

BL Scripting with JavaScript

- Introduction
- Business Rule scripting with JavaScript
 - JS FIX functionality
 - Error handling during script execution
 - Logging in JavaScript
 - print(<text>)
 - Operations with session
 - startSession(<sessionName>)
 - startSession(<senderCompld>,<targetCompld>)
 - startSession(<senderCompld>,<targetCompld>,<sessionQualifier>)
 - disconnectSession(<sessionName>,<reason>)
 - disconnectSession(<senderCompld>,<targetCompld>,<reason>)
 - disconnectSession(<senderCompld>,<targetCompld>,<sessionQualifier>,<reason>)
 - terminateSession(<sessionName>,<reason>)
 - terminateSession(<senderCompld>,<targetCompld>,<reason>)
 - terminateSession(<senderCompld>,<targetCompld>,<sessionQualifier>,<reason>)
 - Operations with group fields
 - getGroup(<fieldTag>)
 - getGroup(<parentGroupHandle>, <entry>, <fieldTag>)
 - removeField(<group handle>, <entry>, <fieldTag>)
 - isGroupValid(<handle>)
 - Operations with message fields
 - removeField(<fieldTag>)
 - createReject("session")
 - isSupportedField(<fieldTag>)
 - isSupportedField(<group handle>, <fieldTag>)
 - getNumField(<fieldTag>)
 - getNumField(<group handle>, <entry>, <fieldTag>)
 - getDateField(<fieldTag>, <format>)
 - getDateField(<group handle>, <entry>, <fieldTag>, <format>)
 - getStringField(<fieldTag>)
 - getStringField(<group handle>, <entry>, <fieldTag>)
 - getMStringField(<fieldTag>)
 - getMStringField(<group handle>, <entry>, <fieldTag>)
 - getRawDataField(<fieldTag>)
 - getRawDataField(<group handle>, <entry>, <fieldTag>)
 - setNumField(<fieldTag>, <new value>)
 - setDateField(<fieldTag>, <format>, <new value>)
 - setStringField(<fieldTag>, <new value>)
 - setMStringField(<fieldTag>, <new value>)
 - setRawDataField(<fieldTag>, <new value>)
 - setNumField(<group handle>, <entry>, <fieldTag>, <new value>)
 - setDateField(<group handle>, <entry>, <fieldTag>, <format>, <new value>)
 - setStringField(<group handle>, <entry>, <fieldTag>, <new value>)
 - setMStringField(<group handle>, <entry>, <fieldTag>, <new value>)
 - setRawDataField(<group handle>, <entry>, <fieldTag>, <new value>)
 - swapFields(<fieldTag1>, <fieldTag2>)
 - swapFields(<group handle field 1>, <entry field 1>, <fieldTag1>, <fieldTag2>)
 - swapFields(<group handle field 1>, <entry field 1>, <fieldTag1>, <group handle field 2>, <entry field 2>, <fieldTag2>)
 - Additional Date operations
 - Create date from current date or using passed arguments.
 - getCurrentDate(<format>)
 - createDate(<format>, <val1>, <val2>, ...<valn>)
 - getCurrentDateStr(<format>)
 - createDateStr(<format>, <val1>, <val2>, ...<valn>)
 - Convert from date to string
 - dateToString(<date handle>)
 - dateToString(<date handle>, <format>)
 - Get part of date - year, month, day, etc.
 - getYear(<date handle>)
 - getMonth(<date handle>)
 - getWeek(<date handle>)
 - getDay(<date handle>)
 - getHour(<date handle>)
 - getMinutes(<date handle>)
 - getSec(<date handle>)
 - getMSec(<date handle>)
 - getNSec(<date handle>)
 - Set date part - year, month, day, etc.
 - setYear(<date handle>, <new val>)
 - setMonth(<date handle>, <new val>)
 - setWeek(<date handle>, <new val>)
 - setDay(<date handle>, <new val>)
 - setHour(<date handle>, <new val>)
 - setMinutes(<date handle>, <new val>)

- setSec(<date handle>, <new val>)
- setMSec(<date handle>, <new val>)
- setNSec(<date handle>, <new val>)
- Change date part with automatic date recalculation
 - changeYear(<date handle>, <new val>)
 - changeMonth(<date handle>, <new val>)
 - changeDay(<date handle>, <new val>)
 - changeHour(<date handle>, <new val>)
 - changeMinutes(<date handle>, <new val>)
 - changeSec(<date handle>, <new val>)
 - changeMSec(<date handle>, <new val>)
 - changeNSec(<date handle>, <new val>)
- Routing functions
 - getSourceClientId()
 - getSourceSessionId()
 - getSourceSessionQualifier()
 - isSessionActive(<Session Name>)
 - convert(<source protocol>, <target protocol>)
 - transform(<target protocol>, <target message type>)
 - getMsgBySeqNum(<SenderCompld>, <TargetCompld> , <SeqNum>)
 - getMsgBySeqNum(<SenderCompld>, <TargetCompld>, <SessionQualifier>, <SeqNum>)
 - send(<SenderCompld>, <TargetCompld>)
 - send(<SenderCompld>, <TargetCompld>, <SessionQualifier>)
 - send(<session source identifier>)
 - handler(<handler name>, <array of additional parameters>)
 - processNetStatusRequest()
 - createNotification (<category>, <reason>)
 - strategySend (<strategyName>)
 - strategyReject (<historyName>, <createBusinessRejectIfFailed>)
 - serializeMessage()
 - serializeMessage(<tagDelimiter>)
 - parseMessage(<msg>)
 - parseMessage(<msg>, <version>)
- History functions
 - saveToHistory (<historyName>, <keyFields>, <commonFields>, <expireDateTime>)
 - updateHistory (<historyName>, <keyFields>, <commonFields>, <expireDateTime>)
 - getFromHistory (<historyName>, <keyFields>, <columnName>)
 - getRecordFromHistory (<historyName>, <keyFields>)
 - removeRecordFromHistory (<historyName>, <keyFields>)
 - removeRecordFromHistoryByCompositeKey (<historyName>, <keyFields>)
- Security functions
 - decryptString(<text>)
 - hashString(<text>)
- Predefined FIX date formats

Introduction

FIXEdge offers flexible solution for Business Rules purposed to provide routing, transferring, and data manipulations for FIX messages that go through the Business Layer of FIXEdge. In addition to XML Rules, there are two ways to enhance the flexibility of message transferring and data manipulation. Those ways are scripting with JavaScript and XSLT. To use this feature properly, scripts should be written in one of the scripting languages and assigned to the Script instruction of Action section section in a Business Rule.

The scripting subsystem gives user the following advantages:

- Saves time spent on business logic implementation
- Flexibility and simplicity of modification
- Does not bring significant performance breakdown

Business Rule scripting with JavaScript

It is enough to mention that the script is a JavaScript in the Script tag, in order to run the script and do the proper transformations. This structure facilitate user with all the standard Javascript features and functionality. In addition to this, it enables user to use functions that provide access to the FIX message inside data, allows manipulations with fields and groups as well as whole FIX messages.

JS FIX functionality

A trivial example of a complete JS script for a business rule is shown below:

```

//take the value the 11th tag from the fix message and get the last 4 characters from it
//assign the new value for the tag 11 with format BBBBSSSSCCYYMMDD
//where
//BBBB - any 4 characters
//SSSS - last 4 characters from the old value of field 11
//CC - century
//YY- year
//MM - month
//DD - day

field_11 = getStringField(11);
if(field_11.length() < 4)
    return;
last4 = field_11.slice(field_11.length() - 4);
currDate = getCurrentDateStr(YYYYMMDDUtc);
century = currDate.slice(1, 3);
century++;
result = "bbbb" + last4 + century + currDate.substr(3);
setStringField(11, result);

```

Error handling during script execution

In case of syntax error in script - script execution will be terminated and failed. There is no way to handle syntax errors in script during script execution. User able to stop script execution using throw statement, the result in this case will be the same. User must be able to verify and handle logical errors using verification functions (like isNaN, isValidMessage(), etc) in condition statements.

Logging in JavaScript	
print(<text>)	
Example Code	Execution of this line results in the following record in the application log:
print (getCurrentDateStr(DATETIMEUtc));	[NOTE] 20131023-08:28:53.651 [4320] [./FixEdge/conf/test.js] - JavaScript '.. /FixEdge/conf/test.js' output: 20131023-08:28:53
Operations with session	
startSession(<sessionName>)	
Starts session "sessionName"	
Example Code	Starts session "StartByEvent"
startSession("StartByEvent")	
startSession(<senderCompId>,<targetCompId>)	
Starts session where SenderCompID = "senderCompID" and TargetCompID = "targetCompID"	
Example Code	Starts session where SenderCompID = "FIXEdge" and TargetCompID = "SimpleClient"
startSession("FIXEdge" , "SimpleClient")	
startSession(<senderCompId>,<targetCompId>,<sessionQualifier>)	
Starts session where SenderCompID = "senderCompID", TargetCompID = "targetCompID",SessionQualifierValue = "sessionQualifier"	

<p>Example Code</p> <pre>startSession("FIXEdge", "SimpleClient", "Q1")</pre>	<p>Starts session where SenderCompID = "FIXEdge", TargetCompID = "SimpleClient", SessionQualifierValue = "Q1"</p>
<p>disconnectSession(<sessionName>,<reason>)</p> <p>Disconnects session "sessionName" with reason "reason"</p>	
<p>Example Code</p> <pre>disconnectSession("DisconnectByEvent", "Disconnect by message")</pre>	<p>Disconnects session "DisconnectByEvent" with reason "Disconnect by message"</p>
<p>disconnectSession(<senderCompId>,<targetCompId>,<reason>)</p> <p>Disconnects session where SenderCompID = "senderCompId" and TargetCompID = "targetCompId" with reason "reason"</p>	
<p>Example Code</p> <pre>disconnectSession("FIXEdge", "SimpleClient", "Disconnect by message")</pre>	<p>Disconnects session where SenderCompID = "FIXEdge" and TargetCompID = "SimpleClient" with reason "Disconnect by message"</p>
<p>disconnectSession(<senderCompId>,<targetCompId>,<sessionQualifier>,<reason>)</p> <p>Starts session where SenderCompID = "senderCompID", TargetCompID = "targetCompID", SessionQualifierValue = "sessionQualifier" with reason "reason"</p>	
<p>Example Code</p> <pre>disconnectSession("FIXEdge", "SimpleClient", "Disconnect by message", "Q1")</pre>	<p>Disconnects session where SenderCompID = "FIXEdge", TargetCompID = "SimpleClient", SessionQualifierValue = "Q1" with reason "Disconnect by message"</p>
<p>terminateSession(<sessionName>,<reason>)</p> <p>Terminates session "sessionName" with reason "reason"</p>	
<p>Example Code</p> <pre>terminateSession("TerminateByEvent", "Terminate by message")</pre>	<p>Terminates session "TerminateByEvent" with reason "Terminate by message"</p>
<p>terminateSession(<senderCompId>,<targetCompId>,<reason>)</p> <p>Terminates session where SenderCompID = "senderCompID" and TargetCompID = "targetCompID" with reason "reason"</p>	
<p>Example Code</p> <pre>terminateSession("FIXEdge", "SimpleClient", "Terminate by message")</pre>	<p>Terminates session where SenderCompID = "FIXEdge" and TargetCompID = "SimpleClient" with reason "Terminate by message"</p>
<p>terminateSession(<senderCompId>,<targetCompId>,<sessionQualifier>,<reason>)</p> <p>Terminates session where SenderCompID = "senderCompID", TargetCompID = "targetCompID", SessionQualifierValue = "sessionQualifier"</p>	
<p>Example Code</p> <pre>terminateSession("FIXEdge", "SimpleClient", "Q1", "Terminate by message")</pre>	<p>Terminates session where SenderCompID = "FIXEdge", TargetCompID = "SimpleClient", SessionQualifierValue = "Q1" with reason "Terminate by message"</p>

Please note:

1. The session will not be visible in FIXICC or manageable (start, disconnect, etc) from FIXICC/BL/JS/scheduler.
2. The termination and the disconnection occur with async way after rule completion. In case of fail <OnRuleFailEvent> will be called.

Operations with group fields

getGroup(<fieldTag>)

Returns group handle for <fieldTag> group field


Example Code

```
hndl = getGroup(78);  
isGroupValid(hndl)
```

Gets handle on group 78 and tests whether the handle 78 is valid

getGroup(<parentGroupHandle>, <entry>, <fieldTag>)

Returns group handle for <fieldTag> group field from <parentGroupHandle>[<entry>] group

 Numeration of <entry> starts with 0.

Example Code

```
hndl = getGroup(552);  
isGroupValid(hndl);  
  
hndl2 = getGroup(hndl, 0, 518);  
isGroupValid(hndl2);  
  
if(hndl2 != hndl){  
    setNumField(552, 4);  
}
```


Gets handle on group 552 and tests whether the handle 552 is valid

Gets handle from the first record of 552 group on group 518 and tests whether the handle 518 is valid

If the group handles are not identical then increase the number of group 552 elements up to 4

removeField(<group handle>, <entry>, <fieldTag>)

Removes field from group field

 Numeration of <entry> starts with 0.

Example Code

```
hndl = getGroup(78);  
isGroupValid(hndl);  
removeField(hndl, 0, 79);
```

Gets handle on group 78 and tests whether the handle 78 is valid

Removing tag 79 from group 78

isGroupValid(<handle>)

Returns true or false depending on the handle value

Example Code

```
//see the example above
```

Operations with message fields

removeField(<fieldTag>)

Removes field from message

Example Code

```
removeField(58);
```

Removing tag 58 from message

createReject("session")

Creating reject

Example code

```
createReject("Target42sohSender42");
```

Creating reject and sending status message 35=3

isSupportedField(<fieldTag>)

Returns true if field is supported in the message

Example Code

```
if ( isSupportedField( 78 ) )  
  
    removeField( 78 );
```

Tests whether the tag is supported in message

Removes tag 78

isSupportedField(<group handle> , <fieldTag>)

Returns true if field is supported in the group

Example Code

```
hndl = getGroup(78);  
  
if ( isGroupValid(hndl) && isSupportedField(hndl,  
79) )  
  
    removeField( hndl , 0 , 79 );
```

Gets handle on group 78.

Tests whether the handle 78 is valid and tag 79 is supported in group
Removes tag 79 from group 78

getNumField(<fieldTag>)

Returns numeric value of the <fieldTag> field

Example Code

```
getNumField(78);
```

returns numeric value of the field 78

getNumField(<group handle>, <entry>, <fieldTag>)

Returns number value of the <fieldTag> field from group <group handle>[<entry>]



Numeration of <entry> starts with 0.

Example Code

```
hndl = getGroup(552);  
  
isGroupValid(hndl);  
  
setNumField(552,getNumField(hndl, 0, 581));
```

Gets handle on group 552 and tests whether the handle 552 is valid.

Gets the numeric value of field 581 and assigns it to the field 552

getDateField(<fieldTag>, <format>)

Returns date handle to the <fieldTag> field value

Example Code

```
getDateField(64, YYYYMMDDLmd);
```

Gets the date value of the field 64 in YYYYMMDD format

getDateField(<group handle>, <entry>, <fieldTag>, <format>)

Returns date handle to the <fieldTag> field value from group <group handle>[<entry>]



Numeration of <entry> starts with 0.

Example Code

```
hndl = getGroup(778);  
isGroupValid(hndl);  
getDateField(hndl, 0, 779, YYYYMMDDLmd);
```

Gets handle on group 778 and tests whether the handle 778 is valid.

Gets the date value of the field 779 in YYYYMMDD format from the first element of group 778

getStringField(<fieldTag>)

Returns string value of the <fieldTag> field

Example Code

```
getStringField(78);
```

Returns string value of the field 78.

getStringField(<group handle>, <entry>, <fieldTag>)

Returns string value of the <fieldTag> field from group <group handle>[<entry>]



Numeration of <entry> starts with 0.

Example Code

```
hndl = getGroup(552);  
isGroupValid(hndl);  
setStringField(552, getStringField(hndl, 0, 581));
```

Gets handle on group 552 and tests whether the handle 552 is valid.

Gets the string value of field 581 and assigns it to the field 552.

getMStringField(<fieldTag>)

Returns array that contains multi string value of the <fieldTag> field

Example Code


```
tmp = getMStringField(18);  
if(3 == tmp.length){  
setStringField(55, tmp[0]);  
setStringField(65, tmp[1]);  
}
```

Gets the multiple string value of the field 18 and assigns it to the variable tmp.

If the number of strings in tmp variable equals to 3, then the value of the first string is assigned to the field 55 and the value of the second string is assigned to the field 65.

getMStringField(<group handle>, <entry>, <fieldTag>)

Returns array that contains multi string value of the <fieldTag> field from group <group handle>[<entry>]

 Numeration of <entry> starts with 0.

Example Code

```
hndl = getGroup(552);  
isGroupValid(hndl);  
tmp = getMStringField(hndl, 0, 18);
```

Gets handle on group 552 and tests whether the handle 552 is valid

Gets the multiple string value of the field 18 - found in the first entry of the group 552 -and assigns it to the variable tmp

getRawDataField(<fieldTag>)

returns array that contains raw data value of the <fieldTag> field


Example Code

```
tmp = getRawDataField(91);
```

Gets the raw data value of the field 91 and assigns it to the variable tmp

getRawDataField(<group handle>, <entry>, <fieldTag>)

Returns array that contains raw data value of the <fieldTag> field from group <group handle>[<entry>]

 Numeration of <entry> starts with 0.

Example Code

```
hndl = getGroup(5521);  
isGroupValid(hndl);  
tmp = getRawDataField(hndl, 0, 91);
```

Gets handle on group 5521 and tests whether the handle 5521 is valid

Gets the raw data value of the field 91 - found in the first entry of the group 5521 -and assigns it to the variable tmp

setNumField(<fieldTag>, <new value>)

assigns value <new value> to the <fieldTag> field as a NUMBER

Example Code

```
//see the sample for the getNumField();
```

setDateField(<fieldTag>, <format>, <new value>)

Assigns value <new value> to the <fieldTag> field as a DATE

Example Code

```
tmp = getDateField(64, YYYYMMDDLmd);  
setYear(tmp, 1979);  
setDateField(64, YYYYMMDDLmd, tmp);
```

Gets the date value of the field 64 in YYYYMMDD format and assigns it to the variable tmp

Sets the year of the date to 1979 in the tmp variable

Assigns the date from tmp variable back to the field 64

setStringField(<fieldTag>, <new value>)

Assigns value <new value> to the <fieldTag> field as a STRING

Example Code

```
// see the example for getMStringField();
```

setMStringField(<fieldTag>, <new value>)

Assigns value <new value> to the <fieldTag> field as a MString

Example Code

```
tmp = ["aa", "abb", "bbbc"];  
setMStringField(18, tmp);
```

Sets the array of strings named tmp

Assigns the array of strings to the multiple string field 18

setRawDataField(<fieldTag>, <new value>)

Assigns value <new value> to the <fieldTag> field as a RawData

Example Code

```
tmp = [12, 56, 0, 34];  
setRawDataField(91, tmp);
```

Sets the array of numbers named tmp

Assigns the array of numbers to the raw data field 91

setNumField(<group handle>, <entry>, <fieldTag>, <new value>)

Assigns value <new value> to the <fieldTag> field from group <group handle>[<entry>] as a NUMBER



Numeration of <entry> starts with 0.

Example Code

```
hdl = getGroup(552);  
isGroupValid(hdl);  
setNumField(hdl, 1, 519, 4);
```

Explanations

Gets handle on group 552 and tests whether the handle 552 is valid

The field 519 from the second entry of group 552 is assigned to 4

setDateField(<group handle>, <entry>, <fieldTag>, <format>, <new value>)

Assigns value <new value> to the <fieldTag> field from group <group handle>[<entry>] as a DATE



Numeration of <entry> starts with 0.

Example Code

```
tmp = getDateField(64, YYYYMMDDLmd);  
  
hdl = getGroup(5521);  
isGroupValid(hdl);  
setDateField(hdl, 0, 64, YYYYMMDDLmd, tmp);
```


Gets the date in format YYYYMMDD from the field 64 and assigns the obtained value to the tmp variable

Gets handle on group 5521 and tests whether the handle 5521 is valid

The field 64 from the first entry of group 5521 is assigned the value from variable tmp

setStringField(<group handle>, <entry>, <fieldTag>, <new value>)

Assigns value <new value> to the <fieldTag> field from group <group handle>[<entry>] as a STRING

 Numeration of <entry> starts with 0.

Example Code

```
tmp = getStringField(581);  
  
hndl = getGroup(5521);  
isGroupValid(hndl);  
setDateField(hndl, 0, 581, tmp);
```


Gets the string value from the field 581 and assigns it to the tmp variable

Gets handle on group 5521 and tests whether the handle 5521 is valid

The field 581 from the first entry of group 5521 is assigned the value from variable tmp

setMStringField(<group handle>, <entry>, <fieldTag>, <new value>)

Assigns value <new value> to the <fieldTag> field from group <group handle>[<entry>] as a MString

 Numeration of <entry> starts with 0.

Example Code

```
tmp = getMStringField(18);  
  
hndl = getGroup(5521);  
isGroupValid(hndl);  
setMStringField(hndl, 0, 18, tmp);
```


Gets the multiple strings value from the field 18 and assigns it to the tmp variable

Gets handle on group 5521 and tests whether the handle 5521 is valid

The field 18 from the first entry of group 5521 is assigned the value from variable tmp

setRawDataField(<group handle>, <entry>, <fieldTag>, <new value>)

Assigns value <new value> to the <fieldTag> field from group <group handle>[<entry>] as a RawData

 Numeration of <entry> starts with 0.

Example Code

```
tmp = getRawDataField(91);  
  
hndl = getGroup(5521);  
isGroupValid(hndl);  
setRawDataField(hndl, 0, 91, tmp);
```



Gets the multiple raw data value from the field 91 and assigns it to the tmp variable

Gets handle on group 5521 and tests whether the handle 5521 is valid

The field 91 from the first entry of group 5521 is assigned the value from variable tmp

swapFields(<fieldTag1>, <fieldTag2>)

Exchanges values of fields <fieldTag1> and <fieldTag2>

<p>Example Code</p> <pre>swapFields(63, 573);</pre>	<p>Values of field 63 and field 573 was exchanged</p>
<p>swapFields(<group handle field 1>, <entry field 1>, <fieldTag1>, <fieldTag2>)</p> <p>Exchanges values of fields <fieldTag1> (from group <group handle field 1>[<entry field 1>]) and <fieldTag2> (which can be both from group <group handle field 1>[<entry field 1>] and outside of it)</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #fff9c4;">  Numeration of <entry field 1> starts with 0. </div>	
<p>Example Code</p> <pre>hndl = getGroup(552); isGroupValid(hndl); swapFields(hndl, 1, 11, 572);]</pre>	<p>Gets handle on group 552 and tests whether the handle 552 is valid</p> <p><i>Exchanges values of field 11 in the second entry of the group 552 and field 572 outside of it</i></p>
<p>swapFields(<group handle field 1>, <entry field 1>, <fieldTag1>, <group handle field 2>, <entry field 2>, <fieldTag2>)</p> <p>Exchanges values of fields <fieldTag1> (from group <group handle field 1>[<entry field 1>]) and <fieldTag2> (from group <group handle field 2>[<entry field 2>])</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #fff9c4;">  Numeration of <entry field 1> and <entry field 2> starts with 0. </div>	
<p>Example Code</p> <pre>hndl = getGroup(552); isGroupValid(hndl); swapFields(hndl, 1, 11, hndl, 1, 37);</pre>	<p>Gets handle on group 552 and tests whether the handle 552 is valid</p> <p>Exchanges values of field 11 in the second entry and field 37 in the first entry of the fro up 552</p>

Additional Date operations

<p>Create date from current date or using passed arguments.</p>	
<p>getCurrentDate(<format>)</p> <p>Returns handle that encapsulates current date in format <format></p>	
<p>Example Code</p> <pre>getCurrentDate(DATETIMEUtc);</pre>	<p>Creates the date data of DATETIMEUtc format</p>
<p>createDate(<format>, <val1>, <val2>, ...<valn>)</p> <p>Returns handle that encapsulates date in format <format> constructed using values <val1>, <val2>, ...<valn></p>	
<p>Example Code</p> <pre>tmp = createDate(YYYYMMDD, 1995, 12, 23); setStringField(55, dateToString(tmp));</pre>	<p>Creates date of YYYYMMDD format with corresponding values and assigns it to the tmp variable converts the value of tmp variable to string and assigns this string to be the value of field 55</p>

getCurrentDateStr(<format>)

Returns current date in format <format>

Example Code

```
setStringField(55,  
getCurrentDateStr(DATETIMEUtc));
```

Gets current date as a string and assigns this string to be the value of field 55

createDateStr(<format>, <val1>, <val2>, ...<valn>)

Returns date in format <format> constructed using values <val1>, <val2>, ...<valn>

Example Code

```
tmp = createDateStr(YYYYMMDD,  
1995, 12, 23);]
```

Creates date of YYYYMMDD format with corresponding values returns this date as string and assigns it to the tmp variable

Convert from date to string

dateToString(<date handle>)

Returns date in format that was mentioned in getCurrentDate() or createDate() call

Example Code

```
//see the example of createDate()
```

dateToString(<date handle>, <format>)

Returns date in format <format>

Example Code

```
tmp = createDate(YYYYMMDD, 1995,  
12, 23);  
  
setStringField(55, dateToString  
(tmp, YYYYMM));
```

Creates date from given data in one format and assigns it to the tmp variable

The field 55 is assigned the value of variable tmp

Get part of date - year, month, day, etc.

getYear(<date handle>)

Returns year of date <date handle>

Example Code

```
tmp = getCurrentDate(DATETIMEUtc);  
  
tmp2 = getYear(tmp);  
  
setDateField(64, YYYYMMDDLmd,  
tmp2);
```

Creates date handle of the current date

Extracts the year from tmp variable and assigns it to the variable tmp2

Sets the value of the field 64 to the beginning of current year

getMonth(<date handle>)

Returns month of date <date handle>

Example Code

```
tmp = getCurrentDate(DATETIMEUtc);  
  
tmp3 = getMonth(tmp);
```

Creates date handle of the current date

Extracts the month value from tmp variable and assigns it to the variable tmp3

getWeek(<date handle>)

Returns weekof date <date handle>

Example code

```
tmp = createDate(YYYYMMWW, 1995, 12, 2);  
setStringField(55, getWeek(tmp));
```

Creates date handle <date handle>

Sets the value of the tmp variable to the field 55

getDay(<date handle>)

Returns day of date <date handle>

Example Code

```
tmp = getCurrentDate(DATETIMEUtc);  
tmp4 = getDay(tmp);
```

Creates date handle of the current date

Extracts the day value from tmp variable and assigns it to the variable tmp4

getHour(<date handle>)

Returns hour of date <date handle>

Example Code

```
tmp = getCurrentDate(DATETIMEUtc);  
tmp5 = getHour(tmp);
```

Creates date handle of the current date

Extracts the hour value from tmp variable and assigns it to the variable tmp5

getMinutes(<date handle>)

Returns minutes of date <date handle>

Example Code

```
tmp = getCurrentDate(DATETIMEUtc);  
tmp6 = getMinutes(tmp);
```

Creates date handle of the current date

Extracts the minutes value from tmp variable and assigns it to the variable tmp6

getSec(<date handle>)

Returns seconds of date <date handle>

Example Code

```
tmp = getCurrentDate(DATETIMEUtc);  
tmp7 = get Sec(tmp);
```

Creates date handle of the current date

Extracts the seconds value from tmp variable and assigns it to the variable tmp7

getMSec(<date handle>)

Returns milliseconds of date <date handle>

Example Code

```
tmp = getCurrentDate(DATETIMEUtc);  
tmp7 = getMSec(tmp);
```

Creates date handle of the current date

Extracts the mile seconds value from tmp variable and assigns it to the variable tmp7

getNSec(<date handle>)

Returns nanoseconds of date <date handle>. Function is available since **FIXEdge 6.2.0**.

Example Code

```
tmp = getCurrentDate  
(DATETIMENanoUtc);  
tmp7 = getNSec(tmp);
```

Creates date handle of the current date

Extracts the nanoseconds value from tmp variable and assigns it to the variable tmp7

Set date part - year, month, day, etc.

setYear(<date handle>, <new val>)

Assign new value <new val> to year of date <date handle>



Validation rule

New value must be greater than 1970 and less than 2038

Example Code

```
tmp = getDateField(64,
YYYYMMDDLmd);
setYear(tmp, 1979);
setDateField(64, YYYYMMDDLmd, tmp);
```

Gets the date value of the field 64 in YYYYMMDD format and assigns it to the variable tmp

Sets the year of the date to 1979 in the tmp variable

Assigns the date from tmp variable back to the field 64

setMonth(<date handle>, <new val>)

Assign new value <new val> to month of date <date handle>



Validation rule

New value must be greater than or equal 1 and less than or equal 12

Example Code

```
tmp = getDateField(64,
YYYYMMDDLmd);
setMonth(tmp, 06);
setDateField(64, YYYYMMDDLmd, tmp);
```

Gets the date value of the field 64 in YYYYMMDD format and assigns it to the variable tmp

Sets the month of the date to 06 in the tmp variable

Assigns the date from tmp variable back to the field 64

setWeek(<date handle>, <new val>)

Assign new value <new val> to week of date <date handle>



Validation rule

New value must be greater than or equal 1 and less than or equal 5

Example Code

```
tmp = createDate(YYYYMMWW, 1995, 12, 2);
setWeek(tmp, 4);
setStringField(55, dateToString(tmp));
```

Gets the date value of the field 64 in YYYYMMWW format and assigns it to the variable tmp

Sets the week of the date to 4 in the tmp variable

Assigns the date from tmp variable to the field 55 in string format




setDay(<date handle>, <new val>)

Assign new value <new val> to day of date <date handle>



Validation rule

New value must be greater than or equal 1 and less than or equal 31

<p>Example Code</p> <pre>tmp = getDateField(64, YYYYMMDDLmd); setDay(tmp, 27); setDateField(64, YYYYMMDDLmd, tmp);</pre>	<p>Gets the date value of the field 64 in YYYYMMDD format and assigns it to the variable tmp</p> <p>Sets the day of the date to 27 in the tmp variable</p> <p>Assigns the date from tmp variable back to the field 64</p>
<p>setHour(<date handle>, <new val>)</p> <p>Assign new value <new val> to hour of date <date handle></p> <div data-bbox="142 447 1479 554" style="border: 1px solid #ccc; padding: 5px; background-color: #fff9c4;"> <p> Validation rule</p> <p>New value must be greater than or equal 0 and less than 24</p> </div>	
<p>Example Code</p> <pre>tmp = getDateField(64, DATETIMEUtc); setHour(tmp, 10); setDateField(64, DATETIMEUtc, tmp);</pre>	<p>Gets the date value of the field 64 in DATETIMEUtc format and assigns it to the variable tmp</p> <p>Sets the hour of the date to 10 in the tmp variable</p> <p>Assigns the date from tmp variable back to the field 64</p>
<p>setMinutes(<date handle>, <new val>)</p> <p>Assign new value <new val> to minutes of date <date handle></p> <div data-bbox="142 951 1479 1058" style="border: 1px solid #ccc; padding: 5px; background-color: #fff9c4;"> <p> Validation rule</p> <p>New value must be greater than or equal 0 and less than or equal 59</p> </div>	
<p>Example Code</p> <pre>tmp = getDateField(64, DATETIMEUtc); setMinutes(tmp, 20); setDateField(64, DATETIMEUtc, tmp);</pre>	<p>Gets the date value of the field 64 in DATETIMEUtc format and assigns it to the variable tmp</p> <p>Sets the minutes of the date to 20 in the tmp variable</p> <p>Assigns the date from tmp variable back to the field 64</p>
<p>setSec(<date handle>, <new val>)</p> <p>Assign new value <new val> to seconds of date <date handle></p> <div data-bbox="142 1455 1479 1562" style="border: 1px solid #ccc; padding: 5px; background-color: #fff9c4;"> <p> Validation rule</p> <p>New value must be greater than or equal 0 and less than or equal 59</p> </div>	
<p>Example Code</p> <pre>tmp = getDateField(64, DATETIMEUtc); setSec(tmp, 20); setDateField(64, DATETIMEUtc, tmp);</pre>	<p>Gets the date value of the field 64 in DATETIMEUtc format and assigns it to the variable tmp</p> <p>Sets the seconds of the date to 20 in the tmp variable</p> <p>Assigns the date from tmp variable back to the field 64</p>

setMSec(<date handle>, <new val>)

Assign new value <new val> to milliseconds of date <date handle>



Validation rule

New value must be greater than or equal 0 and less than or equal 59

Example Code

```
tmp = getDateField(64,
DATETIMEUtc);

setMSec(tmp, 50);

setDateField(64, DATETIMEUtc, tmp);
```

Gets the date value of the field 64 in DATETIMEUtc format and assigns it to the variable tmp

Sets the milliseconds of the date to 50 in the tmp variable

Assigns the date from tmp variable back to the field 64

setNSec(<date handle>, <new val>)

Assign new value <new val> to nanoseconds of date <date handle>. Function is available since **FIXEdge 6.2.0**.

Example Code

```
tmp = getDateField(64,
DATETIMENanoUtc);

setNSec(tmp, 50);

setDateField(64, DATETIMENanoUtc,
tmp);
```

Gets the date value of the field 64 in DATETIMENanoUtc format and assigns it to the variable tmp

Sets the nanoseconds of the date to 50 in the tmp variable

Assigns the date from tmp variable back to the field 64

Change date part with automatic date recalculation

changeYear(<date handle>, <new val>)

Add value <new val> to year of date <date handle>

Example Code

```
tmp = getDateField(64,
DATETIMEUtc);

changeYear(tmp, 2);

setDateField(64, DATETIMEUtc, tmp);
```

Gets the date value of the field 64 in DATETIMEUtc format and assigns it to the variable tmp

Adds 2 years to the value of the date in the tmp variable

Assigns the date from tmp variable back to the field 64

changeMonth(<date handle>, <new val>)

Add value <new val> to month of date <date handle>

Example Code

```
tmp = getDateField(64,
DATETIMEUtc);

changeMonth(tmp, 5);

setDateField(64, DATETIMEUtc,
tmp);
```

Gets the date value of the field 64 in DATETIMEUtc format and assigns it to the variable tmp

Adds 5 months to the value of the date in the tmp variable

Assigns the date from tmp variable back to the field 64

changeDay(<date handle>, <new val>)

Add value <new val> to day of date <date handle>

Example Code

```
tmp = getDateField(64,
DATETIMEUtc);

changeDay(tmp, 10);

setDateField(64, DATETIMEUtc,
tmp);
```

Gets the date value of the field 64 in DATETIMEUtc format and assigns it to the variable tmp

Adds 10 days to the value of the date in the tmp variable

Assigns the date from tmp variable back to the field 64

changeHour(<date handle>, <new val>)

Add value <new val> to hour of date <date handle>

Example Code

```
tmp = getDateField(64,
DATETIMEUtc);

changeHour(tmp, 24);

setDateField(64, DATETIMEUtc, tmp);
```

Gets the date value of the field 64 in DATETIMEUtc format and assigns it to the variable tmp

Adds 24 hours to the value of the date in the tmp variable

Assigns the date from tmp variable back to the field 64

changeMinutes(<date handle>, <new val>)

Add value <new val> to minutes of date <date handle>

Example Code

```
tmp = getDateField(64,
DATETIMEUtc);

changeMinutes(tmp, 600);

setDateField(64, DATETIMEUtc,
tmp);
```

Gets the date value of the field 64 in DATETIMEUtc format and assigns it to the variable tmp

Adds 600 minutes to the value of the date in the tmp variable

Assigns the date from tmp variable back to the field 64

changeSec(<date handle>, <new val>)

Add value <new val> to seconds of date <date handle>

Example Code

```
tmp = getDateField(64,
DATETIMEUtc);

changeSec(tmp, 3600);

setDateField(64, DATETIMEUtc, tmp);
```

Gets the date value of the field 64 in DATETIMEUtc format and assigns it to the variable tmp

Adds 3600 seconds to the value of the date in the tmp variable

Assigns the date from tmp variable back to the field 64

changeMSec(<date handle>, <new val>)

Assign new value <new val> to milliseconds of date <date handle>

Example Code

```
tmp = getDateField(64,
DATETIMEUtc);

changeMSec(tmp, 36);

setDateField(64, DATETIMEUtc, tmp);
```

Gets the date value of the field 64 in DATETIMEUtc format and assigns it to the variable tmp

Adds 36 milliseconds to the value of the date in the tmp variable

Assigns the date from tmp variable back to the field 64

changeNSec(<date handle>, <new val>)

Assign new value <new val> to nanoseconds of date <date handle>. Function is available since **FIXEdge 6.2.0**.

Example Code

```
tmp = getDateField(64,
DATETIMENanoUtc);

changeNSec(tmp, 36);

setDateField(64, DATETIMENanoUtc,
tmp);
```

Gets the date value of the field 64 in DATETIMENanoUtc format and assigns it to the variable tmp

Adds 36 nanoseconds to the value of the date in the tmp variable

Assigns the date from tmp variable back to the field 64

Routing functions

getSourceClientId()

Returns Client ID of source

Example Code

```
clientId = getSourceClientId();
```

Gets the TA client ID of the session context where script is applied

Note: Returns NULL value, if session is not TA client

getSourceSessionId()

Returns Session ID of source

Example Code

```
clientID = getSourceSessionId();
```

Gets the FIX session ID of the session context where script is applied

Note: Returns NULL value, if session is not FIX session.

getSourceSessionQualifier()

Returns the name of SessionQualifier (String) in case if SessionQualifier is available for source session

Example Code

```
sessionQualifier =  
getSourceSessionQualifier();
```

Gets the SessionQualifier name of the session context where script is applied

isSessionActive(<Session Name>)

Checks is FIX Session with name <Session Name> active

Example Code

```
active = isSessionActive  
( "SOME_SESSION" );  
  
if (active)  
  
    print("SOME_SESSION  
active");
```

Checks is FIX Session with name "SOME_SESSION" active

Prints message.

convert(<source protocol>, <target protocol>)

Convert message from <source protocol> to <target protocol>

Example Code

```
convert("FIX.4.4", "FIX.4.2");
```

Converts message in the given script context, replaces source FIX44 message with converted to FIX42.

Note: please refer to BusinessLayer.dtd to see applicable names of protocols

transform(<target protocol>, <target message type>)

Transform message from current script context to<target protocol> <target message type> message

Example Code

```
transform("FIX.4.2", "8");
```

Transforms message in the given script context, replaces source message with transformed to FIX42 MsgType="8".

Note: please refer to BusinessLayer.dtd to see applicable names of protocols

getMsgBySeqNum(<SenderCompId>, <TargetCompId> , <SeqNum>)

Search for message in session identified as <SenderCompId> and <TargetCompId> pair logs by < SeqNum>

getMsgBySeqNum(<SenderCompId>, <TargetCompId>, <SessionQualifier>, <SeqNum>)

Search for message in session identified as <SenderCompID>/<TargetCompID>/<SessionQualifier> set logs by < SeqNum>

Example Code

```
getMsgBySeqNum("TestSender",  
"TestTarget", 12345);
```

Searches for message with given SeqNum in session identified by 'sender|target' pair.

Note: FIX Session where search occurs must be active and use persistent storage.

send(<SenderCompID>, <TargetCompID>)

Send a message into session identified as <SenderCompID> and <TargetCompID> pair.

send(<SenderCompID>, <TargetCompID>,<SessionQualifier>)

Send a message into session identified as <SenderCompID>/<TargetCompID>/<SessionQualifier> set.

send(<session source identifier>)

Send a message into session that identified by <sessions source identifier>.

Example Code

```
send( "sender", "target" );  
send( "exchange" );
```

Sends a message into session identified by 'sender|target' pair.

Sends a message into session identified by 'exchange' source identifier.

handler(<handler name>, <array of additional parameters>)

Route message into handler <handler name> with array of additional parameters as JSONObject (optional)

Example Code

```
params = [{"Storage", "file.txt"}, {"ClearTime",  
"22:00"}]  
handler( "CMEHandler", params );
```

Routes message in the handler "CMEHandler" and sets additional parameters for processing:

Storage = file.txt, ClearTime = 22:00

processNetStatusRequest()

Response Network Status Response message ("BD") on request

Example Code

```
processNetStatusRequest();
```

Replaces in the routing a Network Status Request message ("BC") with Response ("BD")

createNotification (<category>, <reason>)

Creates notification message ("C") with <category>, <reason> then puts it into the routing

Example Code

```
createNotification( "E", "Session is  
rejected" );
```

Creates notification message ("C") with "Status = ERROR", "Code=201", "Reason = Session is rejected" in the three entries of group (58).

Note: <category> apply values "N" – NOTE, "W" – WARNING, "E" – ERROR, "F" - FATAL

strategySend (<strategyName>)

Processes a message by strategy that identified by <strategyName>

Example Code

```
strategySend( "OrderFlow5" );
```

Processes a some message by the default routing strategy with the name "OrderFlow5".

Note: only default routing strategies can be used for processing: deliveryToStrategy, OrderFlowStrategy and WheelOrderFlowStrategy.

strategyReject (<historyName>, <createBusinessRejectIfFailed>)

Creates reject on message that was stored in history <historyName>, or BusinessReject (MsgType = 'j'), if <createBusinessRejectIfFailed> is 'true' and rejecting has fail. <createBusinessRejectIfFailed> accepts 'false' by default if is absent in parameters.

Example Code

```
strategyReject( "orders", true );
```

Reject a message that was stored by some strategy in the history 'orders'. Creates 'j' Business Reject when message rejecting has failed.

Notes: Method using only in the default routing. Searching in history processes by 'CIOrdID' key in the original message if its type: 'D', 'AB' or by 'OrigCIOrdID' key in original message if its type: 'AC', 'F', 'G'.

Unapplicable for other messages.

serializeMessage()

Returns the string view of the current message. SOH will be used as a delimiter.

Example Code

```
msg = serializeMessage();
```

Returns the string view of the message where SOH will be used as a delimiter and saves it into "msg" variable.

serializeMessage(<tagDelimiter>)

Returns the string view of the current message. Specified symbol will be used as a delimiter.

Example Code

```
msg = serializeMessage("|");
```

Returns the string view of the message where "|" will be used as a delimiter and saves it into "msg" variable.

parseMessage(<msg>)

Parses the specified string message and replaces the current message with the parsed one. After that the old message will be lost. <msg> is a string view of the message. Only messages with SOH as a delimiter can be parsed.

Example Code

```
parseMessage("8=FIX.4.4
\x019=56\x0135=b\x0149=TESTI\x0156=
TESTA\x0134=14\x0152=20030204-09:
25:43\x01297=0\x0110=139\x01");
```

Parses message from the specified string and replaces the current message with it.

parseMessage(<msg>, <version>)

Parses the specified string message using specified FIX protocol and replaces the current message with the parsed one. After that the old message will be lost. <msg> is a string view of the message and <version> is a session version in FIXEdge.properties. Only messages with SOH as a delimiter can be parsed.

Example Code

```
parseMessage("8=FIX.4.4
\x019=56\x0135=b\x0149=TESTI\x0156=
TESTA\x0134=14\x0152=20030204-09:
25:43\x01297=0\x0110=139\x01", "FIX44
");
```

Parses message from the specified string using FIX44 protocol and replaces the current message with it.

Note:

If the 1128 tag is specified in the message for parseMessage(<msg>, <version>) function then the value of the tag has higher priority during the parsing than the version value, for instance:

```
1128=7 and FIXT11:FIX44 and the message has 1092 tag from FIX
5.0:
parseMessage("8=FIXT.1.1
\x019=114\x0135=D\x011128=7\x0149=TESTI\x0156=TESTA\x0134=2\x0152
=20081023-17:03:
05\x0111=90001008\x0155=IBM\x0154=1\x0160=20080717-10:00:
00\x0138=4000\x0140=1\x011092=0\x0110=171\x01", "FIXT11:FIX44");
- the message is parsed successfully as FIX 5.0.
```

```
1128=6 and FIXT11:FIX50 and the message has 1092 tag from FIX
5.0:
parseMessage("8=FIXT.1.1
\x019=114\x0135=D\x011128=6\x0149=TESTI\x0156=TESTA\x0134=2\x0152
=20081023-17:03:
05\x0111=90001008\x0155=IBM\x0154=1\x0160=20080717-10:00:
00\x0138=4000\x0140=1\x011092=0\x0110=171\x01", "FIXT11:FIX50");
- the following error appears:
2017-09-13 11:28:01,558 UTC ERROR [JS_interpreter] 3760
C:\B2BITS\FIXEdge\FIXEdge1\conf\parseMessage.js: failed with
error: Tag 1092 is not defined for this message type. Parsing
stopped at column: 136 in message D with sequence number 2.
```



Please note: the next action in the rule will receive the replaced message!

History functions

saveToHistory (<historyName>, <keyFields>, <commonFields>, <expireDateTime>)

Creates a record in history <historyName> with content: <keyFields> – array of composite key or single key, <commonFields> - array of common data fields, <expireDateTime> - timestamp with an expire date-time for the ClearHistory procedure. Returns true if the function succeeded and returns false otherwise.

Example Code:

```
key = new Array("snd", "tgt");

common = new Array("ClOrdID", "dest");

saveToHistory("orders", key, common, "20101010-10:52:00");
```

Creates new record in history "orders" with fields set: (snd,tgt,ClOrdID,dest,20101010-10:52:00). Record will be deleted automatically when ClearHistory task will executed after date-time 20101010-10:52:00.



Note:

- If <expireDateTime> field is not defined in the history definition, use empty string for it in script so that it works.
- If <expireDateTime> is defined in the history definition and it is an empty string, then it will be treated as NULL and [ClearHistory](#) procedure will delete such a record.



Note:

If <keyFields> is composite then all fields in the key set should be filled.

updateHistory (<historyName>, <keyFields>, <commonFields>, <expireDateTime>)

Modifies fields in the history <historyName> with values specified in <commonFields> and <expireDateTime>. Record for update is searching by <keyFields> criteria. Returns true if the function succeeded and returns false otherwise.

Example Code:

```
key = new Array("snd", "tgt");

common = new Array("", "dest2");

updateHistory("orders", key, common, "20121221-10:52:00");
```

If a record with keys set (snd, tgt) was created by the previous sample for saveToHistory command then field value 'dest' replaces with 'dest2', field value '20101010-10:52:00' replaces with '20121221-10:52:00'.



Note:

- If <expireDateTime> field is not defined in the history definition, use an empty string for it in the script so that it works.
- If <expireDateTime> is defined in the history definition and it is an empty string, then it will be treated as NULL and [ClearHistory](#) procedure will delete a record.



Note: If <keyFields> is composite, then all fields in the key set should be filled. Empty fields in <commonFields> or <expireDateTime> will not be changed.

getFromHistory (<historyName>, <keyFields>, <columnName>)

Retrieves field value in the history <historyName>. Searching criteria is the <keyFields> for a record and <columnName> for column where value reads. Returns target field value, but if such value wasn't found, returns 'undefined'.

Example Code:

```
key = new Array("snd", "tgt");

time = getFromHistory("orders", key, "ExpireDateTime");
```

Returns value '20121221-10:52:00' from the record with key (snd,tgt) that has been updated by previous sample.

Notes: column that created by <expireDateTime> parameter always has name "ExpireDateTime" in a history.

Other columns have names as described in history description in BL_Config.xml or as tag number in string representation if column name is absent in history description.

getRecordFromHistory (<historyName>, <keyFields>)

Retrieves fields array from the record with key <keyFields>, located in the history <historyName>. Returns target record, but if such record wasn't found, returns 'undefined'.

<p>Example Code:</p> <pre>key = new Array("snd", "tgt"); record = getFromHistory("orders", key); print(record);</pre>	<p>Returns fields array of record with key (snd,tgt) then prints content into log. The result will shows "snd, tgt, CIOrdID, dest2, 20121221-10:52:00" if record (snd, tgt) has been saved and updated by previous samples.</p> <p>Notes: Empty fields have a NULL values and printing represents it as "<null>". If <keyFields> is composite then all fields in the key set should be filled.</p>
---	--

removeRecordFromHistory (<historyName>, <keyFields>)

Removes entire record with key <keyFields>, located in the history <historyName>. Returns a number of deleted records.

<p>Example Code:</p> <pre>key = new Array("snd", "tgt"); removeRecordFromHistory("orders", key);</pre>	<p>Removes record with key (snd, tgt) from the history 'orders'.</p> <p>Notes: If <keyFields> is composite then all fields in the key set should be filled.</p>
---	---

removeRecordFromHistoryByCompositeKey (<historyName>, <keyFields>)

Removes entire record by matching of composite key with <keyFields>, located in the history <historyName>. Returns a number of deleted records.

<p>Example Code:</p> <pre>key = new Array("snd", ""); ndeleted = removeRecordFromHistoryByCompositeKey ("orders", key);</pre>	<p>Removes all records with keys which have first field "snd" in a key.</p> <p>Notes: the composite <keyFields> set should contain at least one field. All empty fields in a key will be ignored in searching.</p>
--	--

Security functions



Available since FIXEdge 5.10.1

decryptString(<text>)

Decrypts string using AES 128-bit encryption mechanism. It is useful for encrypted tags (554, 925) in Logon messages.

<p>Example Code:</p> <pre>encryptedPass = getStringField ("554"); pass = decryptString (encryptedPass);</pre>	<p>Decrypting 554 tag.</p>
---	----------------------------

hashString(<text>)

Calculates SHA256 hash for the sting.

<p>Example Code:</p> <pre>hash = hashString("MyPassword");</pre>	<p>Getting hash for string "MyPassword".</p> <p>Note: string for calculating hash should be not "null".</p>
--	---

Predefined FIX date formats

1. **HHMMSSUtc** for the UTC "HH:MM:SS" date format
2. **HHMMSSsssUtc** or **HHMMSSMilliUtc** for the UTC "HH:MM:SS.sss" date format
3. **HHMMSSMicroUtc** for the UTC "HH:MM:SS.ssssss" date format
4. **HHMMSSNanoUtc** for the UTC "HH:MM:SS.ssssssss" date format
5. **YYYYMM** for the "YYYYMM" date format
6. **YYYYMMDD** for the "YYYYMMDD" date format
7. **YYYYMMWW** for the "YYYYMMWW" date format, where WW - number of week "w1"... "w4"
8. **DATETIMEUtc** for the UTC "YYYYMMDD-HH:MM:SS" date format
9. **DATETIMEsssUtc** or **DATETIMEMilliUtc** for the UTC "YYYYMMDD-HH:MM:SS.sss" date format
10. **DATETIMEMicroUtc** for the UTC "YYYYMMDD-HH:MM:SS.ssssss" date format
11. **DATETIMENanoUtc** for the UTC "YYYYMMDD-HH:MM:SS.ssssssss" date format
12. **YYYYMMDDLmd** for the local market "YYYYMMDD" date format