shareholder value and strategic financial planning. He also worked for the Futures Group, a futurist consulting firm and he also taught college economics. He holds Bachelor and Master Degrees in Economics from Boston University.

**About Marketing Technologies Group:** MTG is an industry leading provider of OLAP and Business Intelligence consulting and training services, as well as one of the oldest and most technically certified Hyperion Essbase specialty firms on the planet. We are experts in multidimensional financial modeling and have helped hundreds of Fortune 500 companies improve their business performance by making the most of their critical data. MTG is widely recognized for its technical achievement, customer focus, and the highest standards of quality and design.

### Contact us at:

Marketing Technologies Group  
55 Broad Street, 10<sup>th</sup> Floor  
New York City, NY 10004

Phone: 212-824-2900

Fax: 212-824-4929

Web: [www.mtgny.com](http://www.mtgny.com)

Email: [info@mtgny.com](mailto:info@mtgny.com)