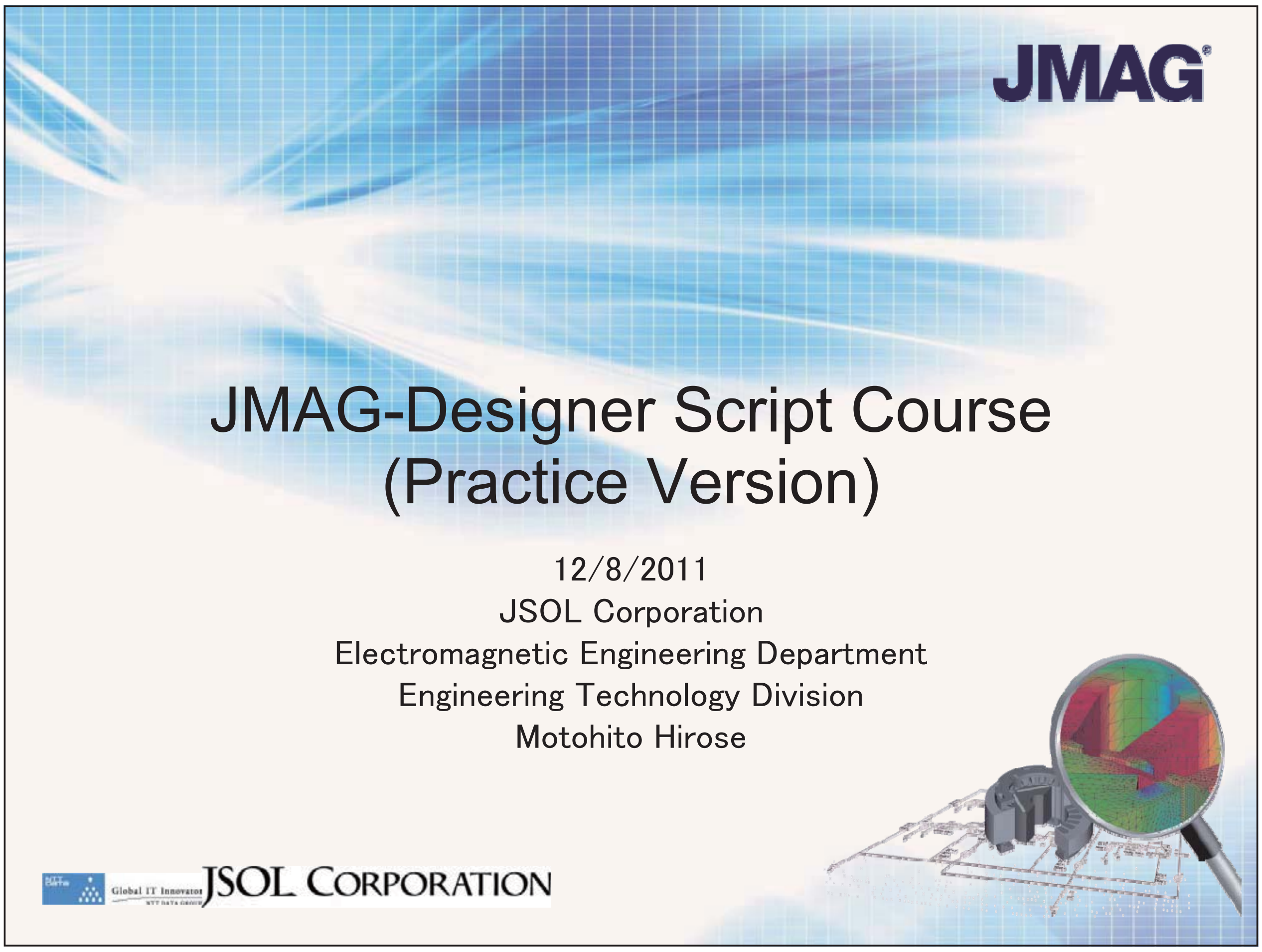
JMAG-Designer Script Course (Practice Version)

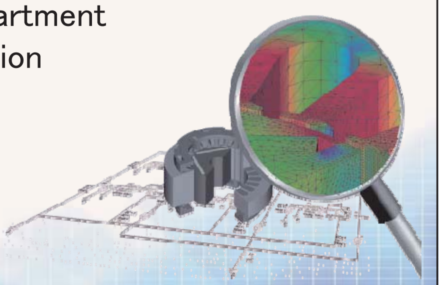
Motohito Hirose,

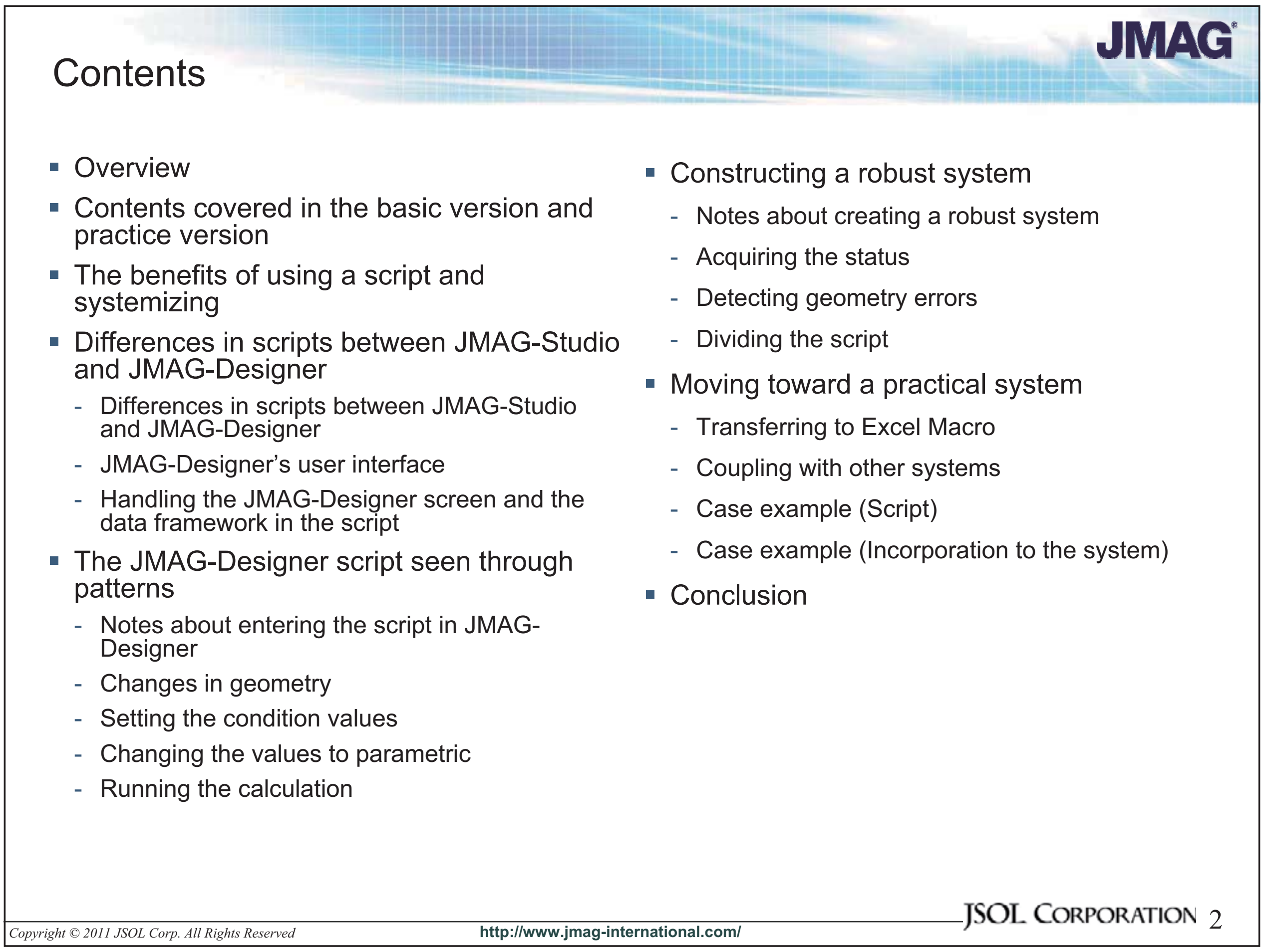
JSOL Corp.



JMAG-Designer Script Course (Practice Version)

12/8/2011
JSOL Corporation
Electromagnetic Engineering Department
Engineering Technology Division
Motohito Hirose





Contents

http://www.jmag-international.com/ JSOL CORPORATION 2

Overview

- Have you ever considered the system that JMAG is installed on, in order to make design more efficient?
- I will introduce the following in this session:
  - Knowledge that allows you to control JMAG from the exterior
  - Differences with JMAG-Studio from the standpoint of script functions
  - Case examples
- This seminar is intended for those who have some experience making systems using the JMAG-Studio script
- The script examples in these materials are from JMAG-Designer Version 11.

Contents Covered in the Basic Version and Practice Version

- JMAG-Designer Script Course (Basic Version)
  - Correcting scripts that have been automatically recorded
  - Achieving light functions in a script that can adapt to the user's needs
  - Creating an analysis model automatically by just entering the necessary variables
- JMAG-Designer Script Course (Practice Version)
  - Differences between JMAG-Studio scripts and JMAG-Designer scripts
  - The JMAG-Designer script seen through patterns
  - Constructing a robust system
  - Moving toward a practical system

The Benefits of Using a Script and Systemizing

- Reducing labor
  - Reliably carrying out uniform settings for an abundance of data
    - In CAE, a mistake with settings can lead to a large loss of time
- Templatizing
  - Changing dimensions and condition settings, and running an analysis based on a typical model
    - Preventing setting failures
    - Reducing education costs
- Packing in knowledge
  - It is possible to implement post-processing that cannot be carried out in JMAG alone.
  - Automating contents that are processed numerous times in programs like Excel

Differences in Scripts Between JMAG-Studio and JMAG-Designer

Differences in Scripts Between JMAG-Studio and JMAG-Designer

- JMAG-Studio's script
  - A unique script language
    - It reaches its limit at processing that does not operate in Studio
  - Command style
    - An list of commands to be processed
    - Operations for the model displayed

```
'+++++ JMAG-STUDIO SCRIPT FILEVersion(8.020030101)
'ReadVariableFile("C:¥work¥variable.var")
SolverType(-3)
SolverVersion(80)
Policy("001_3d-static.txt")
DATABASE("ST_elmag¥ST_elmag.jsp")
SetDocument("ST_elmag.jsp")
UnitSetting(2,   1)
Tolerance(0.001)
'-----------------------------------------------------------
PAUSE ( "Geometry creation" )
SelectType(0)
RectAngleDialog(   0,   0,   0, 100, 100)
LineDialog(5,   0,  75,   0,0,  75,   0,   0,   0,   0)
LineDialog(6,   0,  50,   0,0,  50,   0,   0,   0,   0)
LineDialog(7,  75,  75,   0,0,   0, -75,   0,   0,   0)
LineDialog(8,  50,  50,   0,0,   0, -50,   0,   0,   0)
```

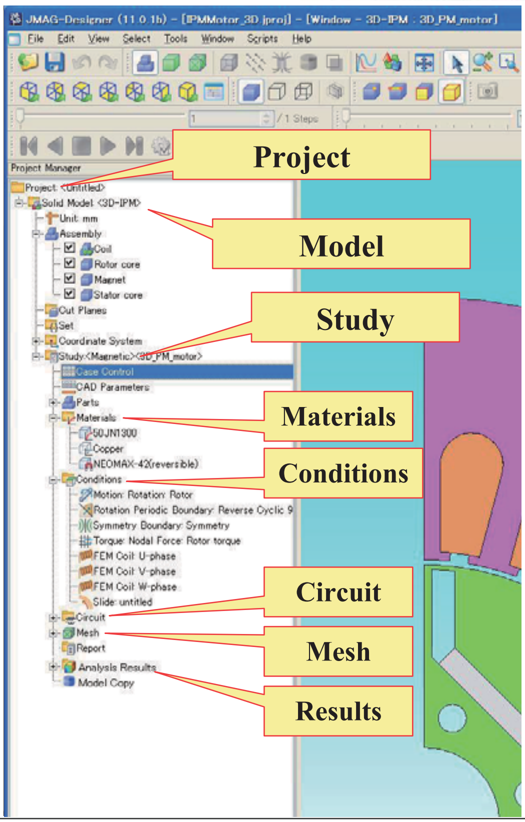
An example of the JMAG-Studio script

- JMAG-Designer's script
  - A general script language (VBScript)
    - It can carry out general processing, not just for Designer operations
  - Object based
    - It commands processing for the "Object" (It calls up the method)

```
Set app = CreateObject("designer.Application")
Call app.Show()

Call app.Load("D:/sample.jproj")
Call app.SetCurrentStudy(0)
Set study = app.GetModel(0).GetStudy(0)
Call study.DeleteMesh

Call study.Run

Set report = study.GetReport
n = report.NumErrorMessages
```

An example of the JMAG-Designer script

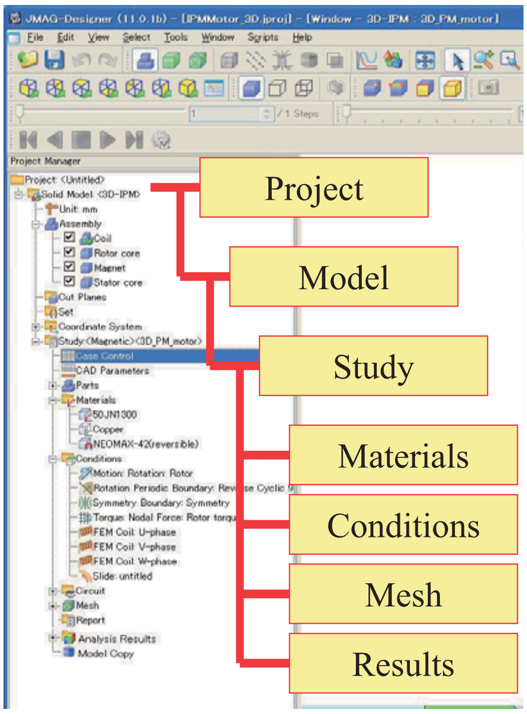
JMAG-Designer's User Interface



- In JMAG-Designer, analysis related information appears in the "Treeview."
  - Project
    - Manages the entire analysis and has several models.
  - Model
    - The geometry used for the analysis. It has several studies.
  - Study
    - The units for the analysis settings. It has the materials, conditions, circuits, and analysis results.
  - Materials
  - Conditions
  - Circuit
  - Mesh
  - Results

Handling the JMAG-Designer Screen and the Data Framework in the Script

- The tree configuration = the data configuration



●Getting Project (Application)

```
Set app = CreateObject("designer.Application")
```

●Getting Model from Project

```
Set model = app.GetModel(0)
```

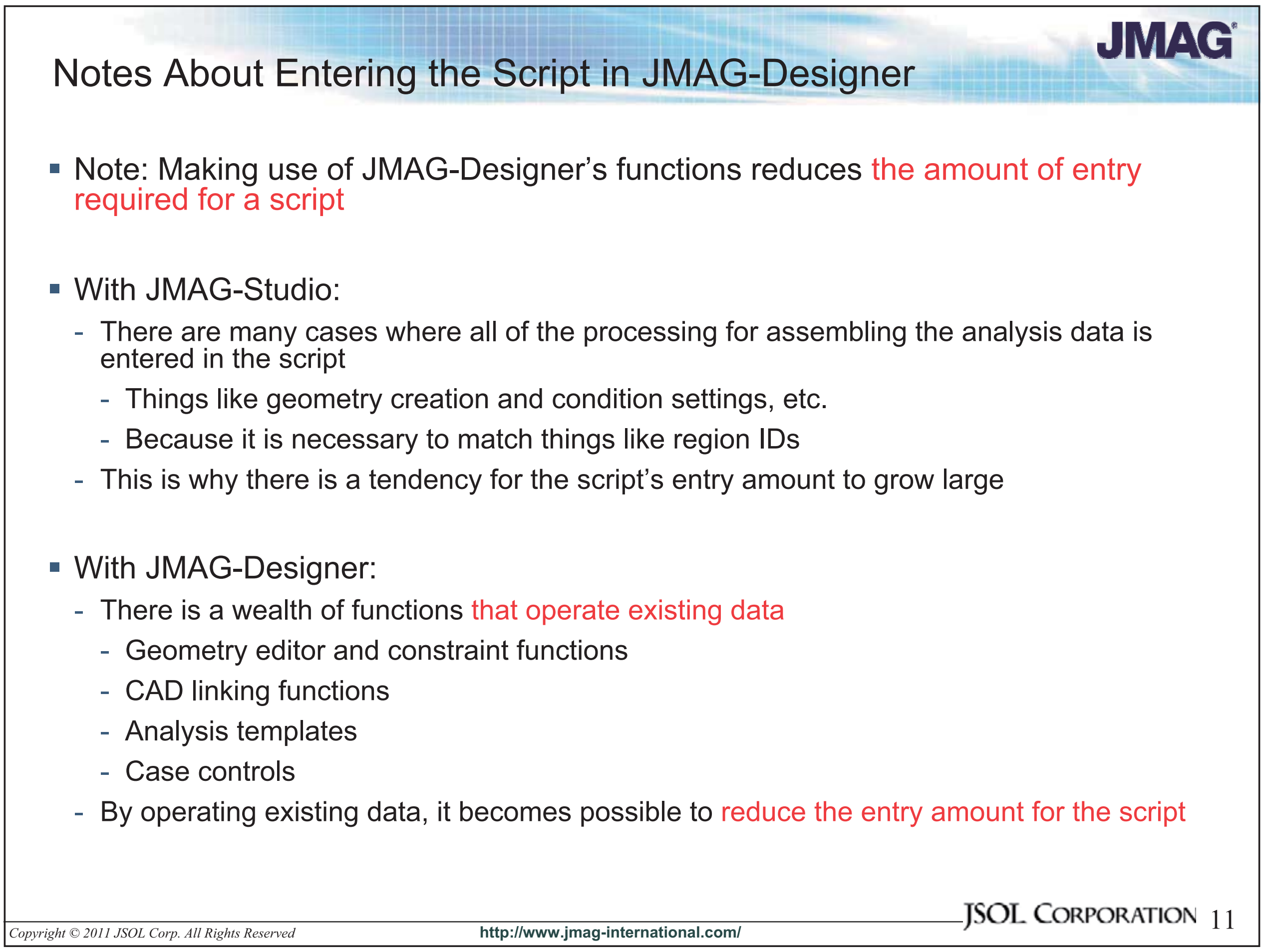
●Getting Study from Model

```
Set study = model.GetStudy(0)
```

●Getting Materials, Conditions, Mesh, and Results from Study

```
Set mat = study.GetMaterial(0)
Set cond = study.GetCondition(0)
Set mesh = study.GetMeshControl()
Set result_table = study.GetResultTable()
```

The JMAG-Designer Script Seen Through Patterns



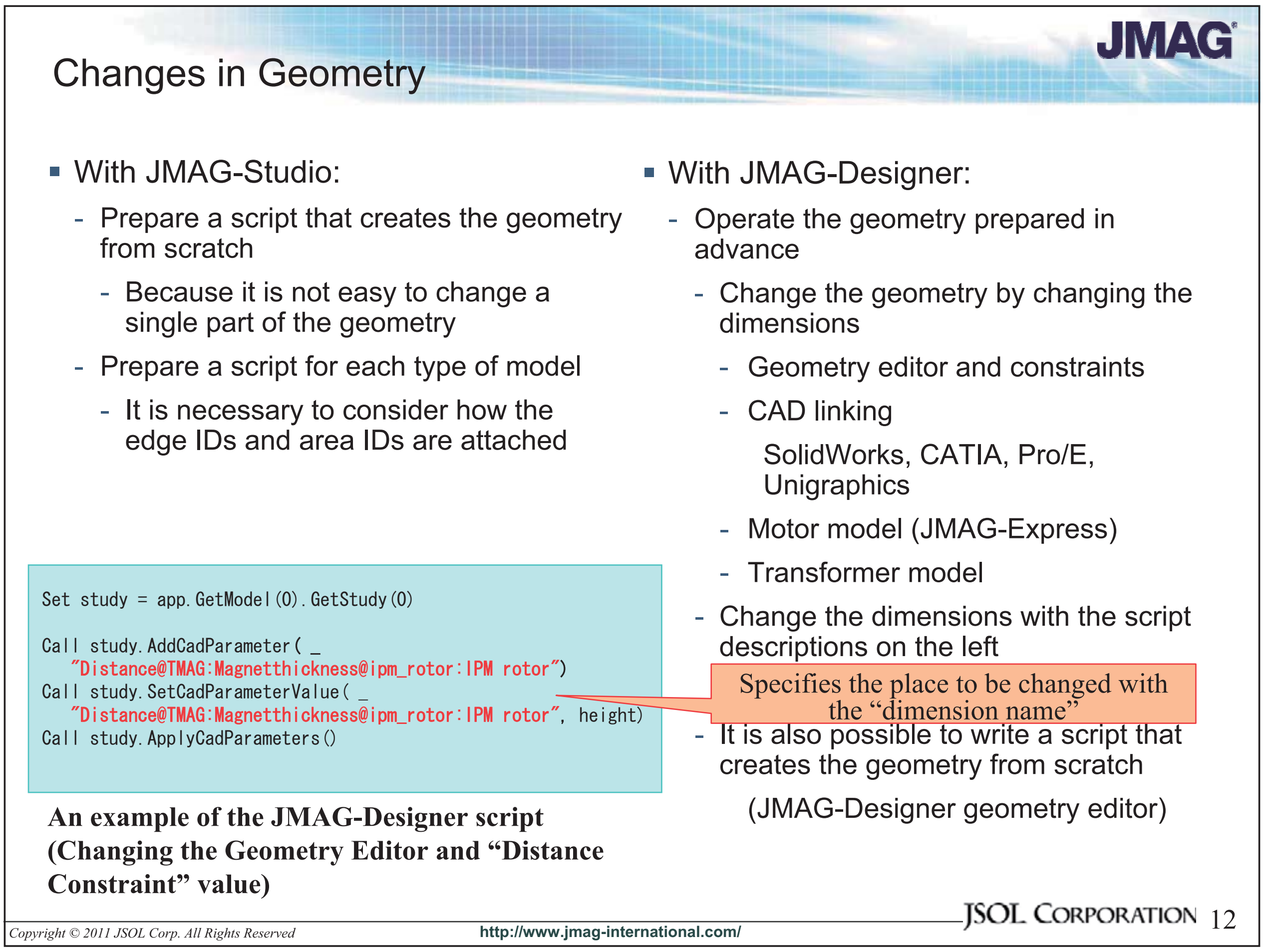
Notes About Entering the Script in JMAG-Designer

- Note: Making use of JMAG-Designer's functions reduces the amount of entry required for a script

- With JMAG-Studio:
  - There are many cases where all of the processing for assembling the analysis data is entered in the script
    - Things like geometry creation and condition settings, etc.
    - Because it is necessary to match things like region IDs
  - This is why there is a tendency for the script's entry amount to grow large

- With JMAG-Designer:
  - There is a wealth of functions that operate existing data
    - Geometry editor and constraint functions
    - CAD linking functions
    - Analysis templates
    - Case controls
  - By operating existing data, it becomes possible to reduce the entry amount for the script

http://www.jmag-international.com/ JSOL CORPORATION 11



Changes in Geometry

- With JMAG-Studio:
  - Prepare a script that creates the geometry from scratch
    - Because it is not easy to change a single part of the geometry
  - Prepare a script for each type of model
    - It is necessary to consider how the edge IDs and area IDs are attached

```
Set study = app.GetModel(0).GetStudy(0)

Call study.AddCadParameter( _
   "Distance@TMAG:Magnetthickness@ipm_rotor:IPM rotor")
Call study.SetCadParameterValue( _
   "Distance@TMAG:Magnetthickness@ipm_rotor:IPM rotor", height)
Call study.ApplyCadParameters()
```

An example of the JMAG-Designer script (Changing the Geometry Editor and "Distance Constraint" value)

- With JMAG-Designer:
  - Operate the geometry prepared in advance
    - Change the geometry by changing the dimensions
      - Geometry editor and constraints
      - CAD linking

        SolidWorks, CATIA, Pro/E, Unigraphics
      - Motor model (JMAG-Express)
      - Transformer model
    - Change the dimensions with the script descriptions on the left

Specifies the place to be changed with the "dimension name"

  - It is also possible to write a script that creates the geometry from scratch

    (JMAG-Designer geometry editor)

http://www.jmag-international.com/ JSOL CORPORATION 12

Setting the Condition Values

- With JMAG-Studio:
  - Calculate the region ID and edge ID, and assign the conditions to the IDs
    - "Predicting the ID" can be complicated

```
AddMat80(1, “Air,” 0,0, 1,0,0.000000e+000, 0,0, 0,0,1,1._
                000000e+002, 0,0,2.000000e+001, 0.000000e+000,0)
AddMat80(2, “Coil”,1,0, 1,0,0.000000e+000, 0,0, 0,0,1,1._
                000000e+002, 0,0,2.000000e+001, 0.000000e+000,0)
AddMat80(3, “Magnetic material”,3,0, 1,0,0.000000e+000, 0,0, 0,0,1,1._
                000000e+002, 0,0,2.000000e+001, 7,_
                1.000000e+000,0.000000e+000,131, 6,1,_
                ”NipponSteel/Non-oriented/35H230.hb”, _
                0.000000e+000,0,0.000000e+000,0,-1, 0.000000e+000,_
                0.000000e+000,0.000000e+000,0.000000e+000,0,0,0,0._
                000000e+000,0.000000e+000,0.000000e+000)


SetSelectMode(1)
Select(R1)
AddSelectRegionToMat(3)
Select(R3, R2)
AddSelectRegionToMat(2)
ClearSelectRegion()
```

It is necessary to predict the region ID

An example of the JMAG-Studio script

- With JMAG-Designer:
  - Search for the assignment location for conditions with part names
  - Combine with analysis templates
    - Matching with part names is possible
      - By unifying the naming rules, applying analysis templates has become easy
    - Applying an analysis template with the descriptions in the script below:

```
' Part information
ReDim refarray(191)   ‘ 6×the name of part
refarray(0) =  “Coil”  ' Template part name
refarray(1) = ”0”       ' Template part index
refarray(2) = ”0”   ‘ Template part type (Part: 0, group: 1)
refarray(3) =  “Coil”  ' Name of part being specified
refarray(4) = ”0”       ' Part index being specified
refarray(5) = ”0”       ' Type of part being specified (Part:0, group:1)
  :
  (Omitted)
  :
' Set information
ReDim refarray2(83)    ‘ 6× the name of set
  :
  (Omitted)

' Analysis template application
Call app.GetModel(0).ImportAnalysisTemplateMultiParts( _
          ”D:/template.jtmpl”, refarray, refarray2)
```

It is possible to designate with the part name

An example of the JMAG-Designer script

Changing the Values to Parametric

- With JMAG-Studio:
  - Enter the loop in the script via "for"
  - Parameterize the parts that need commands for material and condition settings

```
for n=1 to NN
  Lavel= “Magnet” & Char(n)
  DEP=90-DEG(n)
  if P(n)=0 then
    DEP=DEP+180
  End If

  if DEP<0 then
    DEP=DEP+360
  End if

  AddMat(n+2,Lavel,1,0,0.00000000000e+000,1.00000000000e+000,0,0,...
  1,2,0.00000000000e+000,1,0,0,0.00000000000e+000,0.00000000000e+...
  0000000e+000,1,DEP,BR(n),0.00000000000e+000,0.00000000000e+000....
  AddRegionToMat(n+2,n)
Next n
```

An example of the JMAG-Studio script

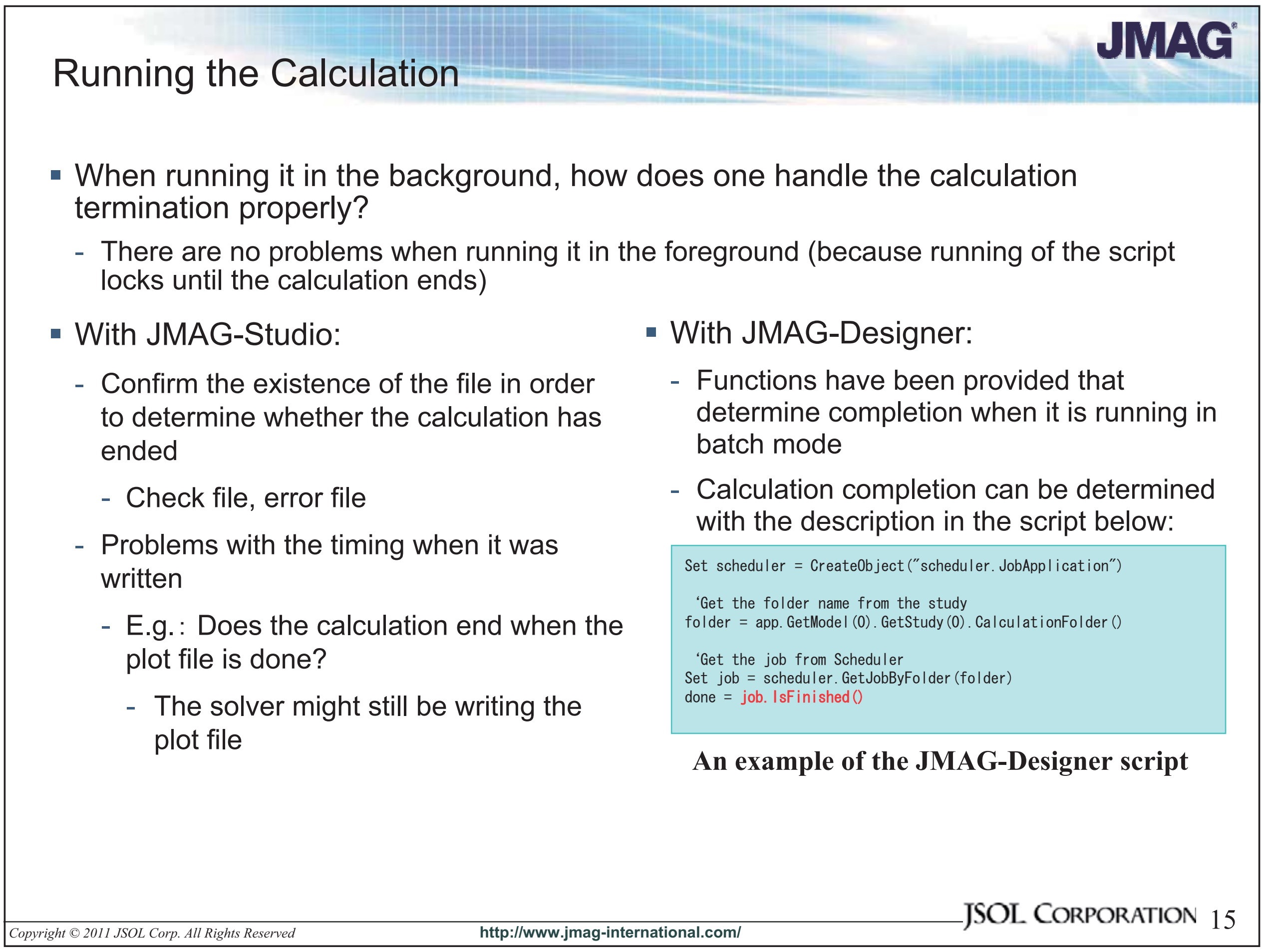
- With JMAG-Designer:
  - Use case control
  - It is possible to select parameters in the GUI
    - [Select Parametric Parameters] under [Case Control] in the treeview
  - Script descriptions are at a minimum
    - It can handle geometry as well

```
Set app = CreateObject(”designer.Application”)
app.Show()

Call app.Load(”D:¥sample.jproj”)
Set study = app.GetModel(0).GetStu

Set designTable = study.GetDesignTable()
For i=0 to (designTable.numParameters-1)

    name = designTable.ParameterName(i)    ' Get parameter name
    Call designTable.SetValue(0, i, 2.0)
Next
```

Getting the settings for case control

Value settings

An example of the JMAG-Designer script

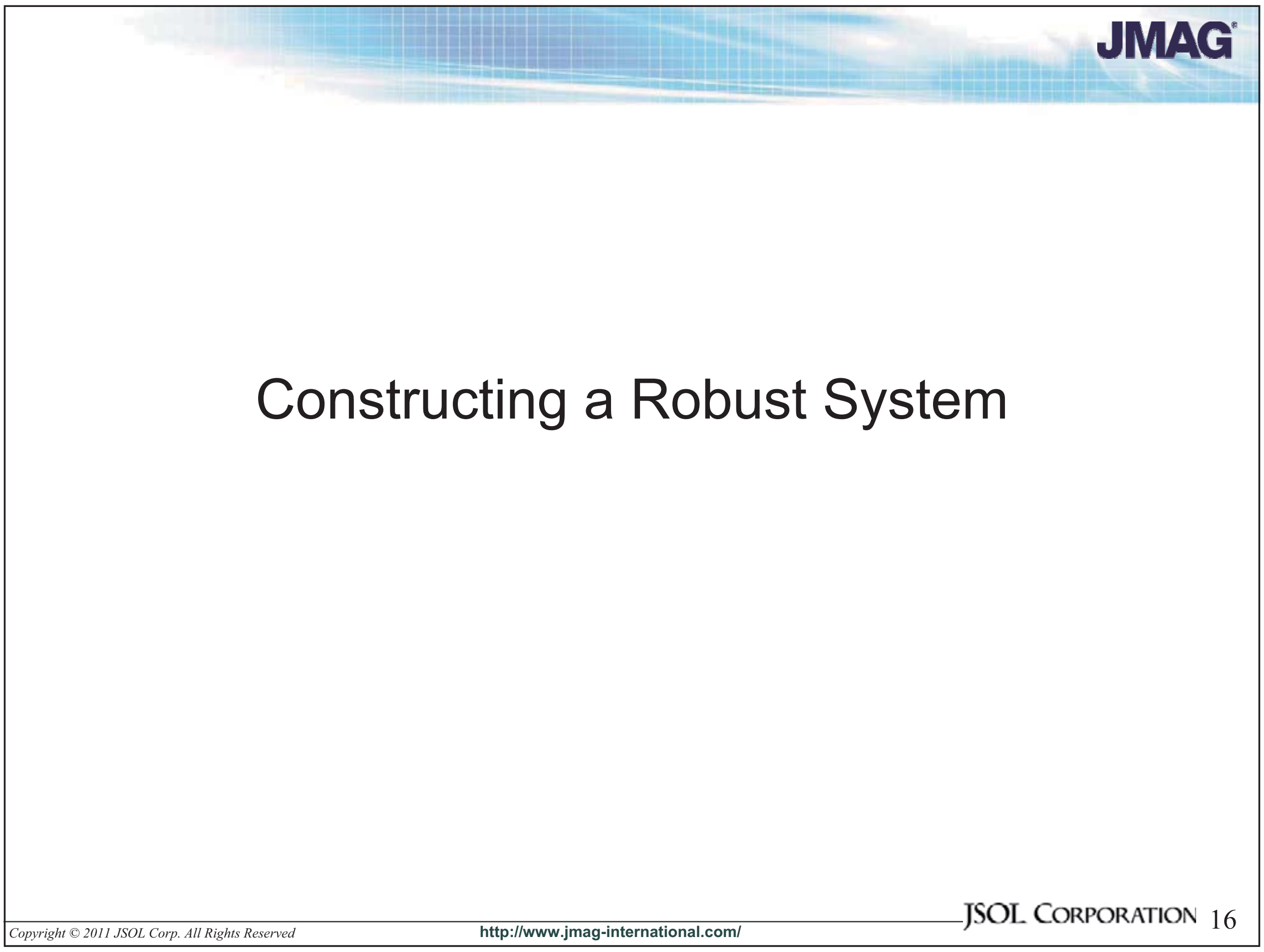


Running the Calculation

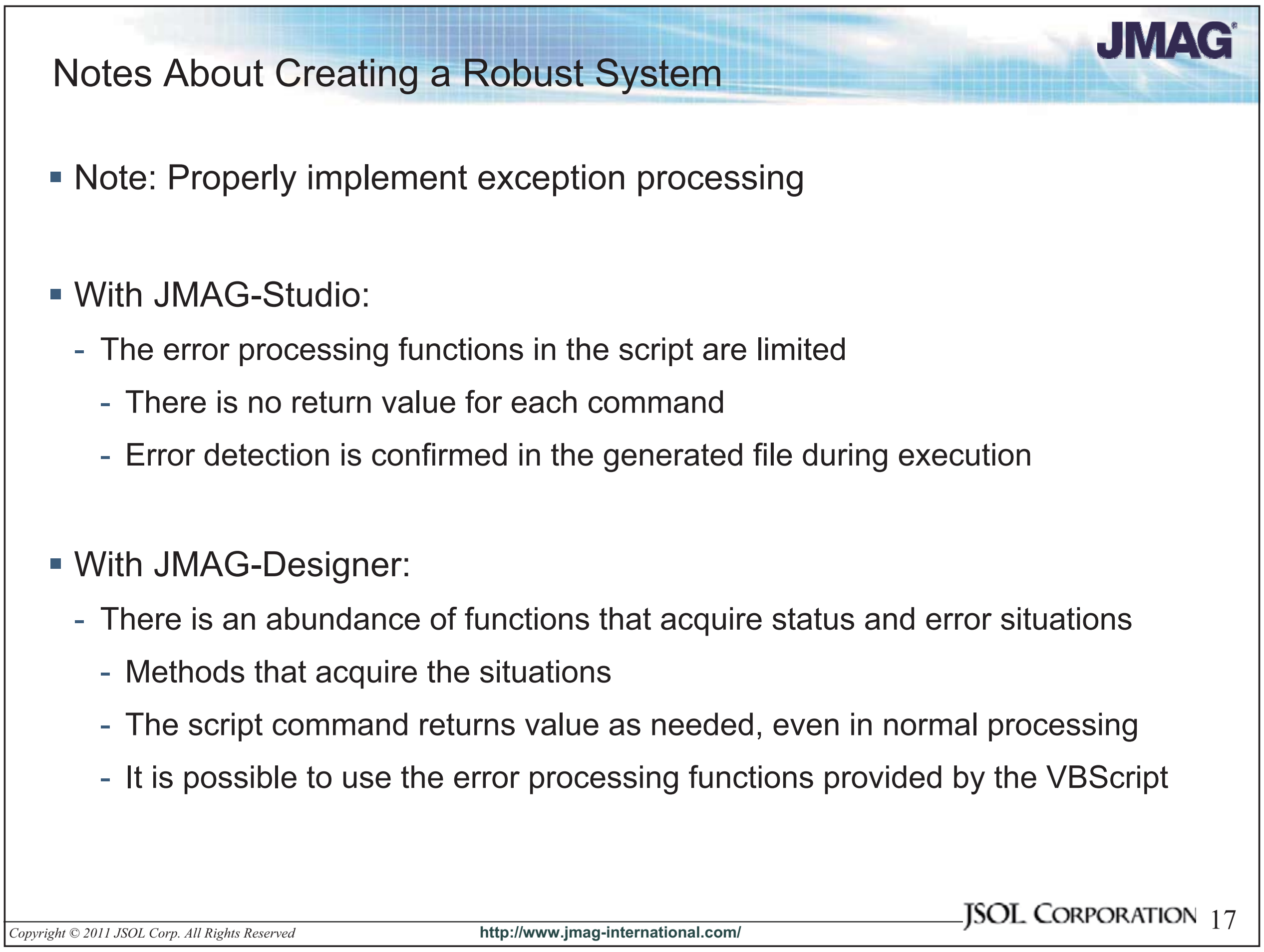
- When running it in the background, how does one handle the calculation termination properly?
  - There are no problems when running it in the foreground (because running of the script locks until the calculation ends)
- With JMAG-Studio:
  - Confirm the existence of the file in order to determine whether the calculation has ended
    - Check file, error file
  - Problems with the timing when it was written
    - E.g.: Does the calculation end when the plot file is done?
      - The solver might still be writing the plot file
- With JMAG-Designer:
  - Functions have been provided that determine completion when it is running in batch mode
  - Calculation completion can be determined with the description in the script below:

```
Set scheduler = CreateObject("scheduler.JobApplication")

 'Get the folder name from the study
folder = app.GetModel(0).GetStudy(0).CalculationFolder()

 'Get the job from Scheduler
Set job = scheduler.GetJobByFolder(folder)
done = job.IsFinished()
```

An example of the JMAG-Designer script



Constructing a Robust System



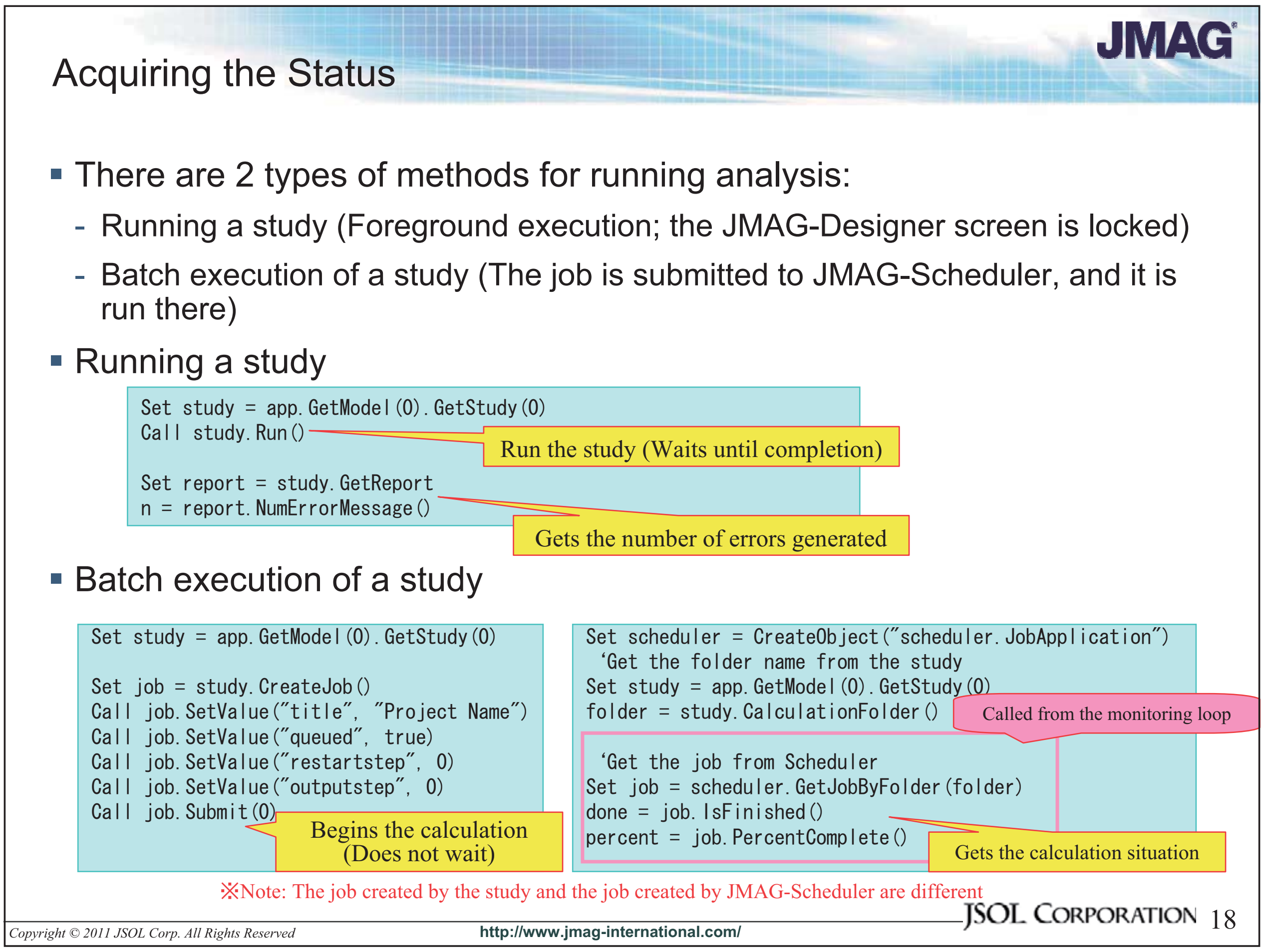
Notes About Creating a Robust System

- Note: Properly implement exception processing

- With JMAG-Studio:
  - The error processing functions in the script are limited
    - There is no return value for each command
    - Error detection is confirmed in the generated file during execution

- With JMAG-Designer:
  - There is an abundance of functions that acquire status and error situations
    - Methods that acquire the situations
    - The script command returns value as needed, even in normal processing
    - It is possible to use the error processing functions provided by the VBScript

17



Acquiring the Status

- There are 2 types of methods for running analysis:
  - Running a study (Foreground execution; the JMAG-Designer screen is locked)
  - Batch execution of a study (The job is submitted to JMAG-Scheduler, and it is run there)
- Running a study

```
Set study = app.GetModel(0).GetStudy(0)
Call study.Run()

Set report = study.GetReport
n = report.NumErrorMessage()
```

Run the study (Waits until completion)

Gets the number of errors generated

- Batch execution of a study

```
Set study = app.GetModel(0).GetStudy(0)

Set job = study.CreateJob()
Call job.SetValue("title", "Project Name")
Call job.SetValue("queued", true)
Call job.SetValue("restartstep", 0)
Call job.SetValue("outputstep", 0)
Call job.Submit(0)
```

Begins the calculation (Does not wait)

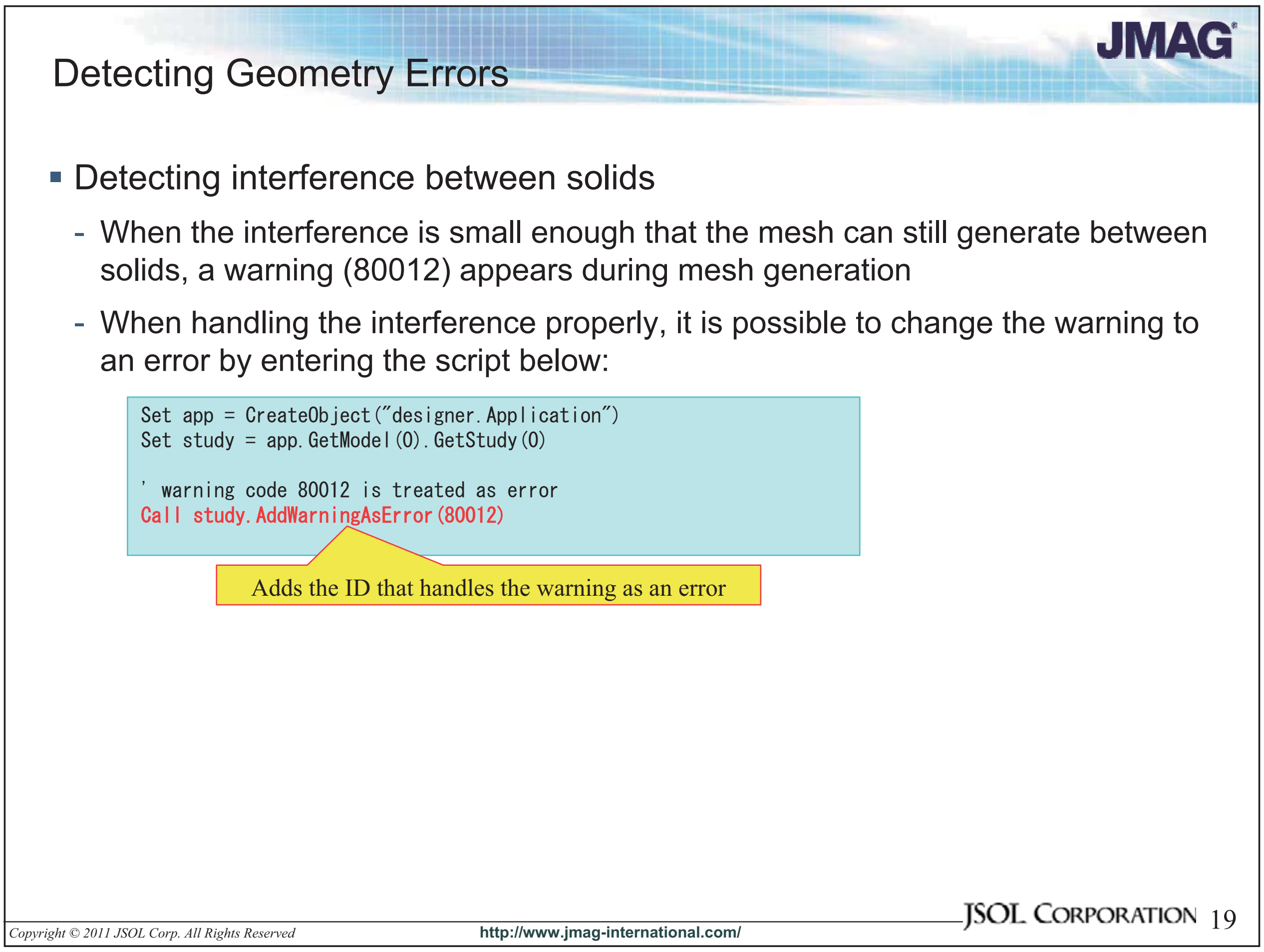
```
Set scheduler = CreateObject("scheduler.JobApplication")
 'Get the folder name from the study
Set study = app.GetModel(0).GetStudy(0)
folder = study.CalculationFolder()

 'Get the job from Scheduler
Set job = scheduler.GetJobByFolder(folder)
done = job.IsFinished()
percent = job.PercentComplete()
```

Called from the monitoring loop

Gets the calculation situation

※Note: The job created by the study and the job created by JMAG-Scheduler are different

18



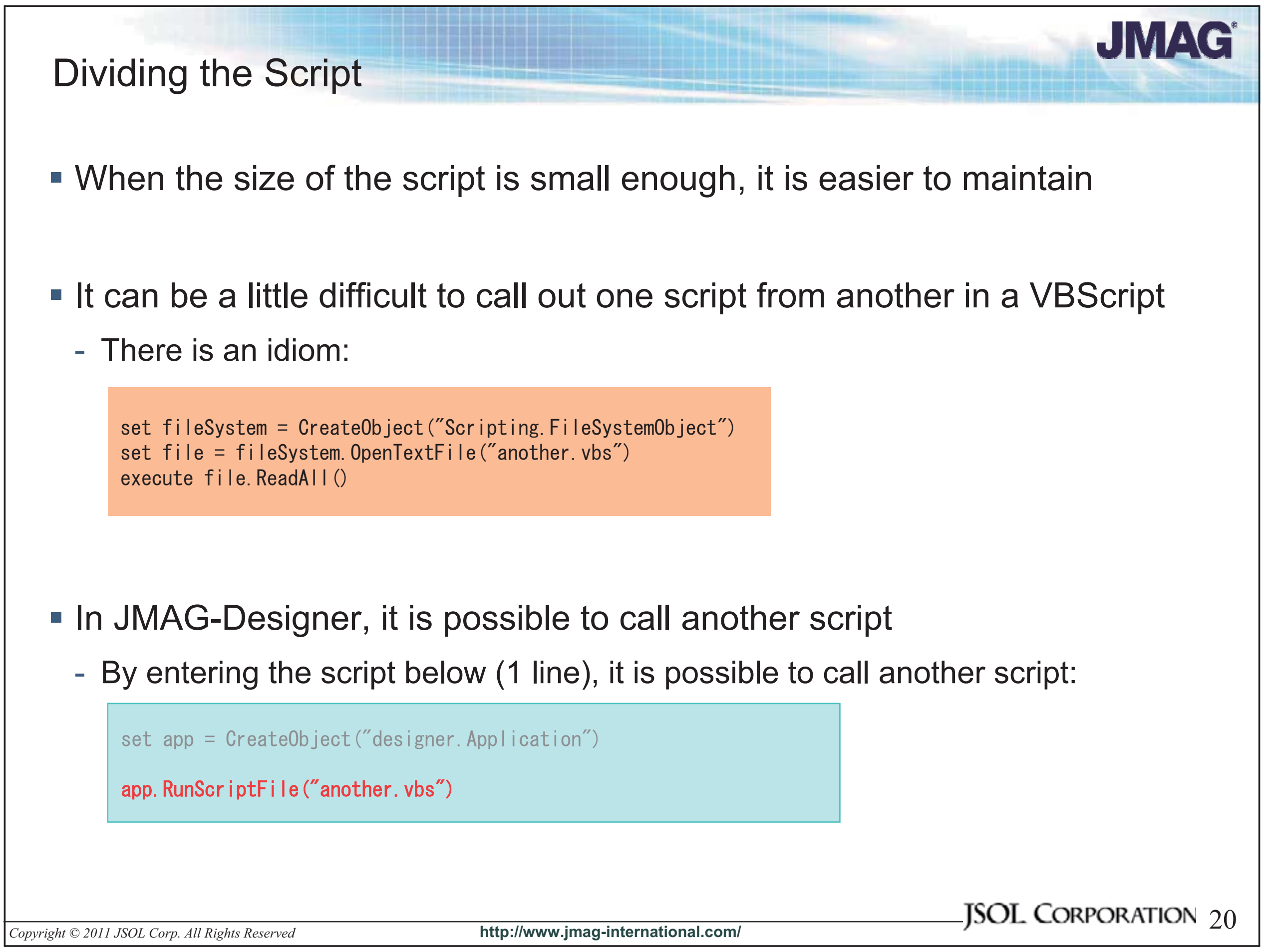
Detecting Geometry Errors

- Detecting interference between solids
  - When the interference is small enough that the mesh can still generate between solids, a warning (80012) appears during mesh generation
  - When handling the interference properly, it is possible to change the warning to an error by entering the script below:

```
Set app = CreateObject("designer.Application")
Set study = app.GetModel(0).GetStudy(0)

' warning code 80012 is treated as error
Call study.AddWarningAsError(80012)
```

Adds the ID that handles the warning as an error

http://www.jmag-international.com/ JSOL CORPORATION



Dividing the Script

- When the size of the script is small enough, it is easier to maintain
- It can be a little difficult to call out one script from another in a VBScript
  - There is an idiom:

```
set fileSystem = CreateObject("Scripting.FileSystemObject")
set file = fileSystem.OpenTextFile("another.vbs")
execute file.ReadAll()
```

- In JMAG-Designer, it is possible to call another script
  - By entering the script below (1 line), it is possible to call another script:

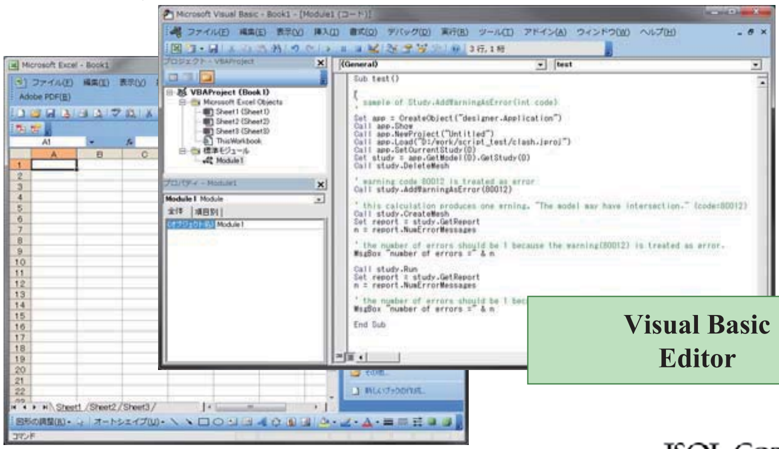
```
set app = CreateObject("designer.Application")

app.RunScriptFile("another.vbs")
```

http://www.jmag-international.com/ JSOL CORPORATION

Moving Toward a Practical System

Transferring to Excel Macro

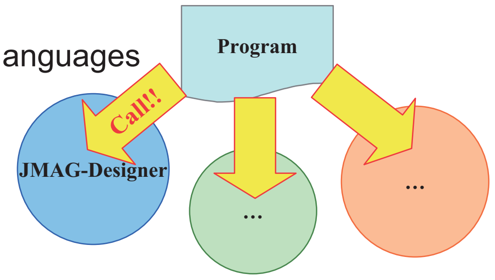
- There are often cases where Microsoft Excel is used as the system's front end
- The simplest method is to call JMAG-Designer along with VBScript
- It is relatively easy to transfer from VBScript to Excel VBA (Macro)
  - When a person wants to consolidate to a single Excel file
  - VBScript is grammatically close to ExcelVBA


```
Sub test()

'
' sample of Study.AddWarningAsError(int code)
'
Set app = CreateObject("designer.Application")
Call app.Show
Call app.NewProject("Untitled")
Call app.Load("D:/work/script_test/clash.jproj")
Call app.SetCurrentStudy(0)
Set study = app.GetModel(0).GetStudy(0)
Call study.DeleteMesh

' warning code 80012 is treated as error
Call study.AddWarningAsError(80012)

' this calculation produces one erning. "The model may have intersection." (code:80012)
Call study.CreateMesh
Set report = study.GetReport
n = report.NumErrorMessages

' the number of errors should be 1 because the warning(80012) is treated as error.
MsgBox "number of errors =" & n

Call study.Run
Set report = study.GetReport
n = report.NumErrorMessages

' the number of errors should be 1 bec
MsgBox "number of errors =" & n

End Sub
```

Visual Basic Editor

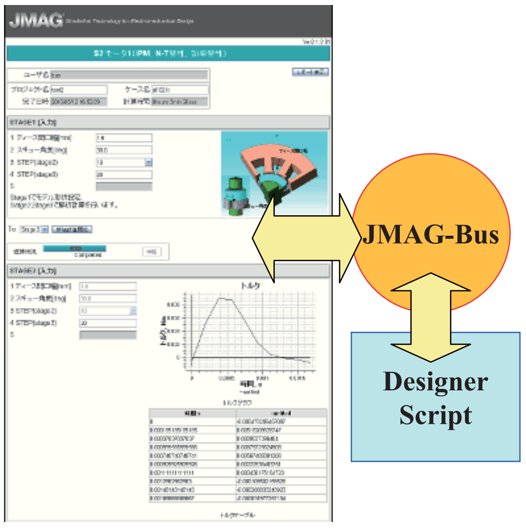
Coupling With Other Systems

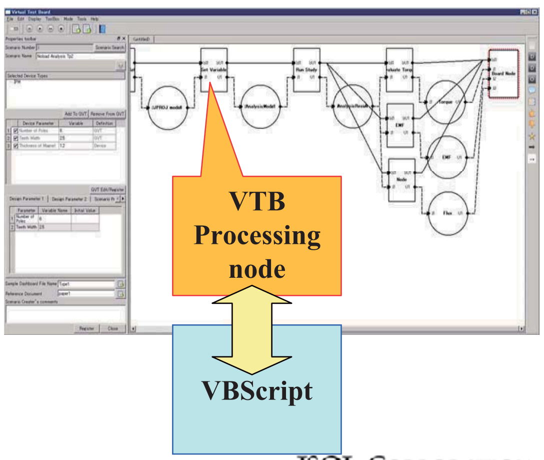
- JMAG-Designer is a COM Component, so it is possible to call it from various programming languages
  - COM = Component Object Model
    - Technology that allows programmers to use “Components” that can be used in Windows
  - VBScript is also one of “several programming languages
  - C, C++, Java, etc…
- Most (recent) applications can:
  - Be used as COM Components
  - Call COM Components from Macro and the application’s script
- The level of coupling freedom is high



Case Example (Script)

- JMAG-Bus
  - Setting and result confirmation are possible from a Web browser
    - Allows usage by a number of users
  - Scripts have been prepared for processing on the server side
- JMAG-VTB (Virtual Test Bench)
  - Analysis story ＝ running a “scenario”
  - The “scenario” calls the command (VBScript) in the background
  - JSOL provides the commands (VBScript)
    - The user can enter it, as well

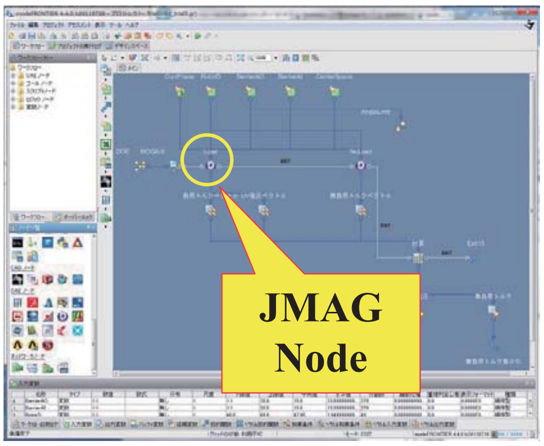




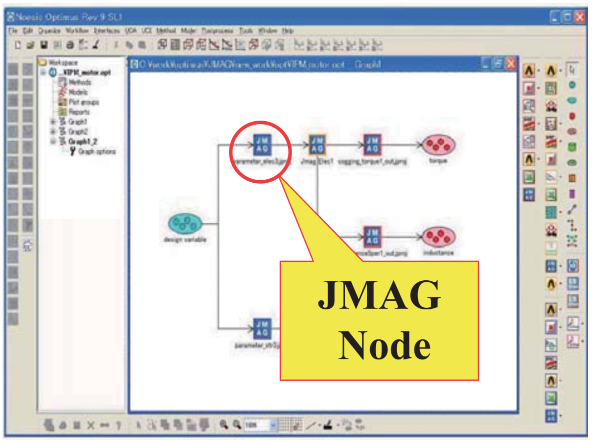
Case Example (Incorporation to the System)

- Coupling with optimization software
  - It calls JMAG-Designer from the system as a COM Component

- modeFRONTIER



- Optimus

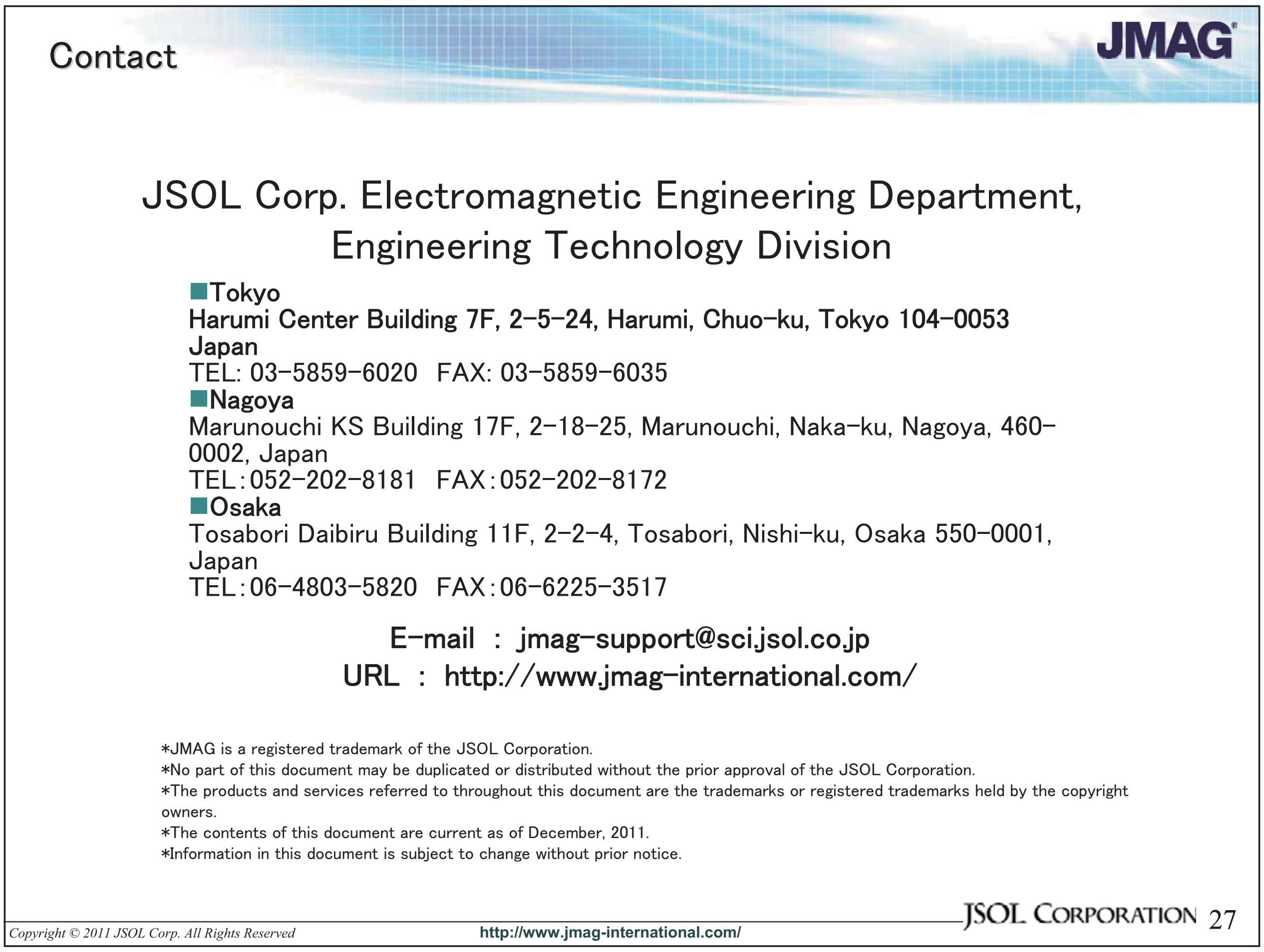


Conclusion

- By using the script function, it is possible to construct a system that has incorporated JMAG-Designer

- Along with saving labor, it is also possible to systemize expertise

- By using JMAG-Designer’s functions in coordination, it is possible to reduce the amount of script while constructing a robust system



Contact

JMAG

JSOL Corp. Electromagnetic Engineering Department,
Engineering Technology Division

■Tokyo
Harumi Center Building 7F, 2-5-24, Harumi, Chuo-ku, Tokyo 104-0053 Japan
TEL: 03-5859-6020 FAX: 03-5859-6035
■**Nagoya**
Marunouchi KS Building 17F, 2-18-25, Marunouchi, Naka-ku, Nagoya, 460-0002, Japan
TEL：052-202-8181 FAX：052-202-8172
■**Osaka**
Tosabori Daibiru Building 11F, 2-2-4, Tosabori, Nishi-ku, Osaka 550-0001, Japan
TEL：06-4803-5820 FAX：06-6225-3517

E-mail ： jmag-support@sci.jsol.co.jp
URL ： http://www.jmag-international.com/