Generic Object Services (BC-SRV-GBT)

HELP.BCSRVOBS

Release 4.6C

Copyright

Icons

Icon	Meaning
⚠	Caution
💬	Example
➡	Note
🧭	Recommendation
syn	Syntax
💡	Tip

Contents

Generic Object Services (BC-SRV-GBT)

Use

There are general functions that you want to execute with various documents (in general: object). Some of these functions, like for example archiving, require detailed knowledge about the object and can therefore only be realized for a specific object type with a great deal of programming. On the other hand, the **generic object services** only need to know that the object exists to be effective.

Entering a note for a document is a generic object service. The object is the application document.

Prerequisites

In order to be able to take part in the generic object services, a document must belong to an object type in the Business Object Repository (BOR). The application must also publish the document as an application object. For further information see Application Connection to Object Services [Page 25].

Features

For Users:

List of Object Services [Page 6]

For Developers:

Application Connection to Object Services [Page 25]

Create New Object Service [Page 34]

See also:

BC - Relationship Service [Ext.]

Activities

With the exception of the object history, all services are offered as pushbuttons in a separate "toolbox“ window. The symbol appears on the top left of the screen as a pushbutton in an application that uses the object services. The toolbox appears when the button is pressed.

List of Object Services

The following list shows all possible object services. Not all object services are available - it depends on objects that are already active. Individual object services can also only be accessed using a submenu, so that the symbol is not displayed in the toolbox.

Adding a PC Document (to the Storage System) [Page 7]

Adding a PC Document (in the R/3 System) [Page 9]

Entering a Note with an Object [Page 10]

Sending an Object with a Note [Page 11]

Starting a Workflow for an Object [Page 12]

Linking an Internet Address [Page 13]

Entering a Bar Code [Page 14]

Tracing Events for the Object (Subscription) [Page 16]

Telephony [Page 19]

Displaying Object History [Page 20]

Displaying Workflows for the Object [Page 21]

Displaying Transmissions for an Object [Page 22]

Displaying Relationships [Page 23]

Displaying Attachments [Page 24]

Adding a PC Document (to the Storage System)

Use

You use this to store a PC document with SAP ArchiveLink and link it to your application document (your object).

Prerequisites

As well as the general prerequisites (see BC – Generic Object Service [Ext.]), the following Archive Link specific prerequisites also apply:

- A storage system is connected
- You have defined a content repository [Ext.] (IMG activity Maintain Content Repositories [Ext.], transaction OAC0)
- You have maintained document types with the status "active“ (IMG activity Maintain document types [Ext.], transaction OAC2).
- You have defined Relationships [Ext.] for your object type with the status "active" (IMG activity Maintain relationships [Ext.] transaction OAC3)

Procedure

1. In your object display, open the submenu with the symbol 🗋 in the Toolbox [Ext.].
2. Choose *Store business document.*
3. Choose your document type from the selection window by double-clicking on it.

   The document **type** is assigned a specific document **class** (see the graphic below), for example TIF for PC screens or DOC for Word documents.
4. Select your PC document in the window *File name*. Also use the F4 Help. Make sure that you select a document type that you defined in the previous step.

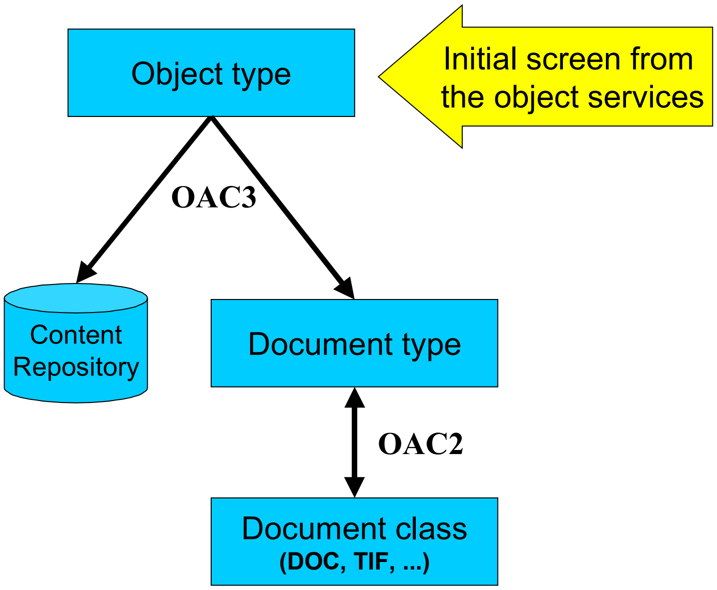
   Your document is stored. To that end, a copy with a technical name is created which is placed in the storage system that you defined in Customizing.

Result

Your PC document is stored. This can be reached using the Attachment List [Page 24] for your object.

Connection of SAP Archive Link objects, maintenance transactions and object services

Adding a PC Document (to the Storage System)



Adding a PC Document

Use

This enables you to link a PC document with your object.

Prerequisites

The general prerequisites [Ext.] apply.

Procedure

5. In your document display (your object), open the submenu with the symbol 🗋 in the Toolbox [Ext.].
6. Choose *Create attachment*.
7. In the next screen select the directory and the file (therefore the PC document) that you want to copy into the SAP System and link with your application document.
8. Choose *open*.

    The document is stored in the Business Workplace and linked to your object.

Result

You have added a document that you can display using the Attachment List [Page 24]. The original PC document is not deleted from the PC.

Entering an Object with a Document

Use

You enter either a note that can be seen by all users or a personal note for your application document (your object).

Prerequisites

The general prerequisites [Ext.] apply.

Procedure

1. In the object display choose in the Toolbox [Ext.]
   - **either** 📝 personal note **or**
   - *Create Note* (in 🗋 - submenu) for the notes that should be generally viewable.
2. You write your note in the next screen, which is stored as a Business Workplace document of type RAW.

   ➡

   The relationship service is called for this service because notes are Business Object Repository (BOR) objects. Generally, an application program can link any BOR objects with one another via an API. For further information see Object relationships [Ext.].

Result

You have created a note that you can display using the List of relationships [Page 23].

Sending an Object with a Note

Use

You use this to send notes to other users within the SAP System which contain attachments with links to your document (your object). When the attachment is displayed, the object is opened for the recipient.

Procedure

9. In your object display, open the submenu with the symbol in the toolbox.
10. Choose *Sending object with a note*.
11. You enter the notes in the next screen. Enter the recipient(s) of the document with the note. To send choose .

Result

The recipients receive the note as a short message in their inbox. The message contains a link to the object as an attachment. By double-clicking you go to this default method. That can, for example, be the display method.

You can likewise display a sent note [Page 22] from the source object.

Starting a Workflow for an Object

Use

This is used for starting workflows that use the object. All suitable workflows automatically determine the system and are offered for execution.

Prerequisites

The SAP System contains active workflows that fulfill the following prerequisites:

- They must be allowed to start the workflow.
- The workflow is only allowed to have the object as an obligatory import parameter.

Tasks that can be executed in the dialog are not included.

Procedure

1. In your object display, open the submenu with the symbol ▶ in the toolbox.
2. Choose *Start workflow*.

   All active workflows which fulfill the prerequisites above are displayed.
3. Select the workflow that you want to start.

For example, the display of a standard order can concern a workflow, within which a further standard order is stored.

Linking an Internet Address

Use

This enables you to link your document (object) with an Internet address.

Procedure

1. In your object display, open the submenu with the symbol 🗋 in the Toolbox [Ext.].
2. Choose *Add external document (URL)*.
3. Enter the Internet address in the next screen.

    Normally it will involve an http protocol hence the address begins with **`http://`**.

Result

You have linked an Internet address to your object which you can select using the Attachment list [Page 24].

Entering a Bar Code

Use

You can assign a bar code to your application document (your object) for late storage with bar code [Ext.]. A scanned document that contains this bar code can therefore be automatically stored via bar code recognition.

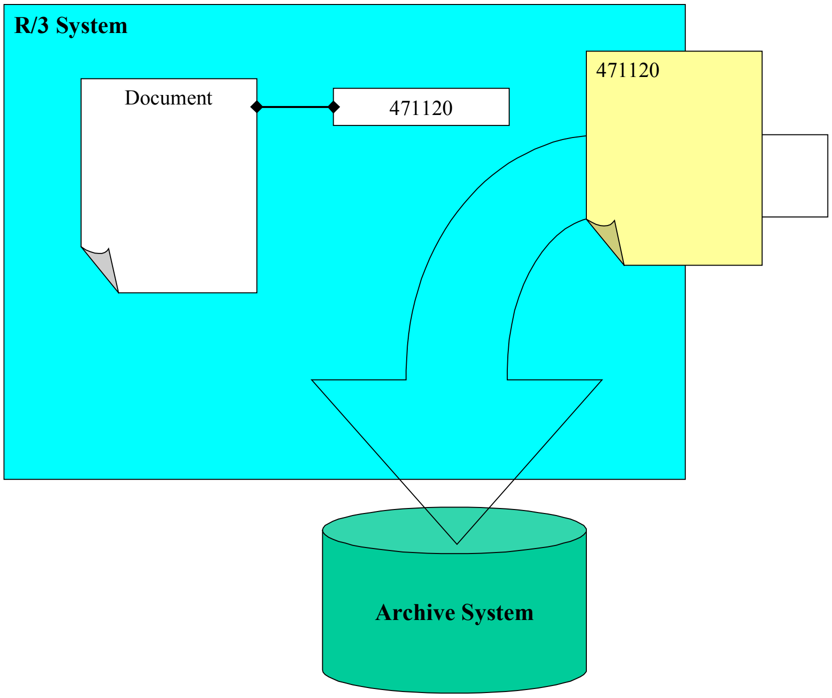
Prerequisites

- A storage system is connected
- Storing with bar code is activated for your object type (IMG activity activate barcode storage [Ext.], transaction OAC5)
- You have defined Relationships [Ext.] for your object type with the status "active" (IMG activity Maintain relationships [Ext.] transaction OAC3)

Procedure

12. In your object display, open the submenu with the symbol 🗋 in the toolbox.
13. Choose *Enter bar code.*
14. Enter a bar code for your document.

Late, automatic content repository with bar code recognition and generic object services



Tracing Events for the Object (Subscription)

Use

If you subscribe to a document (an object) you are informed of relevant events by mail. In the standard system this is changes to objects and deletion of objects. You can store these events or also add further events via the Settings [Page 17].

Prerequisites

The general prerequisites [Ext.] apply.

If necessary, the subscription service must be explicitly activated for your object type and event (see settings [Page 17]).

Procedure

1. In the object display choose 🔔 *Change subscription* in the Toolbox [Ext.].

    You are entered as the subscriber of the object. If you are already a subscriber, you can terminate your subscription in this way.

Result

For example, as soon as changes are made to the object, you receive a corresponding message in the *Business Workplace*. You can define the message text and title yourself using the Settings [Page 17].

Configuring Subscription Service

Use

You determine the events for the object types that the subscription service should be available for and enter the message text.

Without table entries, the service is available for both events CHANGED (document has been changed) and DELETED (document has been deleted) and all object types.

Procedure

1. Go to the view maintenance (transaction SM30) for the table SGOSSUB.
2. Check or change the requested *Object type* and the *Events*. Also see the F1/F4 Help. Descriptions of the object types can be found in the *Business Object Builder* (transaction SWO1).
3. Do not set the indicator *Subscription* if the service is not to be generally available for the event.

   You want to switch off the object service for purchase orders, that is, there should not be a pushbutton offered in the Toolbox [Ext.]. You must also switch it off for the events CHANGED and DELETED as they are automatically available. The simplest way to do this is to enter an asterisk in the field *Event* for the object type **BUS2012** (and naturally remove the indicator).
4. Determine a title for the mails to the subscribers.

   You can transfer SAP*Script* parameters (within two '&' characters):

   - Object attribute. The parameter is called <OBJECTTYPE>.<ATTRIBUTE>
   - The object key (document number). The parameter is called OBJKEY
   - The special default attribute. The parameter is called ATTRIBUT
   - System fields, for example SY-DATE

   You want to subscribe to events for purchase orders and requests and to be able to see from the mail title whether a purchase order or a request has been changed and which purchase order or request this is. Also define the title

   "The purchase order &OBJKEY& has been changed"

   for object type **BUS2012** and a corresponding title for the object type **BUS2032**.
5. You can determine a text for the mails to the subscribers. The text must exist as *general text* in the document maintenance (transaction SE61).

You can transfer the same SAP*Script* parameters as the title. An example is the general text SGBT_DEF_SUB.

You want to see the release group, an attribute for the corresponding object type, in the mail text for the purchase order. Also define the text in the document maintenance.

"The purchase order &OBJKEY& has been changed. It belongs to the release group &BUS2012.ReleaseGroup&.“

and enters these for object type `BUS2012`.

Telephony

Use

From the SAP System outwards you use your telephone system. For example, you can initiate, accept or transfer calls. For the full functional scope see the SAPphone Documentation [Ext.].

Prerequisites

SAPphone [Ext.] is installed in your system.

Procedure

Choose ☎ *Telephony* in the toolbox.

In the next screen enter the number of your partner and start the call.

Displaying Object History

Use

The object history is a log of the documents (objects) that you have recently viewed or edited. It contains the access time and a link to the objects. For example, you can therefore go directly to the object that you edited yesterday

Procedure

1. Choose *System* → *Object history*

   The objects edited last are displayed in time order in a list.

2. If required, you can go to an object by double-clicking on it.

Displaying Workflows for an Object

Use

All workflows that are running or have been executed, in which your document (object) has already been edited, are displayed. You can search in the database or in archives.

> When searching in the archive you should use the SAP archive info system. You must also activate [Ext.] the archive info system SAP_BO_2_WI_001 (from the central archiving transaction SARA via *Infosystem* → *Customizing*) and create [Ext.] it (via *Infosystem* → *Status*).

Procedure

4. In your object display, open the submenu with the symbol ▶ in the toolbox.
5. If you want to search in the database in the SAP System, you must choose *Workflow Overview*.

   The corresponding workflows are displayed. You do not need to execute the following steps.
6. If you want to search archives, you must choose *Archived Workflows*. The search report selection screen is displayed in the next screen, in which your object type and its key is already entered. For further information see *Help* → *Application Help.* Start the report.

   If the SAP archive info system is connected (see above), the archives available are searched immediately. Otherwise you must still explicitly select the archive that is to be searched for workflows.

Displaying Transmissions for an Object

Use

You can display all transmissions that refer to your document (your object). This for example can be:

- Notes that have been sent with the generic object services [Ext.]
- Message objects (for example notes) that were generated directly from the application program
- Message objects that were generated with Message Control

Prerequisites

The general prerequisites [Ext.] apply. If the application program has generated a BOR object MESSAGE directly, it must enter the object ID in the import parameter *Reference Type*.

Procedure

15. In your document display, open the submenu with the symbol in the menu of the services available.
16. Choose *Sent Documents.*

Result

As well as the documents sent, the number of recipients is also displayed and the number who have already read the documents.

Displaying Relationships

Use

This service is used to display relationships that already exist (for example, with other application documents). This does not apply to Business Workplace attachments (for example, PC documents) which can be found in the Attachment List [Page 24].

Prerequisites

The general prerequisites [Ext.] apply.

Procedure

2. In your document display (your object) choose *Display Relationships* in the Toolbox [Ext.].

   The list of existing relationships is then displayed. You can arrange this list in columns (double-click on the column or the button). Other typical list functions are available, for example exporting as a spreadsheet (using) or printing (using).

3. If required, you can select an object from the list and display further relationships by using .

4. By selecting or double-clicking you can go to the display of any object from the list.

Displaying Attachments

Use

In contrast to the relationship list, BOR objects are not displayed here, but rather Business Workplace Attachments (stored/not stored in an archive). These are mostly PC documents.

Prerequisites

The general prerequisites [Ext.] apply.

Procedure

5. In the display of your document, choose *Attachment list* in the menu of available services.

   The list of existing relationships is then displayed. You can arrange this list in columns (double-click on the column or the button). Other typical list functions are available, for example exporting as a spreadsheet (using) or printing (using).

6. By selecting or double-clicking you can go to the display of any object from the list.

Application Connection to Object Services

Purpose

Every application that will use at least one of the object services can access this through a small number of program steps. Here, it can decide

- which object services it will offer
- how the object services should be offered

Process Flow

- Some generic object services use the object relationships. The BOR IDs are linked. Your document (document header, item...) must be available and released as an object type in the BOR. Also see the workflow documentation Create object type [Ext.].
- Calling the services depends on whether or not your object is unique:
  - In your application the object is unique. Then you **publish** it and generate an instance of *cl_gos_manager*. You must put the object ID in the export parameter of the type *borident*:

    ```
    data:               go_myobject type ref to
    cl_gos_manager,

                        ls_object type borident
    ```

    …

    ```
    create object go_myobject

                        exporting is_object = ls_object

                                            no_commit = ‚x‘

                        exceptions     others = 1
    ```

    If the parameter *no_commit* is set, you enter that your application starts the *Commit Work* command. That is also the default value. If you explicitly transfer the parameter empty, the object services create the *Commit Work*. However, your data should be consistent when called because the database changes can no longer be reversed using *Rollback*.

    ➡

    If you do not know which object type your application document has, choose *Tools → Business Workflow → Development, Business Object Builder, Business Object Repository, Repository Browser* and choose *All object types*. Search for your application document in the outputted list using *Edit → Search*. For further information see *Business Object Repository Browser* in the workflow documentation

  - Which object the services are to take up is not decided until the process flow of the transaction. To avoid having to program several publication calls, you should in this case publish a callback [Page 27].

You can always only publish one object. If you publish many times, then it is always the case that only the last publication is valid.

For multiple publication, you can use the old function module SWU_OBJECT_PUBLISH. A window is then displayed when calling the services where the published objects can be selected. You should avoid this additional hurdle for the user if possible.

- In the standard system, the services are displayed in their own window (the "toolbox") which appears when the user presses the button. The following possibilities are available for more specific displays:
  - You do not want to offer a separate window but rather the services in the context menu ("right mouse button“). In this case you do not publish anything, but rather request a context menu for your object [Page 29].
  - You want to offer the toolbox directly [Page 32]. Also in this case you do not publish.
- All available services are offered in the standard system. You can restrict [Page 30] the services available or also only offer one service [Page 31].

Publishing Callback

Use

You use the callback to return the responsibility for determining the object to generic object services. For this, the services call a method that you have implemented. At runtime, your application only publishes the callback.

Procedure

1. Implement the interface method `if_gos_callback~get_object` in a local class:

```
class lcl_myclass definition.
      interfaces if_gos_callback.
      …
endclass.

class lcl_myclass implementation.
      method if_gos_callback~get_object.
      …
      endmethod.
endclass.
```

2. Program the callback (more precise: the publication of the callback)

```
data: go_myobject type ref to cl_gos_manager,
      lo_callback type ref to lcl_myclass
…
create object lo_callback.
…
create object go_myobject
      exporting   io_callback = lo_callback
      exceptions  others = 1
```

Result

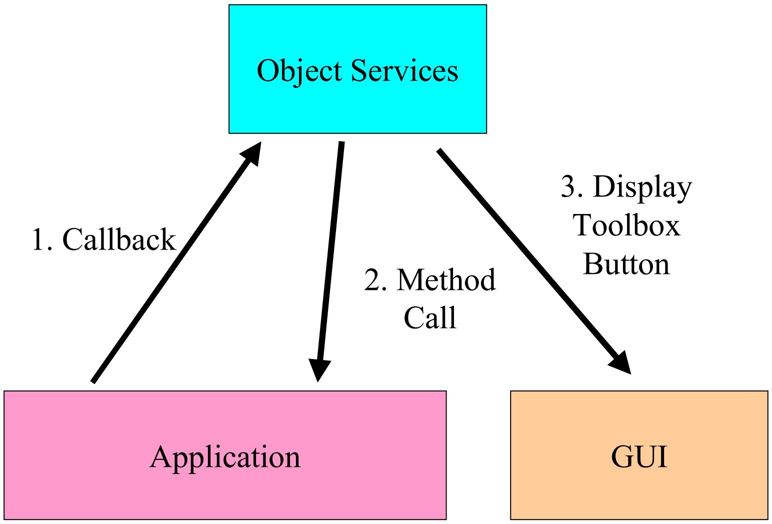
At runtime, the following now happens:

1. Your application publishes the callback and provides an instance of your local class *lcl_myclass*.

Publishing Callback

2. As soon as the user demands the menu of available services, method *get_object* is called from your local class and thus determines the current object.

3. The button for the toolbox is inserted at the top of the screen (GUI).



Offering the Context Menu

Use

Instead of a toolbox in a separate window, you can insert the service in a context menu that can be displayed by positioning the cursor on the object and right mouse-clicking.

Procedure

1. When generating an instance of *cl_gos_manager* you do not transfer an object identifier (or callback class). No object is published:

   ```
   data:     go_myobject type ref to cl_gos_manager.
   …
   create object go_myobject.
   ```

2. You do not transfer the object until you request the context menu:

   ```
   data:   ...,
               lo_menu type ref to cl_ctmenu,
               ls_object type borident.
   …
   call method go_myobject->get_context_menu
               exporting     is_object = ls_object
               importing     eo_menu = lo_menu.
   ```

   The import parameter *lo_menu* contains a reference to the context menu.

3. You can append the context menu to various *controls*. Normally you must write an *event handler* which forwards the menu entries selected by the user. In the *event handler* call a method from *cl_gos_manager* for generic object services:

   ```
   call method go_myobject->dispatch_menu_command
                   importing ip_fcode = lp_fcode
                                      io_menu = lo_menu.
   ```

   You only have to request *lo_menu* here if you have requested more than one context menu within the same instance from *cl_gos_manager*.

Restricting Available Services

Use

As far as possible, you should avoid users being offered pointless services for your object. You can restrict the services in the standard menu window (the "toolbox") or also in a context menu [Page 29].

Procedure

When generating an instance from *cl_gos_manager* enter a selection table for the constructor, with which you can enter intervals or exclude specific services. For information on possible entries see the Documentation for Selection Tables [Ext.].

You want to forbid Notes [Page 10] for an object.

```
data:   go_myobject type ref to cl_gos_manager,
   lt_services type tgos_sels,
   ls_service type sgos_sels,
   ls_object type borident.
…
ls_service-sign = ‚E‘.
ls_service-option = ‚EQ‘.
ls_service-low = ‚NOTE_CREA‘.
append ls_service to lt_services.
ls_service-low = ‚PERS_NOTE‘.
append ls_service to lt_services.

create object go_myobject
             exporting       is_object = ls_object

   it_service_selection = lt_services
             exceptions others = 1.
```

Starting Only One Service Directly

Use

You only want to offer one service for your object. The service is started directly, that is, not only when the button is pressed.

Procedure

1. When generating an instance from *cl_gos_manager* you do not enter an object ID. Thus the object is not published:

```
data:     go_myobject type ref to cl_gos_manager.
…
create object go_myobject.
```

2. The name of the service requested (from table SGOSATTR) provides you with the method *start_service_direct*, for example for the service Displaying Attachments [Page 24]:

```
data:     ...,
     lo_container type ref to cl_gui_custom_container,
     ls_object type borident.
…
call method go_myobject->start_service_direct
     exporting      ip_service = ‚VIEW_ATTA‘
                              is_object = ls_object
                              io_container =
lo_container.
```

You can enter a control container for specific services in the parameter *io_container*. The services that allow this option can be found in table SGOSATTR.

Offering the Toolbox Directly

Use

You don't want the services to be available only when the button is pressed, but rather immediately.

Procedure

3. When generating an instance from *cl_gos_manager* you do not enter an object ID. Thus the object is not published:

   ```
   data: go_myobject type ref to cl_gos_manager.

   …

   create object go_myobject.
   ```

4. When calling directly, you transfer the object ID as a parameter:

   ```
   data: ...,
         lo_container type ref to cl_gui_custom_container,
         ls_object type borident.

   …

   call method go_myobject->display_toolbox
               exporting         is_object = ls_object
                                       io_container = lo_container.
   ```

   The parameter *io_container* is optional: If you do not transfer a reference to a container, then the toolbox is started in a separate system like in the standard system.

Calling Services During Creating or Changing

Use

Create

The key is not yet known when creating an object. However, many services need the key, so you must subsequently hand it in.

Change

When changing an object, you must ensure that an implicit database commit is set after every screen change.

Procedure when Creating

1. When generating an instance of *cl_gos_manager* set the constructor parameter *ip_no_instance* and leave the object key empty:

```
data: go_myobject type ref to cl_gos_manager,
      ls_object type borident
…
ls_object-objtype = ‚BUS0815‘
create object go_myobject

exporting
            is_object = ls_object
                        ip_no_instance = ‚X‘
      exceptions  others = 1
```

2. Since your object is persistent, you must subsequently hand in the key:

```
ls_object-objkey = my_key.
call method go_myobject->set_id_for_published_object
      exporting        is_object = ls_object.
```

Creating New Object Service

Purpose

A new object service should be, by definition, generally usable.

Prerequisites

The object must be entered in the Business Object Repository (BOR).

Process Flow

You must execute the following steps:

- Develop your service (as a function module, for example).
- Your new service must be enclosed as the method *execute* in a separate class. The class inherits this instance method from the abstract class *cl_gos_service*. Your service is the composite definition of the method.
- Enter your class in the table SGOSATTR. Also use the table view maintenance (SM30) and see the F1 documentation.
- Finally enhance the documentation in the Knowledge Warehouse for the documentation for your new service. For this you use the Knowledge Workbench.

Result

Your service is displayed if the application is connected to generic object services [Page 25] and it is not explicitly excluded [Page 30].