**Validation and Substitution**

**Difference between Validation and Substitution**

**Validation**
- validation will check values or value intervals entered by the user
- built-in help: check function is available to ensure the validation is logically correct
- output will be either a warning message (this message can be ignored) or an error message

**Substitution**
- values or value intervals are validated according to certain conditions (prerequisites) defined
- in case the condition is met during the program run a replacement will happen
- output will be a replacement of values or intervals

**Steps of creating a validation / substitution:**

1) **APPLICATION** where ?
2) **CALLUP-POINT** when ?
3) **DEFINITION** what and how ?
4) **RELATION** organizational unit
5) **ACTIVATION** active ?

First step is to define which application should use validation / substitution (where?) The following applications use the VSR-Tool and this makes this tool a cross-application-tool:

- AM Asset Management
- CO Controlling
- CS EC-CS Consolidation
- FI Financial accounting
- SL Special Purpose Ledger
- LC Consolidation
- PS Project system
- RE Real Estate

Second step is to define when exactly a validation / substitution is happening. In tr. GB31 you can check the combinations of Callup-Points and Applications. Each combination builds a so-called Boolean-Class. In table GB31, you will also find the information, if substitutions or validations are possible for a certain Boolean-Class. Some examples:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FI</td>
<td>1</td>
<td>8</td>
<td>x</td>
<td>x</td>
<td>Document Header</td>
</tr>
<tr>
<td>FI</td>
<td>2</td>
<td>9</td>
<td>x</td>
<td>x</td>
<td>Document Line</td>
</tr>
</tbody>
</table>
Third step is the most complex: define the validation / substitution itself. (what to do and how to do)

Forth step to assign the validation / substitution to an appropriate org unit e.g. company code.

Last step is to activate the program.

**Validation**

Many values that you enter into the SAP R/3 System e.g. in tr. fb60 are automatically checked against standard validation rules.

eg. document date, posting period opened, existing company code

These validations are R/3 checks not Customer defined. These are standardized, however, Customer specific validations cannot be standarized. FI-SL*provides a wide range of solution approaches to create customer validations. FI-SL validations allow you to create your own Boolean statements without making any changes to the standard system.

**Validation Structure**

**Each validation step consists of these 3 main points:**

![Validation Structure Diagram]

**Callup points**

Validation/substitution callup points are specific places in an application, which specify the exact location where a validation/substitution occurs.

At each callup point, the system can use information you have already entered (such as a user name, header information, and information entered in previous callup points) to validate/substitute the values you enter at a callup point.

At callup point (1) only those fields above and the system fields can be used for validation.

Exception: Document number, Cross-CC no. are blank when they are created automatically by the system at the end of the process.
**Boolean class**

Boolean class is a technical term, which determines the fields that can be used in a validation/substitution/rule definition, as well as the message class for validation messages (see table GB31 and GB01).

GGB0: Define Validation
GGB4: Activate Validation
GCT9: Transport Validation

When defining a validation in tr. GGB0 the R/3 system generates coding according to validation definition.

Pools for generated Coding:

<table>
<thead>
<tr>
<th>cc</th>
<th>aa</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBTbagai0</td>
<td>Mainprogram</td>
<td></td>
</tr>
<tr>
<td>GBTbagaiD</td>
<td>Validation</td>
<td></td>
</tr>
<tr>
<td>GBTbagaiE</td>
<td>Prerequisites</td>
<td></td>
</tr>
<tr>
<td>GBTbagaiF</td>
<td>Checks</td>
<td></td>
</tr>
</tbody>
</table>

cc = encoded client (SE37: G_CLIENT_ENCODE, G_CLIENT_DECODE)
aa = Application Area

Example:

As stated above the validation step consists of 3 points:

<table>
<thead>
<tr>
<th>Step 001</th>
<th>Test valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prerequisite</td>
<td></td>
</tr>
<tr>
<td>Check</td>
<td></td>
</tr>
<tr>
<td>Message</td>
<td></td>
</tr>
</tbody>
</table>

The validation ensures that postings only with Document type (Belegart) ‘SA’ can be performed in Company Code VATO. In case the Document type is different an error message appears and no posting can be carried out.
In transaction FB01 the two different outcome of the validation can be realized.
In case you would like to analyse it coding side:

You can use „=SHCE“ and „=SHCF“ in the command field while you are on the validation step:

The command will bring you to coding part of „Prerequisite“
The other command will bring you to coding part of "Check"
Breakpoints can be activated here and the debugging procedure can be started. However, R/3 system offers a trace function. You can use the Extras => Switch on total trace function from the Menu. A tracing function allows to test a validation/substitution/rule definition during the posting process. While posting a document, the system will call the validation/substitution/rule and display the rule contents, as well as how the data is used for validation or substitution. This function allows you to diagnose possible errors in the validation/substitution/rule definition. Further information can be found in KBA 1589540.

Substitution

A substitution is the checking process of entered values against a user-defined Boolean statement. If the statement is true, the system substitutes specified values into the system. Important to mention that the system performs substitutions before validations so that substituted values can also be validated.

Process of a Substitution definition
Fields released for substitution

Tables GB01 and GB01C determine which fields can be used for a specific Boolean class. Table GB01 is a client-independent, control table delivered with the SAP R/3 System and cannot be changed; table GB01C is client-dependent and is provided specifically for customer configuration. Only fields having blank in field BEXCLUDE are substitutable. Nevertheless some critical fields may be overwritten by other components after the substitution (for example: profit center field). Customer can modify, however, removing the BEXCLUDE-Flag in GB01 is formally modification! note 42615

Create / Change Substitution

Pools for generated Coding

GBTccaa0 Mainprogram (Enter „=SHC0“ in command field)
GBTccaaB Substitution (Enter „=SHCD“ in command field)
GBTccaaC Prerequisites (Enter „=SHCE“ in command field)

cc = encoded client (SE37: G_CLIENT_ENCODE, G_CLIENT_DECODE)

aa = Application Area

Example

A substitution was created with the below settings:
In transaction FB01 the document header is empty:

After the posting process in the header, posting date is visible:
In transaction SE37 with the entering of Function Module G_VSR_SUBSTITUTION_CALL / G_VSR_MULTI_SUBSTITUTION_CALL, all of the substitutions can be checked on coding level. (It's worth to put the breakpoint at the beginning of the function module)

As in case of Validations the trace function is also available here. Further information can be found in KBA 1589540.
User Exit

User exits are user-defined FORM routines, which are used to calculate and/or replace values within a validation, substitution, or rule.

User exits have the following format: Uxyz, for example: U100

User Exits are stored in Formpools. The name of the Formpool is customized in table T80d (transaction GCX2)

In transaction SE38 a copy of template of RGGBR000 (Length: 8 Characters Client dependent) can be performed into e.g. ZGGBR000

User exit definitions:

Customer Includes (zgbbgd00) may cause problems, here any modification is strictly prohibited.
Validation and Substitution in S4 Hana

In case you would like to have information about how this functionality works in S4 Hana environment, please see note 2462725.