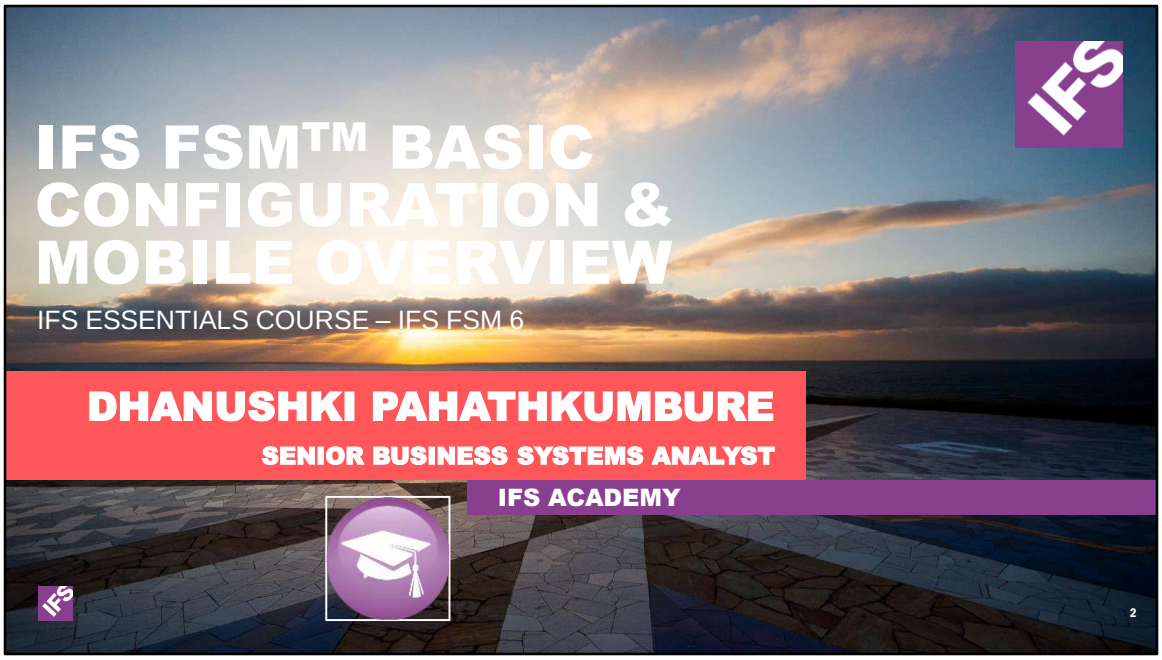
IFSworld.com


IFS FSM™ BASIC CONFIGURATION & MOBILE OVERVIEW
IFS ESSENTIALS COURSE – IFS FSM 6
DHANUSHKI PAHATHKUMBURE
SENIOR BUSINESS SYSTEMS ANALYST
IFS ACADEMY
2

This is the agenda for the week's course.
This morning is highlighted in the green box

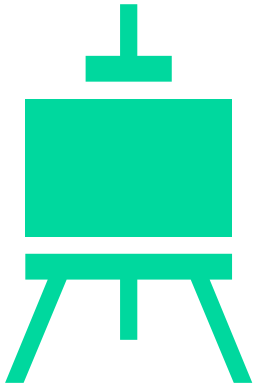

AGENDA
01
INTRODUCTION
02
INTRODUCTION TO BUSINESS RULES (BR)
03
BR CREATION AND NAVIGATION
04
TYPES OF BUSINESS RULES
05
BUSINESS PROCESS EXECUTION ORDER

OUR TRAINING APPROACH

HOW WE LIKE TO DELIVER TRAINING

Learning Activities

Facilitator-led instruction
System Demos
Take home Exercises
Knowledge Tests



LEARNING ACTIVITIES
Facilitator-led Instruction
We try not to do too much of this, but it's inevitable that some of that we do during the course will involve me taking you through some slides and providing some information about how the system is structured, is designed to work and to show you how to configure the solution.

System Demos
Then, of course, we'll show you what this looks like in the system so that you can fully appreciate what we've been talking about

Hands-On
The most important part of what you'll do while you're here is get the opportunity to do some hands-on work in the training environment. This will always be a bit artificial compared to what you'll do back in the real world, but it's really important to have a go

Activities
These will be designed to give you a break from the screen and add a bit of fun!

Review sessions
This is where we'll aim to consolidate the learning that we've covered and make sure everyone has the basis they need to move on to the next steps

Knowledge Tests
Nothing heavy, just a little self-assessment so that you can check your learning progress and understand where your knowledge gaps are

LEARNING APPROACHES

Outcomes-based
For each lesson we'll set out, at the beginning, what we anticipate you'll be able to do by the end of the course – this means we're focused on what you'll be able to do with the training, not just on the training itself

Field service context-based
Field service is quite a distinctive business area and if you understand a bit about that context, it will really help you to appreciate what IFS customers are looking for when they implement this solution

Practically, functionally-based
We want to teach you things that are useful; whilst the code behind the scenes might be interesting to some, this course is about a practical, functional knowledge base that will allow you to implement the solution

Peer learning
There are people here from different backgrounds, with different experiences; if you get the opportunity to hear and share some of that, you'll find it makes the course more useful and more interesting


IFS
INTRODUCTION TO BUSINESS RULES
6

RULES

LEARNING OUTCOMES

- By the end of this lesson, you should:
  - Understand the purpose and importance of rules
  - Recognize the components of business rules
  - Recognize the components of financial rules

7

RULES OVERVIEW

There are 2 types of Rules that control processing within FSM:

BUSINESS RULES

- Represent standard tasks in varying scenarios
- Control how records move through the process
- Control workflows
- Based on conditions set during system configuration
- Process sequentially
- Can be viewed in-process to identify errors

FINANCIAL RULES

- Control billing and financial processing
- Used for both pricing and costing; parts and non-parts

TIP: It is ESSENTIAL to understand the business that you are configuring FSM for. This will be covered in the next part of the Course.

8

First of all, **you need to understand your business** to create business rules or pricing rules.

Let's take the following scenario. Your organization has a call center with many field technicians who services your equipment in the field. In order to create the business rules, you will first need to define your standard tasks. Begin by identifying the various service delivery scenarios performed by your organization. Next, identify the various activities necessary for each scenario. Note that an activity may appear in multiple scenarios or multiple times within a single scenario. For each activity, determine what business rule and pricing rule needs to be set up.

In other courses, such as *IFS FSM Services, IFS FSM Repair Center,* and *IFS FSM Financials*, we discuss different scenarios and potential rule set up.


IFS
TYPES OF BUSINESS RULES
9

BUSINESS RULES

TYPES OF RULES

XML

Sending any XML transaction based on condition(s)

VALUE

Setting certain columns of a table

VALIDATION

Throws an exception upon a condition match

ALERT

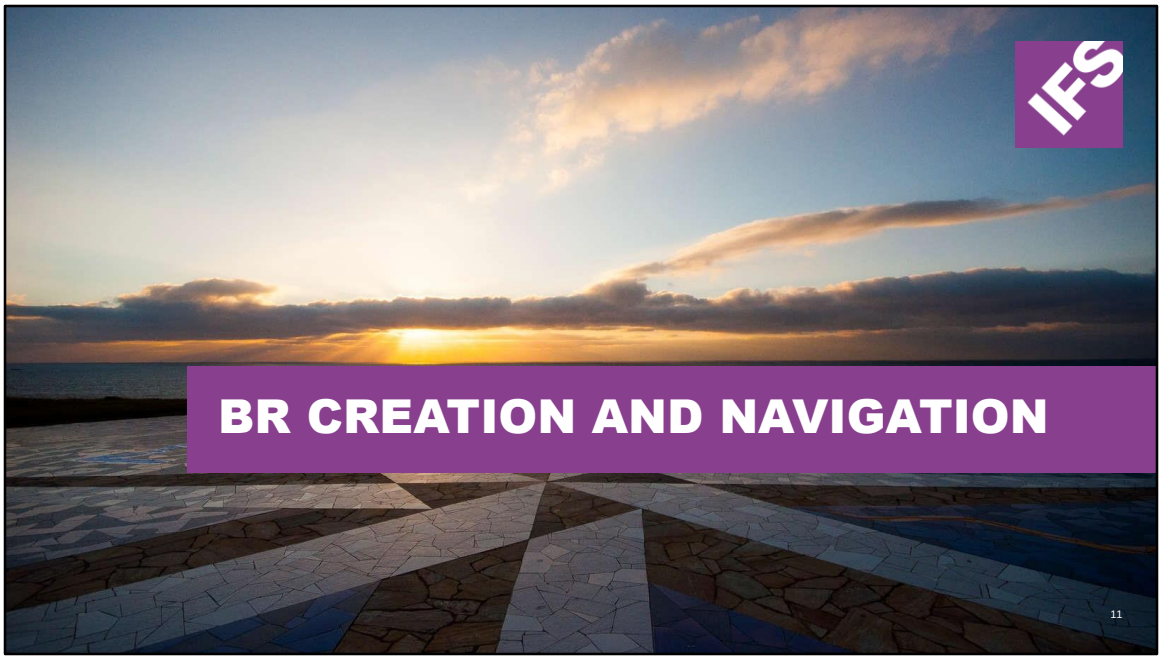
Cause a pop-up to make the user aware. Not an error.

NOTIFICATION

Allows to send email or SMS notifications

COPY

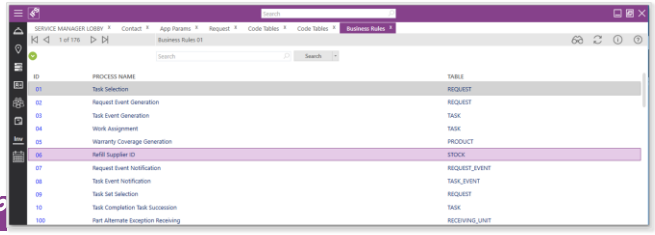
Standard function for copying records.

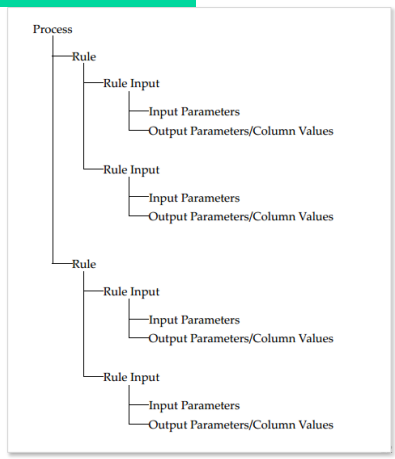

IFS
BR CREATION AND NAVIGATION
11

RULES

BUSINESS RULE COMPONENTS

- Both Baseline and Custom Business Rules follow the same structure
- Each Business Process can have multiple business rules
- Each rule is individually defined





Both baseline and custom business rules follow the same structure. Business rules are applied to one, primary table. It uses built-in metadata to define relationships and may have very large number of input conditions. The output of values or processes are not constrained. A process trigger occurs due to a database change or explicit call. They are evaluated in one pass when invoked. If there are exceptions to the baseline business rules, then custom business rules must be explicitly coded.

Each of the Business Processes listed on the left can have multiple Business Rules. The Business Rules each have their own definition and Rule Input entries. The Rule Input entries are the fields that you want to use in the criteria that will determine whether or not to execute the Business Process. The Rule Input entries listed are for the selected Business Rule. Once you have defined the Business Rules for the Business Process, including their Rule Input entries, you will need to maintain the conditions for each of these rules. After saving the Business Rules, drill into each rule by clicking on the underlined, blue colored Rule ID, which is a hyperlink to the Rule Conditions screen.

Rules define the map of inputs to an output, either a parameter or table columns. A process can have many rules. Rules are evaluated sequentially based on the search order you specify. Depending on the process, evaluation of rules either stops after the first match or after all of the rules are evaluated.

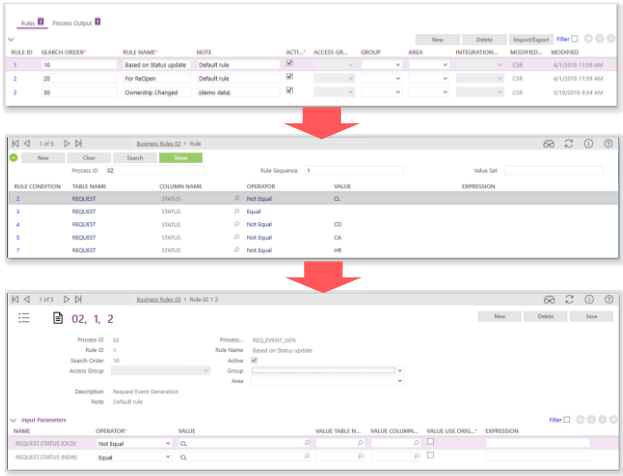
Rule Input are values on the database that you match to an output value, which is either a parameter for an action or column values on the database. Each process has an associated primary table and many processes also have related tables. You can specify any number of columns on these tables as rule inputs. When all values for each rule input match, the specified output value is used. When you specify input values that appear on code tables, for example information that appears on a drop-down list, you specify the code instead of the code description or associated message text. You can specify more than one input parameter that maps to the same output parameter. Specify each parameter separated by a comma (,) without spaces. For example, if you want to specify multiple state codes, you could specify CA,OR,WA,AZ.

Process performs an action. Most processes call an MPM to perform the action while some processes set a column value on a table. We supply multiple processes for you to use. You can also create your own processes using Studio.

RULES

BUSINESS RULE COMPONENTS

- Within the Process there are Rules
  - Each Rule can have a search order
  - Each Rule can have multiple conditions
  - Each condition has an Input variable and dependent Output



Both baseline and custom business rules follow the same structure. Business rules are applied to one, primary table. It uses built-in metadata to define relationships and may have very large number of input conditions. The output of values or processes are not constrained. A process trigger occurs due to a database change or explicit call. They are evaluated in one pass when invoked. If there are exceptions to the baseline business rules, then custom business rules must be explicitly coded.

Each of the Business Processes listed on the left can have multiple Business Rules. The Business Rules each have their own definition and Rule Input entries. The Rule Input entries are the fields that you want to use in the criteria that will determine whether or not to execute the Business Process. The Rule Input entries listed are for the selected Business Rule. Once you have defined the Business Rules for the Business Process, including their Rule Input entries, you will need to maintain the conditions for each of these rules. After saving the Business Rules, drill into each rule by clicking on the underlined, blue colored Rule ID, which is a hyperlink to the Rule Conditions screen.

Rules define the map of inputs to an output, either a parameter or table columns. A process can have many rules. Rules are evaluated sequentially based on the search order you specify. Depending on the process, evaluation of rules either stops after the first match or after all of the rules are evaluated.

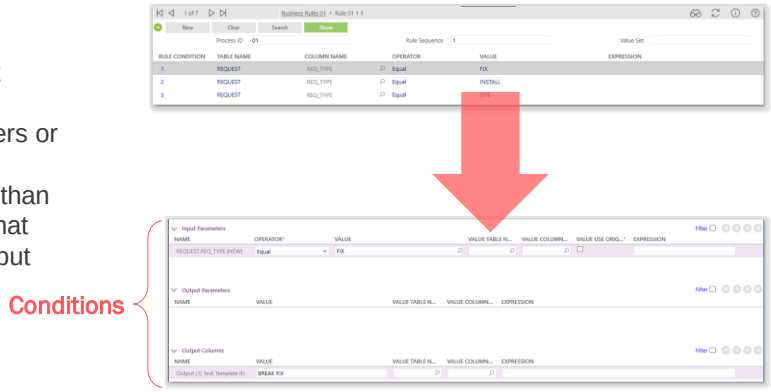
Rule Input are values on the database that you match to an output value, which is either a parameter for an action or column values on the database. Each process has an associated primary table and many processes also have related tables. You can specify any number of columns on these tables as rule inputs. When all values for each rule input match, the specified output value is used. When you specify input values that appear on code tables, for example information that appears on a drop-down list, you specify the code instead of the code description or associated message text. You can specify more than one input parameter that maps to the same output parameter. Specify each parameter separated by a comma (,) without spaces. For example, if you want to specify multiple state codes, you could specify CA,OR,WA,AZ.

Process performs an action. Most processes call an MPM to perform the action while some processes set a column value on a table. We supply multiple processes for you to use. You can also create your own processes using Studio.

RULES

BUSINESS RULE CONDITIONS

- **Rule Conditions** are the individual pairings of input parameters to outputs
  - either output parameters or table columns.
  - You can specify more than one input parameter that maps to the same output parameter.



14

Rule Conditions are the individual pairings of input parameters to outputs, either output parameters or table columns. The output parameters are specified by the process. Some processes output table column values and some processes do not require an output parameter. When you specify input values that appear on code tables, for example information that appears on a drop-down list, you specify the code instead of the code description or associated message text. You can specify more than one input parameter that maps to the same output parameter. Specify each parameter separated by a comma (,) without spaces. For example, if you want to specify multiple state codes, you could specify CA,OR,WA,AZ.

The Rules Condition screen is where you define the conditions upon which execution of the Business Process is contingent. Each condition corresponds to a specific set of criteria and is evaluated separately from every other condition. A condition is a logical set of comparisons between an Input Parameter (a field in the primary or related table record) and some value. Keep in mind that Business Rules and the related processing are all based on some change (transaction) being made to a record in the primary table for the Business Process. Hence, there is the concept of the record in process - the record being acted upon that triggers the Business Rule processing - which has an 'original' state, and the 'in process' or 'new' state that the transaction is in the process of committing when the Business Rules are being evaluated. When all the criteria for a condition evaluate to TRUE, the Business Process will execute using the values defined in Output Parameters and/or Output Columns. Remember that the Business Process will execute for All such conditions or just the first one that evaluates to TRUE, depending on how the Business Process 'Execution Type' was defined. In Studio, it is called 'Execution Type', but in the client, it is called 'Process Trigger'.

Each criteria defined for a condition is a logical comparison between a field on a table (primary, or related to the primary table) and some value. Each of these criteria will need to resolve to TRUE for the Business Process to execute on this condition.

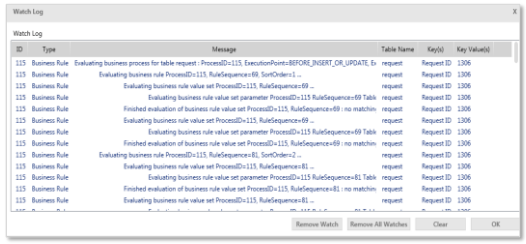
If the metadata has a lookup defined for the field identified in the Name field, the Value field will also be a lookup for Output Columns and Input Parameters.

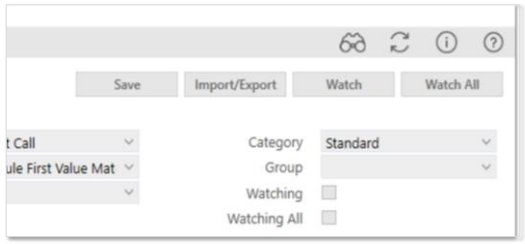
RULES

WATCHING BUSINESS RULES

When testing a business rule, it can be watched, this is useful in determining if the rule is performing correctly or as expected

- You can Watch an individual rule
- Or Watch All





15

When a business rule is being evaluated or performing, you can watch it happen. This is helpful to decide whether the configuration is performing correctly along with helping narrow down and troubleshooting if it is not working as expected. When a business rule is evaluated or actually executed then it will log and prompt the logging information to the user. There is a watch and watch all feature meaning you can watch one rule or client script or watch all of them.

RULES

BUSINESS RULE SUPPORTED OPERATORS

Operator	Description
( )	Parentheses change precedence—operations inside parentheses are performed before those outside the parentheses
*, /. %	Multiplication, division, remainder
+, –	Addition, subtraction
to days, to hours, to minutes, to seconds	Convert to days, convert to hours, convert to minutes, convert to seconds
<, <=, >, >=	Less than, less than or equal, greater than, greater than or equal
=, !=, <>, is null, is not null, like, not like	Equals, not equals, not equals alternate, is null, is not null, like, not like
AND	Logical and
OR	Logical or
?	Conditional operator

16

You can use an expression to represent a value in the rule condition.
You create expressions using operators and operands.
When evaluated, the expression returns a value that is used to evaluate the rule input. When defining the value to compare to, you can specify ONE of:
A literal value entered under 'Value'. This is a static value that you manually enter.
A table and column that will contain the value entered under 'Value Table' and 'Value Column'.
This is a *related* table and column that contains the value for comparison.
An expression that resolves to a value entered under 'Expression'.
An expression that resolves to a value of the same type (date, string, etc.) as the Rule Input Parameter. An example might be: request.user_def_num1 + request.user_def_num2. Another example might be: task.created_dttm > Now(). We will discuss in the next few slides the different expressions that can be used in the field.


IFS
BUSINESS PROCESS EXECUTION ORDER
17

BUSINESS PROCESS

ORDER OF EXECUTION

1) Before_Save

Switch on DataRowState -

Deleted: 2) Before_Delete

Added: 2) Before_Insert, Before_Insert_Or_Update

Modified: 2) Before_Update, Before_Insert_Or_Update

Run Update Process

For each column in the update:

3) Before_Change

Loop

BUSINESS PROCESS

ORDER OF EXECUTION

UPDATE COMMAND SENT

Switch on DataRowState -

Deleted: 4) After_Delete

Added: 4) After_Insert, After_Insert_Or_Update

Modified: 4) After_Update, After_Insert_Or_Update

5) After_Save

COMMIT TRANSACTION

BUSINESS PROCESS

ORDER OF EXECUTION

6) After_Commit

```
Switch on DataRowState -
```

Deleted: 7) After_Commit_Delete

Added: 7) After_Commit_Insert, After_Commit_Insert_Or_Update

Modified: 7) After_Commit_Update, After_Commit_Insert_Or_Update

FEEDBACK ON THE OBJECTIVES OF THIS LESSON

By the end of this lesson, you will:

Be able to create a simple approval process

Have viewed simple escalations

Recognize the features of multi-level approvals

Show the differences between personalization, configuration, customization and adminutesistration.

Personalization: Show how to change the UI Theme (under File, UI Themes)

Configuration: Show how to do a message translation of request_type

Customization: Show how to create a customized screen under the UI Designer and apply it to a role

Adminutesistration: Show the Adminutes menu including the baseline business rules and application parameters


IFS
WRAP-UP
23

IFSworld.com


IFS
#forthechallengers