

Workflow API Reference

All Countries



Typographic Conventions

Type Style	Description
<i>Example</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Textual cross-references to other documents.
Example	Emphasized words or expressions.
EXAMPLE	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
Example	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE	Keys on the keyboard, for example, F2 or ENTER.

Document History

Version	Date	Change
1.0	2014-09-24	First Version

Table of Contents

1	Global Objects and Global Function	5
1.1	Global Object.....	5
1.2	Global Function.....	5
2	Class.....	6
2.1	ExecutionEntity.....	6
2.2	Company.....	7
2.3	Recordset.....	10
2.4	RecordsetParams.....	12
2.5	Field.....	13
2.6	GeneralService.....	14
2.7	GeneralData.....	17
2.8	GeneralDataCollection.....	18
2.9	GeneralDataParams.....	19
2.10	GeneralCollectionParams.....	20
2.11	UserFields.....	21
3	Services	23
3.1	StartService.....	23

1 Global Objects and Global Function

1.1 Global Object

We are introducing the following two global objects:

Object Name	Class	Description
CurrentProcess	ExecutionEntity	Represents the workflow instance which is currently in execution.
company	Company	Represents the company which is currently being connected.

1.2 Global Function

Name:

```
void print(String content)
```

Description:

Output the information to the workflow log file for workflow debugging purposes.

Parameter:

- **Name:** content
- **Type:** String
- **Description:** The string that needs to be output.

Return Value:

```
void
```

2 Class

2.1 ExecutionEntity

Description:

An entity represents the workflow instance that is currently in execution.

Methods:

The following functions are declared in this class:

Method 1: ObjectB1Obj(String dataObjectId)

Description: Get the data object that is currently in execution in the workflow instance. The data object represents an SAP Business One DI object.

Parameter:

Name: dataObjectId

Type: String

Description: TheID of the data object defined in the .xmlworkflow template. The data object represents an SAP Business One DI object.

Return Value:

Type: Object

Description: The specific data object retrieved that represents an SAP Business One DI object.

- **Sample Code:**

```
var order = CurrentProcess.B1Obj('id');  
order.setItemCode('I002');
```

Method 2: ObjectDataObject(String dataObjectId)

Description: Get the customized data object that is currently in execution in the workflow instance.

Parameter:

Name: dataObjectId

Type: String

Description: The ID of the customized data object defined in the .xml workflow template.

- **Return Value:**

Type: Object

Description: The specific customized data object retrieved.

- **Sample Code:**

```
CurrentProcess.DataObject('id').putItem('newItemCode', 'I001');
```

```
var newItemCode = CurrentProcess.DataObject('id').getItem('newItemCode');
```

2.2 Company

Description:

An object represents a single SAP Business One company object.
This object enables you to create business objects in the company.

Methods:

The following functions are declared in this class:

Method 1: Object **getBusinessObject(String objectType, String objectKey)**

Description: Get a data object that represents an SAP Business One DI object.

Parameter:

1. objectType

Name: objectType

Type: String

Description: The type ID of the SAP Business One DI object.

2. objectKey

Name: objectKey

Type: String

Description: The key of the SAP Business One DI object.

- **Return Value:**

Type: Object

Description: The specific data object retrieved that represents an SAP Business One DI object.

- **Sample Code:**

```
var bpService = company.getBusinessService("2");  
bp = company.getBusinessObject("2", "CO01 ");  
bp.getUserFields().put("U_Amity", 600);  
bpService.update(bp);
```

Method 2: **IServicegetBusinessService(String objectType)**

Description: Get a business service for the SAP Business One DI objects.

Parameter:

Name: objectType

Type: String

Description: The type ID of the SAP Business One DI object.

- **Return Value:**

Type: DIService

Description: The specific business service retrieved.

- **Sample Code:**

```
var orderService = company.getBusinessService('17');
var document = company.createDIObject('Document');
document.setCardCode('C001');
document.setDocDueDate('2012-11-26');
var documentLines = doc.getDocumentLines();
if (documentLines == null) {
    documentLines = company.createDIObject('DocumentLines');
}
var documentLine = company.createDIObject('DocumentLine');
documentLine.setItemCode('I001');
documentLine.setQuantity(10);
documentLine.setUnitPrice(100);
documentLines.add(docLine);
document.setDocumentLines(docLines);
orderService.add(doc);
```

Method 3: Object createDIObject(String className)

Description: Create an SAP Business One DI object.

Parameter:

- Name: className
- Type: String
- Description: The class name of the SAP Business One DI object.

- **Return Value:**

Type: Object

Description: The specific SAP Business One DI object created.

- **Sample Code:**

```
var orderService = company.getBusinessService('17');
var document = company.createDIObject('Document');
document.setCardCode('C001');
document.setDocDueDate('2012-11-26');
var documentLines = document.getDocumentLines();
if (documentLines == null) {
    documentLines = company.createDIObject('DocumentLines');
}
var documentLine = company.createDIObject('DocumentLine');
documentLine.setItemCode('I001');
documentLine.setQuantity(10);
```



```
documentLine.setUnitPrice(100);
documentLines.add(docLine);
document.setDocumentLines(docLines);
orderService.add(doc);
```

Method 4: RecordsetgetRecordset()

Description: Get a Recordset object.

- **Return Value:**

Type: Recordset

Description: The DI Recordset object in JavaScript.

- **Sample Code:**

```
var recordsetParams = company.getRecordsetParams();
var query = "select * from OITW where WhsCode = 02 and OnHand < 1000";
recordsetParams.setQuery(query);
var recordset = company.getRecordset();
recordset.doQuery(recordsetParams);
if (0 < recordset.read()) {
    .....
}
```

Method 5: RecordsetParamsgetRecordsetParams()

Description: Get a RecordsetParams object.

- **Return Value:**

Type: RecordsetParams

Description: The DI RecordsetParams object in JavaScript.

- **Sample Code:**

```
var recordsetParams = company.getRecordsetParams();
var query = "select * from OITW where WhsCode = 02 and OnHand < 1000";
recordsetParams.setQuery(query);
var recordset = company.getRecordset();
recordset.doQuery(recordsetParams);
if (0 < recordset.read()) {
    .....
}
```

Method 6: GeneralServicegetGeneralService(String serviceName)

Description: Get a service object.

Parameter:

Name: serviceName

Type: String

Description: The name of the service.

- **Return Value:**

Type: GeneralService

Description: The specific service object retrieved.

- **Sample Code:**

```
var generalService = company.getGeneralService("udoService");
var generalData = generalService.getGeneralData();
var generalDataParam = generalService.getGeneralDataParam();
generalData.put("U_name", "SAP");
generalData.put("U_price", "15");
var childs = generalData.child("childTableName").add();
childs.put("U_name", "SAP Business One");
childs.put("U_amount", "99999");
generalDataParam = generalService.add(generalData);
var newlyAddDocEntry = generalDataParam.get("DocEntry");
```

2.3 Recordset

Description:

A raw data access object that enables you to select data from the database, navigate through the result set, and manipulate user tables.

Methods:

The following functions are declared in this class:

Method 1: void doQuery(RecordsetParams recordsetParams)

Description: Execute a command.

Parameter:

Name: recordsetParams

Type: RecordsetParams

Description: The RecordsetParams object in JavaScript.

- **Return Value:**

Type: void

Description: No return value required.

- **Sample Code:**

```
var recordsetParams = company.getRecordsetParams();
var query = "select * from OITW where WhsCode = 02 and OnHand < 1000";
recordsetParams.setQuery(query);
var recordset = company.getRecordset();
```

```

recordset.doQuery(recordsetParams);
if (0 <recordset.read()){
    .....
}

```

Method 2: booleanread()

Description: Read the next record, and return true if the next record exists.

- **Return Value:**

Type: boolean

Description: True if the next record exists.

- **Sample Code:**

```

var recordsetParams = company.getRecordsetParams();
var query = "select * from OITW where WhsCode = 02 and OnHand< 1000";
recordsetParams.setQuery(query);
var recordset = company.getRecordset();
recordset.doQuery(recordsetParams);
if (0 <recordset.read()){
    .....
}

```

Method 3: Field getField(String fieldName)

Description: Get a field in the current line.

Parameter:

Name: fieldName

Type: String

Description: The name of the field.

- **Return Value:**

Type: Field

Description: The specific field retrieved.

- **Sample Code:**

```

var recordsetParams = company.getRecordsetParams();
var query = "select * from OITW where WhsCode = 02 and OnHand< 1000";
recordsetParams.setQuery(query);
var recordset = company.getRecordset();
recordset.doQuery(recordsetParams);
while (0 <recordset.read()) {
    Field field = recordset.getField("ItemCode");
}

```

Method 4: Field getField(int index)

Description: Get a field in the current line.

Parameter:

Name: index

Type: int

Description: The index of the field.

- **Return Value:**

Type: Field

Description: The specific field retrieved.

- **Sample Code:**

```
var recordsetParams = company.getRecordsetParams();
var query = "select * from OITW where WhsCode = 02 and OnHand< 1000";
recordsetParams.setQuery(query);
var recordset = company.getRecordset();
recordset.doQuery(recordsetParams);
while (0 <recordset.read()) {
    Field field = recordset.getField(1);
}
```

2.4 RecordsetParams

Description:

A parameter used to execute commands.

Method: void setQuery(String query)

Description: Set a query string to the RecordsetParams object.

Parameter:

Name: query

Type: String

Description: A query string.

- **Return Value:**

Type: void

Description: No return value required.

- **Sample Code:**

```
var recordsetParams = company.getRecordsetParams();
var query = "select * from OITW where WhsCode = 02 and OnHand< 1000";
```

```

recordsetParams.setQuery(query);
var recordset = company.getRecordset();
recordset.doQuery(recordsetParams);
if (0 <recordset.read()) {
    .....
}

```

2.5 Field

Description:

An object contains both standard and customized data access properties, which enables you to manipulate the field data.

Methods:

The following functions are declared in this class:

Method 1: String getColumnValue()

Description: Get the value of the current field.

- **Return Value:**

Type: String

Description: The value of the current field.

- **Sample Code:**

```

var recordsetParams = company.getRecordsetParams();
var query = "select * from OITW where WhsCode = 02 and OnHand< 1000";
recordsetParams.setQuery(query);
var recordset = company.getRecordset();
recordset.doQuery(recordsetParams);
while (0 <recordset.read()) {
    Field field = recordset.getField(1);
    var fieldValue= field.getColumnValue();
}

```

Method 2: String getColumnName()

Description: Get the name of the current field.

- **Return Value:**

Type: String

Description: The name of the current field.

- **Sample Code:**

```
var recordsetParams = company.getRecordsetParams();
var query = "select * from OITW where WhsCode = 02 and OnHand < 1000";
recordsetParams.setQuery(query);
var recordset = company.getRecordset();
recordset.doQuery(recordsetParams);
    while (0 < recordset.read()) {
        Field field = recordset.getField(1);
        var fieldName = field.getColumnName();
    }
```

2.6 GeneralService

Description:

The GeneralService provides access to DI objects.

With the service, you can add, retrieve, update, and delete data for DI objects.

Methods:

The following functions are declared in this class:

Method 1: GeneralDataParamsadd(GeneralData data)

Description: Add a record.

Parameter:

Name: data

Type: GeneralData

Description: A GeneralDataobject.

- **Return Value:**

Type: GeneralDataParams

Description: The parameter of the GeneralData.

- **Sample Code:**

```
var generalService = company.getGeneralService("udoService");
var generalData = generalService.getGeneralData();
var generalDataParam = generalService.getGeneralDataParam();
generalData.put("U_name", "SAP");
generalData.put("U_price", "15");
var childs = generalData.child("childTableName").add();
childs.put("U_name", "SAP Business One");
```

```
childs.put("U_amount", "99999");
generalDataParam = generalService.add(generalData);
var newlyAddDocEntry = generalDataParam.get("DocEntry");
```

Method 2: GeneralDataGetGeneralData()

Description: Get a GeneralData object.

- **Return Value:**

Type: GeneralData

Description: The specific GeneralData object retrieved.

- **Sample Code:**

```
var generalService = company.getGeneralService("udoService");
var generalData = generalService.getGeneralData();
var generalDataParam = generalService.getGeneralDataParam();
generalData.put("U_name", "SAP");
generalData.put("U_price", "15");
var childs = generalData.child("childTableName").add();
childs.put("U_name", "SAP Business One");
childs.put("U_amount", "99999");
generalDataParam = generalService.add(generalData);
var newlyAddDocEntry = generalDataParam.get("DocEntry");
```

Method 3: GeneralDataParamsGetGeneralDataParam()

Description: Get a GeneralDataParams object.

- **Return Value:**

Type: GeneralDataParams

Description: The parameter of the GeneralData.

- **Sample Code:**

```
var generalService = company.getGeneralService("udoService");
var generalData = generalService.getGeneralData();
var generalDataParam = generalService.getGeneralDataParam();
generalData.put("U_name", "SAP");
generalData.put("U_price", "15");
var childs = generalData.child("childTableName").add();
childs.put("U_name", "SAP Business One");
childs.put("U_amount", "99999");
generalDataParam = generalService.add(generalData);
var newlyAddDocEntry = generalDataParam.get("DocEntry");
```

Method 4: GeneralDataGetByParams(GeneralDataParamsparam)

Description: Get a GeneralData object by parameters.

Parameter:

Name: param

Type: GeneralDataParams

Description: The parameter of the GeneralData.

- **Return Value:**

Type: GeneralData

Description: The specific GeneralData object retrieved.

- **Sample Code:**

```
var serviceCallSvc = company.getBusinessService("191");
var serviceCallParams = company.createDIObject("ServiceCallParams");
serviceCallParams.setServiceCallID(2);
var serviceCall = serviceCallSvc.getByParams(serviceCallParams);
serviceCall.getUserFields().put("U_ServiceCallTitle", 2);
serviceCallSvc.update(serviceCall);
```

Method 5: GeneralCollectionParamsgetList()

Description: Get a collection of GeneralData objects.

- **Return Value:**

Type: GeneralCollectionParams

Description: The collections of GeneralData objects retrieved.

Method 6: void update(GeneralData generalData)

Description: Update a record.

Parameter:

Name: generalData

Type: GeneralData

Description: A GeneralData object.

- **Return Value:**

Type: void

Description: No return value required.

Method 7: void delete(GeneralDataParams param)

Description: Delete a record.

Parameter:

Name: param

Type: GeneralDataParams

Description: The parameter of the GeneralData object.

- **Return Value:**

Type: void

Description: No return value required.

2.7 GeneralData

Description:

An entity that represents a single record of an object or a child object.

Methods:

The following functions are declared in this class:

Method 1: GeneralDataCollectionchild(String childTableName)

Description: Get a child object in the GeneralData object.

Parameter:

Name: childTableName

Type: String

Description: The name of the child table object.

- **Return Value:**

Type: GeneralDataCollection

Description: A collection of the GeneralData objects retrieved.

Method 2: void clear()

Description: Clear all data in a GeneralData object.

- **Return Value:**

Type: void

Description: No return value required.

Method 3: Object put(String key, Object value)

Description: Put a value in the GeneralData object with a key.

Parameter:

1. key

Name: key

Type: String

2. value

Name: value

Type: Object

- **Return Value:**

Type: Object

Description: The value put in the GeneralData object.

- **Sample Code:**

```
var generalService = company.getGeneralService("udoService");
```

```
var generalData = generalService.getGeneralData();
```

```
var generalDataParam = generalService.getGeneralDataParam();
generalData.put("U_name", "SAP");
generalData.put("U_price", "15");
var childs = generalData.child("childTableName").add();
childs.put("U_name", "SAP Business One");
childs.put("U_amount", "99999");
generalDataParam = generalService.add(generalData);
```

Method 4: Object get(String key)

Description: Get the value in the GeneralData object.

Parameter:

Name: key

Type: String

- **Return Value:**

Type: Object

- **Sample Code:**

```
var generalService = company.getGeneralService("udoService");
var generalData = generalService.getGeneralData();
var generalDataParam = generalService.getGeneralDataParam();
var value = generalData.get("U_name");
```

2.8 GeneralDataCollection

Description:

A collection of GeneralData objects.

Methods:

The following functions are declared in this class:

Method 1: GeneralDataadd()

Description: Add a GeneralData object to the collection.

- **Return Value:**

Type: GeneralData

Description: The GeneralData object added.

- **Sample Code:**

```
var generalService = company.getGeneralService("udoService");
var generalData = generalService.getGeneralData();
```

```
var generalDataParam = generalService.getGeneralDataParam();
generalData.put("U_name", "SAP");
generalData.put("U_price", "15");
var childs = generalData.child("childTableName").add();
childs.put("U_name", "SAP Business One");
childs.put("U_amount", "99999");
generalDataParam = generalService.add(generalData);
```

Method 2: voidclear()

Description: Clear all data in the collection.

- **Return Value:**

Type: void

Description: No return value required.

Method 3: GeneralDataget(int index)

Description: Get a GeneralData object.

Parameter:

Name: index

Type: int

Description: The index of the GeneralData object.

- **Return Value:**

Type: GeneralData

2.9 GeneralDataParams

Description:

An object that holds a collection of key-value pairs.

Method:

The following functions are declared in this class:

Method 1: void clear()

Description: Clear all data in a GeneralDataParams object.

- **Return Value:**

Type: void

Description: No return value required.

Method 2: Object put(String key, Object value)

Description: Put a value in the GeneralDataParams object with a key.

Parameter:

1. objectType
Name: key
Type: String
2. objectKey
Name: value
Type: Object

- **Return Value:**

Type: Object

Description: The value put in the GeneralDataParams.

Method 3: Object get(String key)

Description: Get a GeneralDataParams object.

Parameter:

Name: key
Type: String

- **Return Value:**

Type: Object

2.10 GeneralCollectionParams

Description:

A collection of GeneralDataParams objects.

Methods:

The following functions are declared in this class:

Method 1: void add()

Description: Add a GeneralCollectionParams object.

- **Return Value:**

Type: void

Description: No return value required.

Method 2: void clear()

Description: Clear all data in a GeneralCollectionParams object.

- **Return Value:**

Type: void

Description: No return value required.

Method 3: GeneralDataParamsget(int index)

Description: Get a GeneralDataParams object.

Parameter:

Name: index

Type: int

Description: The index of the GeneralDataParams object.

- **Return Value:**

Type: GeneralDataParams

Description: The parameter of the GeneralData.

2.11 UserFields

Description:

A collection of user fields.

Methods:

The following functions are declared in this class:

Method 1: void clear()

Description: Clear all data in a UserFields object.

- **Return Value:**

Type: void

Description: No return value required.

Method 2: Object put(String key, Object value)

Description: Put a user field in the UserFields object.

Parameter:

1. key

Name: key

Type: String

Description: The key of the user field.

2. value

Name: value

Type: Object

Description: The value of the user field.

- **Return Value:**

Type: Object

Description: The value of the user field.

- **Sample Code:**

```
var svc = company.getBusinessService("1007");
var params = company.createDIObject("ActivityParams");
params.setActivityCode(1);
var svcCall = svc.getActivity(params);
svcCall.getUserFields().put("U_Activity_UDF", 2);
svc.updateActivity(svcCall);
```

Method 3: Object get(Object key)

Description: Get the value of the UserFields object.

Parameter:

Name: key

Type: String

Description: The key of the user field.

- **Return Value:**

Type: Object

Description: The value of the specific user field.

Sample Code:

```
var svc = company.getBusinessService("1007");
var params = company.createDIObject("ActivityParams");
params.setActivityCode(1);
var svcCall = svc.getActivity(params);
var useFieldValue = svcCall.getUserFields().get("U_Activity_UDF");
```

-

3 Services

3.1 StartService

Description:

A service that provides methods in conditional start events only.

Method: Object getStartDataObject(String key)

Description: Retrieve an object from the conditional start event, before an instance starts.

Parameter:

Name: key

Type: String

Description: The ID of the data object defined in the .xaml workflow template.

- **Return Value:**

Type: Object

Description: The specific object retrieved.

- **Sample Code:**

```
var param = company.getRecordsetParams();
var query = "select * from OITW where WhsCode = 02 and OnHand < 1000"
param.setQuery(query);
var recordset = company.getRecordset();
recordset.doQuery(param);
var success = recordset.read();
if (success)
{
    var field = recordset.getField(0);
    var itemC = field.getColumnValue();
    var obj = StartService.getStartDataObject("dataObject4");
    obj.putItem("Key", itemC);
    obj.putItem("ObjType", "4");
}
success > 0;
```

