



Printing With Specify 6's Scriptlet Functions

Specify Software Project Staff
25 June 2010
Version 1.2

Specify 6 uses JasperReports as its Open Source printing engine. JasperReports is written in Java and provides a feature for extending the functionality to include custom Java functions, this additional code is encapsulated into a 'Scriptlet' and the functions can be called from within the report. In simpler terms, JasperReports enables reports and labels to make special function calls to format data. The invoice example that ships with Specify 6 already calls some of the available functions. For example, the following 'code' is used to take the First, Last, and Middle name string fields and format them into a single string:

```
((edu.ku.brc.specify.config.Scriptlet)$P{REPORT_SCRIPTLET}).buildNameString($F{FirstName},  
$F{LastName}, $F{MiddleInitial})
```

Configuring Your Report For Scriptlets

The report must know where to get the scriptlet and this is set in the 'Report Properties.' In the Specify Reports application click on the 'Edit' menu and then 'Report Properties.' The dialog will have a tab named 'Scriptlet Class', click on the tab. In the dropdown list select 'Use this scriptlet class...' and then enter the following value:

```
edu.ku.brc.specify.config.Scriptlet
```

Making Sure The Types Are Correct

The fields which come from the columns in the query that used to fill the report. Each column/field contains a certain type of data for example a string, a double, integer, etc. this is referred to as the field's 'type.' If you look at a field's properties you will see the 'type' value next to the label 'Field Class Type.' A field containing text will have the value 'java.lang.String.'

The textfields that are placed in the report, usually from the fields in the query, also have a 'type.' If you click on a textfield and review the 'Text Field' properties the 'Exp. Class' contains the 'type' that the text field is expecting. If the field's type does not match the type that the text field is set to, then an error will occur. Most of the Scriptlet functions accept one type of value and then returns a java.lang.String. For example, the 'getDirChar' function can accept a BigDecimal value for the latitude or longitude and returns a string for the direction. This means that the query field will be a BigDecimal and then the text field in the report must be set to 'java.lang.String' because the getDirChar return a string.

How To Call a Function

JasperReports enables there to be any number of scriptlets defined and used when the report is 'run.' The Specify scriptlet must be identified so JasperReports can 'hook up' to the correct functions. The following string proceeds all functions calls:

```
((edu.ku.brc.specify.config.Scriptlet)$P{REPORT_SCRIPTLET}).
```

Note: The period is required before entering the name of the function, in the example above the name is `buildNameString`. The reminder inside the parentheses is a list of fields that the function requires:
`($F{FirstName}, $F{LastName}, $F{MiddleInitial})`

Available Functions

Function Name	Description	Output	Parameters
format	Formats a 'float' number	String	#.####
getDirChar	Returns a string with a single character for the direction of the Latitude or Longitude: 'N', 'S', 'E', or 'W' (Localized)	String	<ol style="list-style-type: none"> 1. Float - The Latitude or the Longitude value 2. Boolean - True if the first param is a Latitude
getDirChar	Returns a string with a single character for the direction of the Latitude or Longitude: 'N', 'S', 'E', or 'W' (Localized)	String	<ol style="list-style-type: none"> 1. String - The Latitude or the Longitude value 2. Boolean - True if the first param is a Latitude
getDirChar	Returns a string with a single character for the direction of the Latitude or Longitude: 'N', 'S', 'E', or 'W' (Localized)	String	<ol style="list-style-type: none"> 1. BigDecimal - The Latitude or the Longitude value 2. Boolean - True if the first param is a Latitude
formatLatLon	Returns a formatted Lat or Lon with a trailing directions character	String	<ol style="list-style-type: none"> 1. BigDecimal - The Latitude or the Longitude value 2. OriginalLatLonUnit - From the database 3. Boolean - True if the first param is a Latitude
localityDB	Formats a LocalityName field, it's Lat and Lon with OriginalLatLonUnit	String	<ol style="list-style-type: none"> 1. String - The LocalityName 2. BigDecimal - The Latitude value from the database 3. BigDecimal - The Longitude value from the database 4. OriginalLatLonUnit - From the database
locality	Formats a LocalityName field, it's Lat and Lon with OriginalLatLonUnit	String	<ol style="list-style-type: none"> 1. String - The LocalityName 2. Float - The Latitude value from the database 3. Float - The Longitude value

			from the database 4. OriginalLatLonUnit - From the database
formatFieldNo	This is actually a generic function that accepts a string and returns the same string if it not null or return and empty string when null.	String	1. String - Any string field
buildNameString	Returns a string formatted "Last, First Middle" and it checks for null or empty strings to make sure it looks correct.	String	1. String - First Name 2. String - Last Name 3. String - Middle
buildLocalityString	Builds a comma separated string from the fields.	String	1. String - Geography Name 2. String - Locality Name 3. String - Latitude 4. String - Longitude
dateDifference	Returns a localized string "xx months" where xx is the difference in months from the start date to the end date.	String	1. java.sql.Date - Start Date 2. java.sql.Date - End Date
formatDate	Returns a date formatted into a string. The second parameter is the standard Java date format characters.	String	1. java.sql.Date - A date 2. String format (e.g. yyyy-MM-dd)
formatDate	Returns a date formatted into a string. The second parameter is the standard Java date format characters.	String	1. java.util.Date - A date 2. String format (e.g. yyyy-MM-dd)
getCurrentDate	Returns the current date formatted into a string. The parameter is the standard Java date format characters.	String	1. String format (e.g. yyyy-MM-dd)
getCollectors	Returns a string listing the collectors for a given Collecting Event ID.	String	1. Integer - Collecting Event ID

(NOTE: Java allows functions to have the same name as long as the parameters are of different types)

Additional Examples

Getting the Current Date as a String

```
((edu.ku.brc.specify.config.Scriptlet)$P{REPORT_SCRIPTLET}).formatDate(new java.sql.Date(java.util.Calendar.getInstance().getTime().getTime()), "yyyy-MM-dd")
```