# 1、DEE API

## 1.1、com.seeyon.v3x.dee

### 1.1.1、Document

**简要描述：**

* com.seeyon.v3x.dee.Document
* Document是DEE任务中数据流转对象
* 每次DEE任务执行都会有一个内置的document变量贯穿始终
* 在脚本适配器中可以直接使用document变量

**Document格式**

|  |
| --- |
| <root>  <table1>  <row>  <field1>value</field1><field2>value</field2><fieldN>value</fieldN>...  </row>  <row>  <field1>value</field1><field2>value</field2><fieldN>value</fieldN>...  </row>  ...  </table1>  <table2>  <row>  <field1>value</field1><field2>value</field2><fieldN>value</fieldN>...  </row>  <row>  <field1>value</field1><field2>value</field2><fieldN>value</fieldN>...  </row>  ...  </table2>  ...  </root> |

**例子**

|  |
| --- |
| document.getRootElement();//获取document根元素 |

**方法列表**

| **方法名称** | **方法简介** |
| --- | --- |
| Element getRootElement() | 获取Document对象的根元素 |
| Element createElement(String name) | 通过指定名称创建Element对象 |
| Element createElement(String name, String value) | 通过指定名称创建Element并且设置Element的值 |
| TransformContext getContext() | 获取当前DEE任务的上下文 |

**方法签名：**

Element getRootElement()

**简要描述：**

* 获取Document对象的根元素

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| 无 |  |  |  |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| com.seeyon.v3x.dee.Document.Element | Element对象 |

**使用例子**

|  |
| --- |
| Element root = document.getRootElement();//获取document根元素 |

**备注**

* 无

**方法签名：**

Element createElement(String name)

**简要描述：**

* 通过指定名称创建Element对象

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| name | String | Element名称 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| com.seeyon.v3x.dee.Document.Element | Element对象 |

**使用例子**

|  |
| --- |
| Element table1 = document.createElement("table1");//创建名称为“table1”的元素 |

**备注**

* 无

**方法签名：**

Element createElement(String name, String value)

**简要描述：**

* 通过指定名称创建Element并且设置Element的值

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| name | String | Element名称 |
| value | String | Element值 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| com.seeyon.v3x.dee.Document.Element | Element对象 |

**使用例子**

|  |
| --- |
| Element table1 = document.createElement("table1","hello world");//创建名称为“table1”的元素，并且将元素的值设置为“hello world” |

**备注**

* 无

**方法签名：**

TransformContext getContext()

**简要描述：**

* 获取当前DEE任务的上下文

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| 无 |  |  |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| com.seeyon.v3x.dee.TransformContext | 上下文 |

**使用例子**

|  |
| --- |
| document.getContext()；//获取当前DEE任务的上下文 |

**备注**

* 通过上述例子，就可以获取到当前任务的上下文，从而获取/设置上下文中的数据

### 1.1.2、Document.Element

**简要描述：**

* com.seeyon.v3x.dee.Document.Element
* Element是Document的元素
* 一级元素为Document根元素，只有唯一一个，名称为“root”
* 二级元素为“root”的子元素，定义为表名称，名称自定义但不允许重复，可以有多个
* 三级元素为表元素的子元素，定义为表中行记录，名称固定为“row”，可以有多个
* 四级元素为“row”的子元素，定义为字段，名称自定义但不允许重复，可以有多个，字段的值为数据值

**格式如下：**

|  |
| --- |
| <root>  <table1>  <row>  <field1>value</field1><field2>value</field2><fieldN>value</fieldN>...  </row>  <row>  <field1>value</field1><field2>value</field2><fieldN>value</fieldN>...  </row>  ...  </table1>  <table2>  <row>  <field1>value</field1><field2>value</field2><fieldN>value</fieldN>...  </row>  <row>  <field1>value</field1><field2>value</field2><fieldN>value</fieldN>...  </row>  ...  </table2>  ...  </root> |

**方法列表：**

| **方法名称** | **方法简介** |
| --- | --- |
| String getName() | 获取当前元素的名称 |
| Object getValue() | 获取当前元素的值 |
| void setValue(Object value) | 设置当前元素的值 |
| List getChildren() | 获取当前元素的所有直接子元素 |
| List getChildren(String name) | 根据指定名称获取元素的直接子元素 |
| List getAttributes() | 获取当前元素的所有属性 |
| Attribute getAttribute(String name) | 根据指定名称获取当前元素的属性 |
| Attribute setAttribute(String name, Object value) | 设置当前元素的属性 |
| Element getChild(String name) | 根据名称获取当前元素的子元素 |
| Element addChild(String name) | 为当前元素增加指定名称的子元素 |
| Element addChild(Element element) | 为当前元素增加子元素 |
| void removeChild(String name) | 根据指定元素名称删除当前元素的子元素 |

**方法签名：**

String getName()

**简要描述：**

* 获取当前元素的名称

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| 无 |  |  |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| String | 元素名称 |

**使用例子**

|  |
| --- |
| document.getRootElement().getName();//可以获取root元素的名称 |

**备注**

* 无

**方法签名：**

Object getValue();

**简要描述：**

* 获取当前元素的值

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| 无 |  |  |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| Object | 元素的值 |

**使用例子**

|  |
| --- |
| List<Element> list = document.getRootElement().getChild("table1").getChildren();//先获取table1的所有row元素  list.get(0).getChild("field1").getValue();//获取第一个row元素的子元素field1元素的值 |

**备注**

* 无

**方法签名：**

void setValue(Object value)

**简要描述：**

* 设置当前元素的值

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| value | Object | 元素的值 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| void | 没有返回值 |

**使用例子**

|  |
| --- |
| List<Element> list = document.getRootElement().getChild("table1").getChildren();//先获取table1的所有row元素  list.get(0).getChild("field1").setValue("hello world");//获取第一个row元素的子元素field1，并且设置field1元素的值为"hello world" |

**备注**

* 无

**方法签名：**

List<Element> getChildren()

**简要描述：**

* 获取当前元素的所有直接子元素

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| 无 |  |  |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| List | 所有直接子节点Element对象的集合 |

**使用例子**

|  |
| --- |
| document.getRootElement().getChildren();//可获取root节点下所有直接子节点（即表节点） |

**备注**

* 无

**方法签名：**

List<Element> getChildren(String name)

**简要描述：**

* 根据指定名称获取元素的直接子元素

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| name | String | 子节点名称 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| List | 所有名称为name的直接子节点Element对象的集合 |

**使用例子**

|  |
| --- |
| document.getRootElement().getChild("table1").getChildren("row");//可获取表table1节点下的所有row节点的集合 |

**备注**

* 无

**方法签名：**

List<Attribute> getAttributes()

**简要描述：**

* 获取当前元素的所有属性

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| 无 |  |  |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| List | 元素所有属性的集合 |

**使用例子**

|  |
| --- |
| document.getRootElement().getChild("table1").getAttributes();//获取表节点table1的所有属性 |

**备注**

* 无

**方法签名：**

Attribute getAttribute(String name)

**简要描述：**

* 根据指定名称获取当前元素的属性

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| name | String | 属性名称 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| com.seeyon.v3x.dee.Document.Attribute | 元素属性对象 |

**使用例子**

|  |
| --- |
| document.getRootElement().getChild("table1").getAttribute("totalCount");//获取表节点table1的“totalCount”属性 |

**备注**

* 无

**方法签名：**

Attribute setAttribute(String name, Object value)

**简要描述：**

* 设置当前元素的属性

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| name | String | 属性名称 |
| value | Object | 属性值 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| com.seeyon.v3x.dee.Document.Attribute | 设置到元素的属性对象 |

**使用例子**

|  |
| --- |
| document.getRootElement().getChild("table1").setAttribute("id","888");//设置元素table1的属性id的值为“888”，如果元素table1没有属性“id”，那么会新建属性名称为“id”，值为“888”，并且设置给元素 |

**备注**

* 无

**方法签名：**

Element getChild(String name)

**简要描述：**

* 根据名称获取当前元素的子元素

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| name | String | 子元素名称 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| com.seeyon.v3x.dee.Document.Element | 子元素对象 |

**使用例子**

|  |
| --- |
| document.getRootElement().getChild("table1");//获取根元素下的名称为table1的表元素 |

**备注**

* 无

**方法签名：**

Element addChild(String name)

**简要描述：**

* 为当前元素增加指定名称的子元素

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| name | String | 子元素名称 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| com.seeyon.v3x.dee.Document.Element | 子元素对象 |

**使用例子**

|  |
| --- |
| document.getRootElement().addChild("table2");//增加根元素的子元素，名称为“table2” |

**备注**

* 无

**方法签名：**

Element addChild(Element element)

**简要描述：**

* 为当前元素增加子元素

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| element | com.seeyon.v3x.dee.Document.Element | 子元素Element对象 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| com.seeyon.v3x.dee.Document.Element | 子元素对象 |

**使用例子**

|  |
| --- |
| Element table3 = document.createElement("table3");//创建名称为“table3”的元素  document.getRootElement().addChild(table3);//将“table3”元素插入根元素下，作为根元素的子元素 |

**备注**

* 无

**方法签名：**

void removeChild(String name)

**简要描述：**

* 根据指定元素名称删除当前元素的子元素

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| name | String | 子元素名称 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| void | 没有返回值 |

**使用例子**

|  |
| --- |
| document.getRootElement().removeChild("table1");//删除根元素下名称为“table1”的子元素 |

**备注**

* 无

### 1.1.3、Document.Attribute

**简要描述：**

* com.seeyon.v3x.dee.Document.Attribute
* Attribute是Element元素上的属性

|  |
| --- |
| <table totalCount="1000"> //totalCount就是Attribute  ...  </table> |

* 如何获取Attribute

|  |
| --- |
| document.getRootElement().getChild("table").getAttribute("totalCount");//可以获取table元素的名称为“totalCount”的Attribute对象 |

**方法列表**

| **方法名称** | **方法简介** |
| --- | --- |
| String getName() | 获取Attribute对象的名称 |
| Object getValue() | 获取当前Attribute对象的值 |
| void setValue(Object value) | 给当前Attribute对象设置值 |

**方法签名：**

String getName()

**简要描述：**

* 获取Attribute对象的名称

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| 无 |  |  |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| String | Attribute对象的名称 |

**使用例子**

|  |
| --- |
| List<Attribute> list = document.getRootElement().getChild("table").getAttributes();  for (Attribute attribute:list){  attribute.getName();  }//可以获取table元素的所有属性的名称 |

**备注**

* 无

**方法签名：**

Object getValue()

**简要描述：**

* 获取当前Attribute对象的值

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| 无 |  |  |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| Object | Attribute对象的值，类型为Object |

**使用例子**

|  |
| --- |
| document.getRootElement().getChild("table").getAttribute("totalCount").getValue();//可以获取table元素的名称为“totalCount”的Attribute对象的值 |

**备注**

* 无

**方法签名：**

void setValue(Object value)

**简要描述：**

* 给当前Attribute对象设置值

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| value | Object | Attribute对象的值 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| void | 没有返回值 |

**使用例子**

|  |
| --- |
| document.getRootElement().getChild("table").getAttribute("totalCount").setValue("500");//将table元素的名称为“totalCount”的属性的值设置为“500” |

**备注**

* 无

### 1.1.4、TransformContext

**简要描述：**

* com.seeyon.v3x.dee.TransformContext
* 转换上下文
* 用于整个DEE任务生命周期中传递数据

如何获取上下文，例子：

|  |
| --- |
| document.getContext();//可以获取当前DEE任务的转换上下文 |

**方法列表**

| **方法名称** | **方法简介** |
| --- | --- |
| void setAttribute(String name, Object object) | 向当前DEE任务的转换上下文中添加属性 |
| Object getAttribute(String name) | 根据名称获取当前DEE任务转换上下文中的属性值 |
| Parameters getParameters() | 获取当前DEE任务的参数容器 |

**方法签名：**

void setAttribute(String name, Object object)

**简要描述：**

* 向当前DEE任务的转换上下文中添加属性

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| name | String | 属性名称 |
| object | Object | 属性值 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| void | 没有返回值 |

**使用例子**

|  |
| --- |
| document.getContext().setAttribute("MyData","hello world");//向当前任务的转换上下文中添加了一个名称为“MyData”的属性，值为“hello world” |

**备注**

* 无

**方法签名：**

Object getAttribute(String name)

**简要描述：**

* 根据名称获取当前DEE任务转换上下文中的属性值

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| name | String | 属性名称 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| Object | 属性值 |

**使用例子**

|  |
| --- |
| document.getContext().getAttribute("MyData");//获取当前DEE任务的转换上下文中名称为“MyData”的属性的值 |

**备注**

* 无

**方法签名：**

Parameters getParameters()

**简要描述：**

* 获取当前DEE任务的参数容器

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| 无 |  |  |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| com.seeyon.v3x.dee.Parameters | 参数容器，当前DEE任务的所有参数都在该容器中 |

**使用例子**

|  |
| --- |
| document.getContext().getParameters();//获取当前DEE任务的参数容器 |

**备注**

* 无

### 1.1.5、Parameters

**简要描述：**

* com.seeyon.v3x.dee.Parameters
* Parameters是当前DEE任务的参数容器

**如何获取Parameters**

|  |
| --- |
| document.getContext().getParameters();//可以获取当前DEE任务的参数容器 |

**方法列表**

| **方法名称** | **方法简介** |
| --- | --- |
| public Parameters add(String name, Object value) | 向参数容器中添加参数 |
| public Object getValue(String name) | 获取指定名称的参数值 |

**方法签名：**

public Parameters add(String name, Object value)

**简要描述：**

* 向参数容器中添加参数

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| name | String | 参数名称 |
| value | Object | 参数值 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| com.seeyon.v3x.dee.Parameters | 参数容器 |

**使用例子**

|  |
| --- |
| document.getContext().getParameters().add("myData","hello world");//向当前DEE任务的参数容器中添加名称为“myData”的参数，值为“hello world” |

**备注**

* 无

**方法签名：**

public Object getValue(String name)

**简要描述：**

* 获取指定名称的参数值

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| name | String | 参数名称 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| Object | 参数值 |

**使用例子**

|  |
| --- |
| document.getContext().getParameters().add("myData","hello world");  document.getContext().getParameters().getValue("myData");//获取名称为myData的参数值 |

**备注**

* 无

# 2、系统代码库

## 2.1、document函数

### 2.1.1、[根据字段值获取包含该值的document]

**方法签名：**

getDocByVal(document,name,val)

**简要描述：**

* 根据某个字段的值获取包含该值的Document

**参数：**

| **参数名** | **必选** | **类型** | **说明** |
| --- | --- | --- | --- |
| document | 是 | object | Document对象 |
| name | 是 | string | 字段名 |
| val | 是 | string | 字段值 |

**参数传入示例**

|  |
| --- |
| ###### document:  <root>  <b count=2 totalCount=2>  <row>  <b2>a</b2>  <b1>1</b1>  </row>  <row>  <b2>b</b2>  <b1>1</b1>  </row>  </b>  </root>  ###### name: b2  ###### val:a |

**函数调用示例**

|  |
| --- |
| Document doc = getDocByVal(document,"b2","a");  return doc; //该操作将替换原有docment内容 |

**函数返回说明**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| document | object | Document对象 |

**返回示例**

|  |
| --- |
| <root>  <b>  <row>  <b2>a</b2>  <b1>1</b1>  </row>  </b>  </root> |

**备注**

### 1.1.2、[根据字段拆分document]

**方法签名：**

splitDocByName(document, name)

**简要描述：**

* 根据某个字段的拆分document

**参数：**

| **参数名** | **必选** | **类型** | **说明** |
| --- | --- | --- | --- |
| document | 是 | object | document对象 |
| name | 是 | string | 字段名 |

**参数传入示例**

|  |
| --- |
| ###### document:  <root>  <b count=2 totalCount=2>  <row>  <b2>a</b2>  <b1>1</b1>  </row>  <row>  <b2>b</b2>  <b1>1</b1>  </row>  </b>  </root>  ###### name: b2 |

**函数调用示例**

|  |
| --- |
| import java.util.List;  List<Document> docList = splitDocByName(document,"b2"); |

**函数返回说明**

| **类型** | **说明** |
| --- | --- |
| List | 返回拆分为多个document的list对象 |

**返回示例**

|  |
| --- |
| docList.get(0):  <root>  <b>  <row>  <b2>a</b2>  <b1>1</b1>  </row>  </b>  </root>  docList.get(1):  <root>  <b>  <row>  <b2>b</b2>  <b1>1</b1>  </row>  </b>  </root> |

**备注**

### 1.1.3、[获取webservice适配器返回值]

**方法签名：**

getWsResult(document)

**简要描述：**

* 使用了Webservice适配器以后，如果返回值不满足格式不能被合并到document中，那么可以使用该方法获取Webservice适配器的返回值

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| document | com.seeyon.v3x.dee.Document | 当前document对象，脚本中内置了当前任务的document对象，一般不需要变动，可以直接使用 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| Object | 返回Webservice适配器返回值，一般为String数据类型 |

**使用例子**

|  |
| --- |
| //事先的Webservice适配器调用后返回了字符串“hello world”  String wsResult = getWsResult(document);  println wsResult; |

**输出结果**

|  |
| --- |
| hello world |

**备注**

* 无

### 1.1.4、[创建以root为根节点的document]

**方法签名：**

buildDoc()

**简要描述：**

* 快速生成以root为根节点的document

**参数：**

| **参数名** | **必选** | **类型** | **说明** |
| --- | --- | --- | --- |

**参数传入示例**

|  |
| --- |
|  |

**函数调用示例**

|  |
| --- |
| Document doc = buildDoc();  return doc; //此操作将把当前document对象替换为doc |

**函数返回说明**

| **类型** | **说明** |
| --- | --- |
| object | 空document对象 |

**返回示例**

|  |
| --- |
| <root dee\_isNull=true>  </root> |

**备注**

### 1.1.5、[获取document根节点]

**方法签名：**

getRootElement(document)

**简要描述：**

* 获取document根节点

**参数：**

| **参数名** | **必选** | **类型** | **说明** |
| --- | --- | --- | --- |
| document | 是 | object | document对象 |

**参数传入示例**

|  |
| --- |
| ###### document:  <root>  <b count=2 totalCount=2>  <row>  <b2>a</b2>  <b1>1</b1>  </row>  <row>  <b2>b</b2>  <b1>1</b1>  </row>  </b>  </root> |

**函数调用示例**

|  |
| --- |
| Element root = getRootElement(document);  if(root != null){  for(Element e:root.getChildren()){  if(e == null) continue;  println "表名:"+e.getName(); //循环打印根节点下的所有表名  }  } |

**函数返回说明**

| **类型** | **说明** |
| --- | --- |
| Element | 根节点对象 |

**返回示例**

|  |
| --- |
|  |

**备注**

### 1.1.6、[创建Element节点]

**方法签名：**

buildElement(document, name)

**简要描述：**

* 创建Element节点

**参数：**

| **参数名** | **必选** | **类型** | **说明** |
| --- | --- | --- | --- |
| document | 是 | object | document对象 |
| name | 是 | string | 节点名 |

**参数传入示例**

|  |
| --- |
| ###### document:  <root>  <b count=2 totalCount=2>  <row>  <b2>a</b2>  <b1>1</b1>  </row>  <row>  <b2>b</b2>  <b1>1</b1>  </row>  </b>  </root>  ###### name: table1 |

**函数调用示例**

|  |
| --- |
| Element e = buildElement(document, "table1");  if(e != null){  println "节点信息:"+e.toString(); //打印返回节点信息  } |

**函数返回说明**

| **类型** | **说明** |
| --- | --- |
| Element | 节点名为table1的节点对象 |

**返回示例**

|  |
| --- |
| <table1 dee\_isNull="true"></table1> |

**备注**

### 1.1.7、[获取Element所有子节点]

**方法签名：**

getAllChildOfElement(element)

**简要描述：**

* 获取element的所有子节点

**参数：**

| **参数名** | **必选** | **类型** | **说明** |
| --- | --- | --- | --- |
| element | 是 | object | Element对象 |

**参数传入示例**

|  |
| --- |
| ###### element:  <root>  <b count=2 totalCount=2>  <row>  <b2>a</b2>  <b1>1</b1>  </row>  <row>  <b2>b</b2>  <b1>1</b1>  </row>  </b>  </root> |

**函数调用示例**

|  |
| --- |
| Element root = getRootElement(document); //获取document的根节点  if(root != null){  List<Element> chds = getAllChildOfElement(root); //获取root节点的子节点  for(Element e:chds){  if(e == null) continue;  println "节点信息"+e.toString();  }  } |

**函数返回说明**

| **类型** | **说明** |
| --- | --- |
| List | 返回节点的子节点集 |

**返回示例**

|  |
| --- |
|  |

**备注**

### 1.1.8、[根据名称获取Element子节点]

**方法签名：**

getChildByName(element, name)

**简要描述：**

* 根据名称获取element的子节点

**参数：**

| **参数名** | **必选** | **类型** | **说明** |
| --- | --- | --- | --- |
| element | 是 | object | Elementt对象 |
| name | 是 | string | 节点名 |

**参数传入示例**

|  |
| --- |
| ###### element:  <root>  <t1 count=2 totalCount=2>  <row>  <b2>a</b2>  <b1>1</b1>  </row>  <row>  <b2>b</b2>  <b1>1</b1>  </row>  </t1>  <t2 count=1 totalCount=1>  <row>  <b2>c</b2>  <b1>2</b1>  </row>  </t2>  </root>  ###### name: t2 |

**函数调用示例**

|  |
| --- |
| Element root = getRootElement(document);  if(root != null){  Element e = getChildByName(root, "t2"); //获取根节点下节点名为t2的子节点  if(e != null){  println "节点"+e.toString();  }  } |

**函数返回说明**

| **类型** | **说明** |
| --- | --- |
| Element | 返回对应子节点对象 |

**返回示例**

|  |
| --- |
| ###### 示例中e:  <t2 count=1 totalCount=1>  <row>  <b2>c</b2>  <b1>2</b1>  </row>  </t2> |

**备注**

### 1.1.9、[设置Element的值]

**方法签名：**

setVal2Element(element, value)

**简要描述：**

* 设置Element对象的value

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| element | com.seeyon.v3x.dee.Document.Element | Element对象 |
| value | Object | Element的value |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| void | 没有返回值 |

**使用例子**

|  |
| --- |
| Element id = document.getRootElement().getChild("table").getChild("row").getChild("id");//获取名称为id的Element节点  setVal2Element(id, "yangyu");//设置名称为id的Element节点的value为“yangyu” |

**使用方法之前输出结果**

|  |
| --- |
| <root>  <table count=1 totalCount=1>  <row>  <name>a</name>  <id>1001</id>  </row>  </table>  </root> |

**输出结果**

|  |
| --- |
| <root>  <table count=1 totalCount=1>  <row>  <name>a</name>  <id>yangyu</id>  </row>  </table>  </root> |

**备注**

* 无

### 1.1.10、[获取Element的值]

**方法签名：**

getElementValue(element)

**简要描述：**

* 获取Element对象的value值

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| element | com.seeyon.v3x.dee.Document.Element | Element对象 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| Object | Element对象的value |

**使用例子**

|  |
| --- |
| Element id = document.getRootElement().getChild("table").getChild("row").getChild("id");//获取名称为id的Element节点  println getElementValue(id);//打印名称为id的Element节点的value |

**输出结果**

|  |
| --- |
| 1001 |

**备注**

* 无

### 1.1.11、[根据xml生成document]

**方法签名：**

createDocByXml(xml)

**简要描述：**

* 根据xml字符串生成document对象

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| xml | String | XML结构的字符串 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| com.seeyon.v3x.dee.Document | Document对象 |

**使用例子**

|  |
| --- |
| String xml = "<root><table><row><name>yangyu</name></row></table></root>";//xml格式的字符串  createDocByXml(xml);//生成document |

**输出结果**

|  |
| --- |
| <?xml version=1.0 encoding=UTF-8?>  <root>  <table>  <row>  <name>yangyu</name>  </row>  </table>  </root> |

**备注**

* 无

### 1.1.12、[合并document]

**方法签名：**

mergeDoc(document1,document2)

**简要描述：**

* 将两个不同的document合并为一个document

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| document1 | com.seeyon.v3x.dee.Document | 第一个document |
| document2 | com.seeyon.v3x.dee.Document | 第二个document |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| com.seeyon.v3x.dee.Document | 合并后的Document对象 |

**使用例子**

|  |
| --- |
| String xml = "<root><table><row><name>yangyu</name></row></table></root>";  Document doc1 = createDocByXml(xml);//创建第一个document  String xml1 = "<root><table2><row><sex>男</sex></row></table2></root>";  Document doc2 = createDocByXml(xml1);//创建第二个document  mergeDoc(doc1,doc2);//将两个document合并 |

**输出结果**

|  |
| --- |
| <root>  <table count=1 totalCount=1>  <row>  <name>yangyu</name>  </row>  </table>  <table2 count=1 totalCount=1>  <row>  <sex>男</sex>  </row>  </table2>  </root> |

**备注**

* 无

### 1.1.13、[根据名称获取参数值]

**方法签名：**

getParaByName(document,name)

**简要描述：**

* 根据指定名称获取参数值

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| document | com.seeyon.v3x.dee.Document | 当前DEE任务的document，一般情况下不需要变动 |
| name | String | 需要取值的参数名称 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| Object | 指定名称参数的value |

**使用例子**

|  |
| --- |
| setParam(document, "myName", "yangyu");//设置参数  println getParaByName(document,"myName");//获取参数值 |

**输出结果**

|  |
| --- |
| yangyu |

**备注**

* 无

### 1.1.14、[添加参数]

**方法签名：**

setParam(document, name, value)

**简要描述：**

* 向当前DEE任务的参数容器中添加参数，添加了参数以后，后续的任何适配器都可以使用该参数

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| document | com.seeyon.v3x.dee.Document | 当前DEE任务的document，一般情况下不需要变动 |
| name | String | 参数名称 |
| value | Object | 参数value |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| void | 没有返回值 |

**使用例子**

|  |
| --- |
| setParam(document, "myName", "yangyu");//添加了名称为“myName”的参数  println getParaByName(document,"myName");//获取参数值 |

**输出结果**

|  |
| --- |
| yangyu |

**备注**

* 无

### 1.1.15、[根据字段值获取row集合]

**方法签名：**

getRowListByVal(document, name, val)

**简要描述：**

* 根据指定的字段名称、字段值，获取document包含该字段并且值相同的数据行集合

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| document | com.seeyon.v3x.dee.Document | 当前DEE任务的document，一般情况下不需要变动 |
| name | String | 字段名称 |
| value | Object | 字段值 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| List | 数据行集合，也就是名称为“row”的Element对象集合 |

**使用例子**

|  |
| --- |
| List<Element> rows = getRowListByVal(document, "name", "yangyu");//获取所有包含“name”字段，并且值为“yangyu”的数据行集合  for(Element row : rows){//循环并打印集合内容  println getChildByName(row, "id").getValue();  } |

**document数据内容**

|  |
| --- |
| <root>  <table>  <row>  <name>yangyu</name>  <id>111</id>  </row>  <row>  <name>yangyu</name>  <id>222</id>  </row>  <row>  <name>yangyu</name>  <id>333</id>  </row>  </table>  </root> |

**输出结果**

|  |
| --- |
| 111  222  333 |

**备注**

* 无

### 1.1.16、[根据xml文件创建document]

**方法签名：**

createDocByFile(path)

**简要描述：**

* 根据文件创建document，但要求文件内容必须是XML格式的数据

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| path | String | 文件绝对路径 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| com.seeyon.v3x.dee.Document | 返回Document对象 |

**使用例子**

|  |
| --- |
| createDocByFile("/Users/yangyu/Downloads/dee.xml");//读取dee.xml中的数据，并且生成document |

**文件内容**

|  |
| --- |
| <root><table><row><name>yangyu</name><sex>nan</sex></row></table></root> |

**输出结果**

|  |
| --- |
| <?xml version=1.0 encoding=UTF-8?>  <root>  <table>  <row>  <name>yangyu</name>  <sex>nan</sex>  </row>  </table>  </root> |

**备注**

* 无

### 1.1.17、[获取FormData数据]

**方法签名：**

formDataToDoc()

**简要描述：**

* 直接将A8开发高级中传过来的FormData参数转换成为Document对象
* 如果该任务用于A8开发高级调用，A8开发高级会将表单数据传入DEE任务的参数容器中，名称为“FormData”
* 通过该方法，可以直接将参数容器中的“FormData”数据转换为Document对象，供后续适配器使用

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| 无 |  |  |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| com.seeyon.v3x.dee.Document | 返回Document对象 |

**使用例子**

|  |
| --- |
| formDataToDoc();//将FormData参数转换成为Document对象 |

**输出结果**

|  |
| --- |
| <?xml version=1.0 encoding=UTF-8?>  <root formName=表单名>  <formmain\_0012>  <row>  <id>4937605540874345548</id>  <state>1</state>  <start\_member\_id>901373261631474415</start\_member\_id>  <start\_date>2016-12-13 13:36:06.5</start\_date>  <approve\_member\_id>0</approve\_member\_id>  <approve\_date>2016-12-13 13:36:10.642</approve\_date>  <finishedflag>0</finishedflag>  <ratifyflag>0</ratifyflag>  <ratify\_member\_id>0</ratify\_member\_id>  <ratify\_date/>  <sort>0</sort>  <modify\_member\_id>901373261631474415</modify\_member\_id>  <modify\_date>2016-12-13 13:36:06.0</modify\_date>  <field0001 display=编号>123</field0001>  <field0002 display=填写日期>123</field0002>  <field0003 display=姓名>123</field0003>  <field0004 display=部门>123</field0004>  <field0005 display=岗位职务>123</field0005>  <field0006 display=印章名称-公章>123</field0006>  <field0007 display=印章名称-合同专用章>123</field0007>  <field0008 display=印章名称-法人代表名章>123</field0008>  <field0009 display=印章名称-财务专用章>123</field0009>  <field0010 display=印章名称-发票专用章>123</field0010>  <field0011 display=用印文件名称>123</field0011>  <field0012 display=用印文件数量>123</field0012>  <field0013 display=用印文件电子文档>123</field0013>  <field0014 display=用印文件重要程度>123</field0014>  <field0015 display=用印文件用途>123</field0015>  <field0016 display=部门负责人意见>123</field0016>  <field0017 display=行政部门意见>123</field0017>  <field0018 display=部门分管领导意见>123</field0018>  <field0019 display=总经理意见>123</field0019>  <field0020 display=印章使用记录-用印日期>123</field0020>  <field0021 display=印章使用记录-文件份数>123</field0021>  <field0022 display=印章使用记录-用印数量>123</field0022>  <field0023 display=印章使用记录-操作人>123</field0023>  <field0024 display=印章使用记录-说明>123</field0024>  </row>  </formmain\_0012>  </root> |

**备注**

* 无

### 1.1.18、[获取REST适配器返回值]

**方法签名：**

getRestResult(document)

**简要描述：**

* 获取REST适配器返回值
* REST会尝试将返回值转换成为Document对象，如果返回值不符合Document对象格式，那么就需要使用该方法来获取返回值

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| document | com.seeyon.v3x.dee.Document | 当前DEE任务的Document对象，脚本中已经内置该参数，一般不需要改变 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| Object | REST适配器返回值 |

**使用例子**

|  |
| --- |
| getRestResult(document);//直接获取REST适配器返回值 |

**输出结果**

|  |
| --- |
|  |

**备注**

* 无

## 1.2、数据源函数

### 2.2.1、[执行存储过程]

**方法签名：**

exeProcedure(sourceName, procedureName, parms)

**简要描述：**

* 使用该方法可以调用数据库存储过程

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| sourceName | String | 数据源名称 |
| procedureName | String | 存储过程名称，如果是oracle数据库需要加包名 |
| parms | String | 参数值，多个参数值用","号隔开 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| 无 | 会将执行存储过程获取的数据合并到当前任务的document中 |

**使用例子**

|  |
| --- |
| exeProcedure("local", "getid", "a");//调用名称为“getid”的存储过程，传入了一个参数值“a” |

**输出结果**

|  |
| --- |
| <root>  <from count=1 totalCount=1>  <row>  <id>1001</id>  </row>  </from>  </root> |

**备注**

* 无

### 2.2.2、[根据名称获取数据源]

**方法签名：**

getDsByName(name)

**简要描述：**

* 根据名称获取数据源

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| name | String | 数据源名称 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| com.seeyon.v3x.dee.resource.DbDataSource | 数据源对象 |

**使用例子**

|  |
| --- |
| import java.sql.ResultSet;  import java.sql.Statement;  import java.sql.Connection;  import java.sql.SQLException;  import com.seeyon.v3x.dee.resource.DbDataSource;  DbDataSource ds = getDsByName("local"); //获取数据源  Connection connection = ds.getConnection();//获取连接  Statement stmt = null;  ResultSet rs = null;  String sql = "select \* from test";//sql  try {  stmt = connection.createStatement();  rs = stmt.executeQuery(sql);//执行sql，并获取返回值  while(rs.next()){  println( rs.getObject("id"))//打印返回值中所有“id”字段的值  }  } catch (SQLException e) {  throw e;  }finally{//释放资源，必须要做  rs.close();  stmt.close();  connection.close();  } |

**输出结果**

|  |
| --- |
| 1001  1002  1003  1004 |

**备注**

* 无

## 2.3、开发高级函数

### 2.3.1、[设置开发高级字符串]

**方法签名：**

setHighSetStr(document,status,ispop,reason,remark)

**简要描述：**

* 使用A8开发高级时，如果要使用阻塞模式，那么必须要使用此方法设置返回值
* 该方法用于控制A8表单调用DEE任务时是否弹窗阻塞流程或者不阻塞流程

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| document | com.seeyon.v3x.dee.Document | 脚本中自带document对象，所以在没有需要使用自定义document的情况下，该参数不需要变动 |
| status | String | DEE任务执行状态，取值为"T"不阻塞流程,"F"阻塞流程 |
| ispop | String | A8是否需要弹窗，取值为"Y"弹窗，"N"不弹窗；当status取值为"F"时，A8默认必须弹窗 |
| reason | String | A8弹出窗口中需要显示的信息 |
| remark | String | 备注，暂时作为扩展字段，未启用 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| void | 没有返回值 |

**使用例子**

|  |
| --- |
| setHighSetStr(document,"F","Y","需要阻塞A8流程","");//这个列子就代表了需要弹窗并且阻塞A8流程，并且A8弹出窗口上显示的是“需要阻塞A8流程” |

**输出结果**

|  |
| --- |
| <root>  <deeblockrtn count=1 totalCount=1>  <row>  <deestatus>F</deestatus>  <dialogispop>Y</dialogispop>  <reason>需要阻塞A8流程</reason>  <remark></remark>  </row>  </deeblockrtn>  </root>  可以看到输出的document中多了一张名叫deeblockrtn的表，而A8就是通过这张表的数据来判断是否需要阻塞流程 |

**备注**

* 无

## 2.4、调试函数

### 2.4.1、[调试]

**方法签名：**

debug(lineNum,name,val)

**简要描述：**

* 使用弹出窗口显示所调式变量在任务运行到调试函数时的值
* 该函数只有在DEE可视化配置工具中有效，在A8中运行时自动无效

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| lineNum | String | 调试函数所在脚本函数，系统自动生成，不需要用户填写 |
| name | String | 所需要调试的变量名称 |
| name | Object | 所需要调试的变量 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| void | 没有返回值 |

**使用例子**

|  |
| --- |
| String str = "hello world";  debug("2","str",str);  str = "hello everyone"; |

**输出结果**

|  |
| --- |
| 参数名称：str  脚本名称：script  参数行数：2  参数值：hello world |

**备注**

* 无

### 2.4.2、[中断任务]

**方法签名：**

interruptTask()

**简要描述：**

* 使用该函数以后，当前脚本执行完成后自动终止任务

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| 无 |  |  |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| void | 没有返回值 |

**使用例子**

|  |
| --- |
| if(false){  println "成功";  }  else{  interruptTask();  }  println "ok"; |

**输出结果**

|  |
| --- |
| ok |

**备注**

* 注意：设置该函数后，该脚本的后续代码块会继续执行。该脚本执行完后任务将终止后续的适配器不会执行。

### 2.4.3、[FormData模拟数据]

**方法签名：**

getTestFormData(sourceName, formId)

**简要描述：**

* 在使用开发高级时，数据都是A8通过FormData这个参数传递进DEE里面
* 而在DEE可视化配置工具中配置任务时，无法得到该参数
* 而在后续需要使用到FormData的时候，不知道该格式是什么样子
* 所以提供模拟函数，模拟A8将FormData放入DEE中，方便在DEE可视化配置工具中调试任务
* 该方法只在DEE可视化配置工具中生效，在A8运行环境中不生效
* 所以使用该方法后，在将任务导入到A8当中，A8开发高级会传入带有正确数据的FormData

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| sourceName | String | A8数据源名称 |
| formId | String | 表单id，该id在A8数据库表FORM\_DEFINITION中可查到 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| void | 没有返回值，但会将FormData模拟数据放入当前document参数中 |

**使用例子**

|  |
| --- |
| getTestFormData("a8", "-8848292964178769632");//模拟FormData数据，数据源名称为“a8”，表单id为“-8848292964178769632” |

**输出结果**

|  |
| --- |
| 会看到DEE参数中多了一个FormData参数 |

**备注**

* 无

## 2.5、时间函数

### 2.5.1、[当前时间]

**方法签名：**

date\_now()

**简要描述：**

* 获取当前系统时间

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| 无 |  |  |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| String | 时间字符串，格式为yyyy-MM-dd HH:mm:ss |

**使用例子**

|  |
| --- |
| println(date\_now()); |

**输出结果**

|  |
| --- |
| 2016-12-26 14:38:53 |

**备注**

* 无

## 2.6、字符串操作函数

### 2.6.1、[分割字符串]

**方法签名：**

str\_split(str,separator)

**简要描述：**

* 通过指定字符串分割原字符串

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| str | String | 原字符串 |
| separator | String | 分割字符串 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| String[] | 字符串数组 |

**使用例子**

|  |
| --- |
| String str = "hello#world#everyone";  String[] stringList = str\_split(str,"#");  for(String s : stringList){  println(s);  } |

**输出结果**

|  |
| --- |
| hello  world  everyone |

**备注**

* 无

### 2.6.2、[截取字符串]

**方法签名：**

str\_sub(str,fromIndex,toIndex)

**简要描述：**

* 截取指定长度，任意位置的字符串

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| str | String | 原字符串 |
| fromIndex | int | 开始位置 |
| toIndex | int | 结束位置 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| String | 所截取字符串 |

**使用例子**

|  |
| --- |
| String str = "hello world";  int length = str.length();  println(str\_sub(str,0,0)); //两个值相同，那么返回值固定为“”  println(str\_sub(str,0,length));//截取所有  println(str\_sub(str,0,5));//截取0到5的字符，不包含第5个字符 |

**输出结果**

|  |
| --- |
| //第一个截取为“”  hello world //截取所有  hello //截取0到5 |

**备注**

* 无

### 2.6.3、[字符串是否为空]

**方法签名：**

str\_isEmpty(str)

**简要描述：**

* 判断字符串是否为空

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| str | String | 指定需要判断的字符串 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| boolean | 当参数为null，或者字符串为“”时都会放回true，否则返回false |

**使用例子**

|  |
| --- |
| String str = "";  println(str\_isEmpty(str)); |

**输出结果**

|  |
| --- |
| true |

**备注**

* 无

### 2.6.4、[是否以指定字符串开头]

**方法签名：**

str\_startWith(str,prefix,isIgnoreCase)

**简要描述：**

* 是否以指定字符串开头

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| str | String | 指定需要判断的字符串 |
| prifix | String | 判断的前缀 |
| isIgnoreCase | boolean | 判断时是否忽略大小写 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| boolean | 返回true或者false |

**使用例子**

|  |
| --- |
| String str = "Hello world";  String prefix = "hello";  println(str\_startWith(str,prefix,true)); |

**输出结果**

|  |
| --- |
| true |

**备注**

* 无

## 2.7、数学计算

### 2.7.1、[数字金额大小写转换]

**方法签名：**

math\_digitUppercase(number)

**简要描述：**

* 将数字的金额转换为中文大写数字

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| number | 数字类型 | 数字类型的参数 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| String | 返回中文大写数字 |

**使用例子**

|  |
| --- |
| println(math\_digitUppercase(33333)); |

**输出结果**

|  |
| --- |
| 叁万叁仟叁佰叁拾叁元整 |

**备注**

* 无

### 2.7.2、[加法]

**方法签名：**

math\_add(number1, number2)

**简要描述：**

* 加法算数运算，不丢失精度

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| number1 | 数字类型 | 加数1 |
| number2 | 数字类型 | 加数2 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| 数字类型 | 运算结果 |

**使用例子**

|  |
| --- |
| println math\_add(11.333, 22.444); |

**输出结果**

|  |
| --- |
| 33.777 |

**备注**

* 无

## 2.8、日志管理

### 2.8.1、[日志清理]

**方法签名：**

clearLogs(day)

**简要描述：**

* 用于清理指定天数前的日志

**参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| day | int | 清理多少天前的日志 |

**返回值说明**

| **类型** | **说明** |
| --- | --- |
| void | 没有返回值 |

**使用例子**

|  |
| --- |
| int day = 30;  clearLogs(30); //会清理30天前的日志 |

**备注**

* 无